NEGOTIATION AMONG SELF-INTERESTED COMPUTATIONALLY LIMITED AGENTS

A Dissertation Presented

by

TUOMAS W. SANDHOLM

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial
fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 1996

Department of Computer Science

NEGOTIATION AMONG SELF-INTERESTED COMPUTATIONALLY LIMITED AGENTS

A Dissertation Presented

by

TUOMAS W. SANDHOLM

Approved as to style and content by:

Victor R. Lesser, Chair

James Kurose, Member

Shlomo Zilberstein, Member

Herbert Gintis, Member

Mark S. Fox, Member

David W. Stemple, Department Chair
Computer Science

To Leena, Markus, Christina, and Joonas.

ACKNOWLEDGMENTS

I thank my advisor, Professor Victor Lesser, for countless stimulating discussions and for his faith in letting me pursue these new directions. Throughout my graduate school years I have appreciated his warm personality and the good care that he has taken of me academically, financially, and as a friend. I am indebted to him for his support in a remarkable range of matters which include being available for me on a daily basis for both intellectual and mundane matters, putting me in close contact with the multiagent systems research community, and facilitating the waiving of university regulations to allow completion of the dissertation in a shorter time than is normally mandated. His dynamism, flexibility and experience have also inspired, encouraged and taught me in a number of important ways. I hope that in advising graduate students I can do even half as well as Victor.

I thank Professor Neil Immerman for the backward pass idea for Algorithm 5.1, and Professor Herbert Gintis from the Economics Department for game theoretic comments. I also thank my other dissertation committee members Professor Shlomo Zilberstein, Professor Jim Kurose, and Professor Mark Fox for their helpful and insightful comments. I also thank them for accommodating my schedule in theirs'.

I also thank the distributed AI community for interesting discussions, and a number of anonymous referees for important comments. I would particularly like to thank Professors Jeffrey Rosenschein, Sandip Sen and Michael Wellman for the support that they have shown toward my work.

The Umass computer science graduate student populus has been extremely supportive of me socially and on practical issues. I owe them all. I thank Timur Friedman for numerous favors and just for being a friend. I thank Neil Berkman for his intellectually rigorous collaboration on machine learning research that is not part of this dissertation, and for blackjack and other good times. I thank Frank Klassner and Dan Neimann for proof-reading papers of mine, for other favors, and for pleasant get-togethers. I thank Kousha Etessami for in-depth discussions about Algorithm 5.1. I thank Claire Cardie and Alan Garvey for supplying me with template files. I also thank the whole set of people who volunteered their time to listen to my practice talks and to give valuable comments.

I gratefully thank the following sources that provided me with funding at different stages of the dissertation research (the content does not necessarily reflect the position or the policy of any of these and no official endorsement should be inferred): ARPA contract N00014-92-J-1698, National Science Foundation grant IRI-9523419, University of Massachusetts at Amherst Graduate School Fellowship, Technical Research Centre of Finland, Finnish Culture Foundation, Finnish Science Academy, Leo and Regina Wainstein Foundation, Jenny and Antti Wihuri Foundation, Honkanen Foundation, Ella and George Ehrnrooth Foundation, Finnish Information Technology Research Foundation, Transportation Economic Society, and Thanks to Scandinavia Foundation. Early development phases of the TRACONET system—which I carried out in Finland—where funded by the Technology Development Centre of Finland, and the industrial parties of the TOP-project. I also thank the latter for the real-world data that was used in the experiments.

I thank my parents and my brother for all the moral support that they have continuously provided remotely from Finland during my four years in graduate school. I also thank them for providing a supportive, free, and stable environment to grow up in. I especially thank Leena for all her efforts in my upbringing.

Last but certainly not least I thank my partner Christina Fong for all the help and support she has given me during my graduate school years. Christina made this time well-rounded, meaningful and fun. She proof-read more of my papers than anybody else. Much more importantly, she has always been there for me. She repeatedly acted as a task overflow buffer when my schedule turned out too tight. I have been on the receiving end. Thank you.

ABSTRACT

NEGOTIATION AMONG SELF-INTERESTED COMPUTATIONALLY LIMITED AGENTS

SEPTEMBER 1996

TUOMAS W. SANDHOLM

M.S., INDUSTRIAL ENGINEERING AND MANAGEMENT SCIENCE, HELSINKI UNIVERSITY OF TECHNOLOGY

M.S., COMPUTER SCIENCE, UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D.,COMPUTER SCIENCE,UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Victor R. Lesser

In multiagent systems, computational agents find and make contracts for the real world parties that they represent. The significance of such systems is likely to increase due to the developing communication infrastructure, electronic commerce, and the industrial trend toward outsourcing. This dissertation analyzes such negotiations among agents that try to maximize payoff without concern of the global good. Accordingly, the interaction protocols need to be designed normatively so that the desired local strategies are best for the agents—and thus the agents will use them—and then desirable social outcomes follow. This dissertation extends game theory to settings where computational limitations preclude enumerating and evaluating all possible outcomes. It contributes to three interlinked areas of negotiation: contracting, coalition formation, and contract execution.

Within automated contracting, the contract net framework is extended for self-interested, computationally limited agents. It is augmented with a formal model for making bidding and awarding decisions based on marginal cost calculations. These are intractable, so approximations are presented and their properties discussed. It is proven that a leveled commitment contracting protocol enables contracts that are impossible via classical full commitment contracts. Leveled commitment also increases efficiency when full commitment contracts are possible. The analysis incorporates insincere decommitting. Next, an iterative scheme for anytime task reallocation is presented. Necessary and sufficient contract types are devised for reaching a globally optimal task allocation in a finite number of contracts. Finally, contracting implications of limited computation, and issues in distributed asynchronous implementation are discussed.

Within coalition formation, a normative theory for self-interested, computationally limited agents is developed. It states which agents should form coalitions and which coalition structures are stable. These analytical prescriptions depend on the performance profiles of the agents' algorithms.

Within contract execution, a method for carrying out exchanges without enforcement is devised via splitting the exchange into chunks which the agents alternate delivering. Algorithms are devised for chunking and chunk sequencing. Optimal exchange strategies are derived.

The possibility of scaling up is shown experimentally on an $\mathcal{NP}$-complete distributed vehicle routing problem with large-scale real-world data. The techniques work in this setting where classical game theoretic techniques are intractable.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION AND MOTIVATION

The central theme of the dissertation is negotiation among self-motivated agents whose rationality is limited by computational complexity. The developed methods are targeted to inherently distributed combinatorial problems—e.g. resource and task allocation and multiagent planning and scheduling—in situations where agents may have different goals, and each agent is trying to maximize its own good without concern of the global good. In building computer support for negotiations among independent businesses or individuals, the issue of self-interest naturally arises.

The importance of automated negotiation systems with self-interested agents is likely to increase as a result of three developments. One is the growth of a standardized communication infrastructure—EDI, NII, KQML [Finin *et al.*, 1992], Telescript [General Magic, Inc., 1994], JAVA etc—over which separately designed agents belonging to different organizations can interact in an open environment in real-time, and safely carry out transactions [Kristol *et al.*, 1994, Low *et al.*, 1994b, Kristol *et al.*, 1994, Sandholm and Lesser, 1995b]. The second is the advent of small transaction commerce on the Internet for purchasing goods, information, and communication bandwidth [Kalakota and Whinston, 1996, Office of Technology Assesment, 1994]. The third is an industrial trend toward virtual enterprises: dynamic alliances of small, agile enterprises which together can take advantage of economies of scale when available (but do not suffer from diseconomies of scale), and can respond to more diverse orders than they could individually.

Automated negotiation methods best suit domains, where each contract is a small operative level decision. In such domains with low monetary stakes per contract, it is more likely that automatic negotiation systems are relied on and given authority in practise. Although each individual contract is monetarily minor, the negotiation process is usually nontrivial. Moreover, the cumulative stakes of multiple negotiations may be high.

In general, with the speedup of communication links, one of the reasons for distributing computational tasks and reasoning—communication bandwidth limitations—is growing relatively less important (distribution may still be warranted e.g. due to distributed processing units). This is true for domains where each agent's information is mostly in a compact symbolic form such as in planning and scheduling domains. In such settings, the communication of the input data into a central location is usually feasible. On the other hand, in interpretation domains where each node contains large amounts of low level numeric data, distributed processing is still warranted due to bandwidth limitations [Lesser and Erman, 1977, Lesser and Erman, 1980, Lesser and Corkill, 1981, Lesser, 1991, Lesser and Corkill, 1983, Durfee and Lesser, 1989, Durfee and Lesser, 1991]. The bandwidth limitation also warrants distributed processing in some special domains such as autonomous submersibles that have to communicate over a very low bandwidth acoustic link [Turner and Eaton, 1994].

In the setting of this dissertation, the very reason for distributed reasoning in planning and scheduling domains is the self-motivation of the agents—instead of the bandwidth limitation. In a joint problem solving process they want the solution to be as beneficial to themselves as possible. Similarly, an agent may want to hide some information that the other agents could possibly use against it in later problem solving, e.g. proprietary customer information. Thus distributed problem solving is warranted because the problem is inherently distributed as opposed to needing to be distributed due to computational or communication reasons. In such settings, the trend of increasing bandwidth actually enables negotiation applications that were infeasible before because more negotiation message traffic has become viable.

In *cooperative distributed problem solving* [Durfee *et al.*, 1989], the system designer imposes an interaction *protocol* [1] and a *strategy* (a mapping from state history to actions; a way to use the protocol) for each agent. The approach is usually *descriptive*: with the given protocol, if the agents use the imposed strategies, then certain social outcomes follow. On the other hand, in *multiagent systems* [Sandholm and Lesser, 1995c, Sandholm and Lesser, 1996, Sandholm and Lesser, 1995a, Sandholm and Lesser, 1997, Rosenschein and Zlotkin, 1994, Durfee *et al.*, 1993, Kraus *et al.*, 1992, Wellman, 1992], the agents are provided with an interaction protocol, but each agent will choose its own strategy. A self-interested agent will choose the best strategy for itself, which cannot be explicitly imposed from outside. The protocols need to be designed *normatively* so that the desired local strategies are best for the agents—and thus the agents will use them—and then certain social outcomes follow. This allows the agents to be constructed by separate designers and/or represent different real world parties.

Interactions of self-motivated agents have been widely studied in microeconomics—especially in the subfield of game theory [Fudenberg and Tirole, 1991, Kreps, 1990, Rasmusen, 1989]. Most of the results assume perfect rationality of the agents [Schweers Cook and Levi, 1990, Howard, 1971]. Perfect rationality assumptions show up in many ways, for example as assumptions of flawless deduction, optimal reasoning about future contingencies and recursive modeling of other agents. Perfect rationality also assumes that agents can compute their marginal costs for tasks exactly and immediately, which is untrue in many practical situations. Lately, microeconomists have concentrated more on the issue of bounded rationality, where cognitive limitations of the interacting humans are brought into the discussion [Simon, 1982, Good, 1971, Heiner, 1983, Heiner, 1985]. As economists study interactions among humans, their

[1]By protocol we do not mean a low level communication protocol, but a negotiation protocol. An example protocol is the *auction*, where some agents bid to take responsibility for a task, which is awarded to the lowest price bidder. The analog of a negotiation protocol is called a *mechanism* in game theory [Fudenberg and Tirole, 1991, Kreps, 1990].

models of bounded rationality have been mostly *descriptive*. They have been used positively for example in simulations and they have been used to critique some of the earlier results achieved in game theory under the perfect rationality assumption. Humans are variable in their rationality, and a person's level of rationality is hard to characterize formally. Therefore it may be that these descriptive characterizations are the best one can achieve. On the other hand, the rationality of computational agents is limited in a very precise sense. Given an algorithm, an execution architecture, and a problem instance distribution, it can be statistically analyzed how well an agent is able to solve problems. As will be shown in this dissertation, the formal character of these rationality limitations allows for the formulation of *normative* (prescriptive) theories of interactions of self-motivated computationally limited agents.

1.1 Example Application Domains

To motivate the research from a practical perspective, the following sections describe example application domains where the methods developed in this dissertation are needed. The main criteria are that the agents are self-interested, and that there is an underlying intractable combinatorial problem that limits the agents rationality because the problem cannot be solved optimally in practice.

1.1.1 Distributed Vehicle Routing among Dispatch Centers

The distributed vehicle routing problem that we study is structured in terms of a number of geographically dispersed dispatch centers of different companies. Each center is responsible for certain deliveries and has a certain number of vehicles to take care of them. So each agent—representing a dispatch center—has its own vehicles and delivery tasks. The local problem of each agent is a heterogeneous fleet multi-depot routing problem with the following constraints.

- Each vehicle has to begin and end its tour at the depot of its center (but neither the pickup nor the drop-off locations of the orders need to be at the depot).

- Each vehicle has a maximum load weight constraint. These differ among vehicles.
- Each vehicle has a maximum load volume constraint. These also differ among vehicles.
- Each vehicle has the same maximum route length (prescribed by law).
- Every delivery has to be included in the route of some vehicle.

The objective is to minimize transportation costs: The domain cost is the sum of the route lengths of the vehicles in the solution that has been reached.

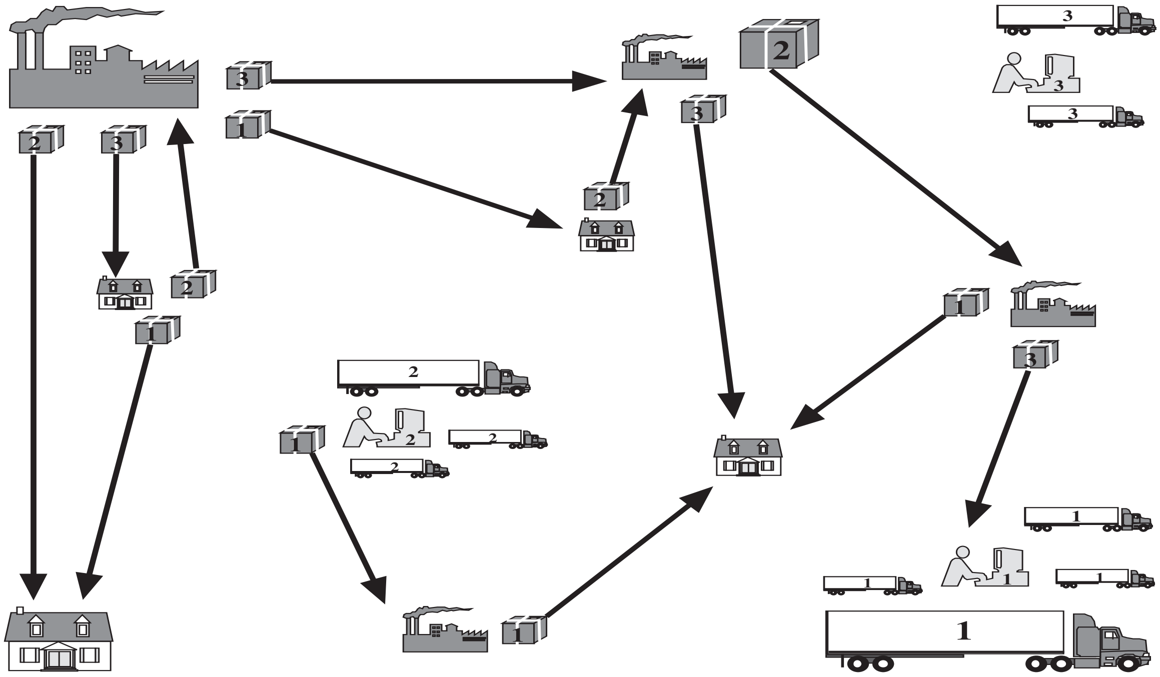

Figure 1.1. Small example problem instance of distributed vehicle routing. This instance has three dispatch centers represented in the figure by computer operators that get the delivery orders and route the vehicles. Each parcel is numbered according to the dispatch center that is responsible for delivering it. A solution to the problem of a certain dispatch center individually is a specification of routes for its vehicles in a manner that all the center's parcels get delivered, the routes begin and end at the dispatch center, and driving distance is minimized. No routing solution is presented in the figure.

The problem is $\mathcal{NP}$-hard, because ΔTSP can be trivially reduced to it. It is in $\mathcal{NP}$, because the cost and feasibility of a solution can easily be checked in polynomial time. Thus, the problem is $\mathcal{NP}$-complete. Moreover, the problem instances in our experiments are so large that even the smallest ones are too hard to solve optimally—unlike the one in Figure 1.1.

The geographical main operation areas of the centers overlap considerably. This provides for the potential for multiple centers to be able to handle a delivery. The cost of handling a delivery may vary between agents because the delivery can often be less expensively integrated into adjacent routes than remote ones—honoring the weight, volume and route length constraints though. So it can often be incorporated with lowest cost into the routing solution of the center that happens to have adjacent routes. The asymmetric costs among agents for handling a delivery make it often beneficial to reallocate delivery tasks among agents. This allows considerable cost savings from negotiation-based coordination among the agents.

Distributed vehicle routing is a real world problem, and the problem instances used in the experiments of this dissertation were collected from five real dispatch centers. They represent one week delivery order and vehicle data. Company A owned the first three centers and company B owned the last two. Even though some of the dispatch centers were owned by the same company, in practise they acted self-interestedly because they had their own fiscal goals. The centers were located around Finland. The collected data is characterized in Table 1.1. Centers 2, 3 and 5 were located near each other, while 1 and 4 were far from each other and the other centers. Centers 1, 3, 4 and 5 transported heavy low volume items, while 2 transported light voluminous items. In general, adjacent centers have more potential savings from cooperation. Secondly, heavy and light items can often be beneficially combined to a load without violating the maximum load weight or maximum load volume constraint.

Table 1.1. One week real vehicle and delivery data of the experiments.

Dispatch center	Number of deliveries	Number of vehicles	Average delivery length
1	65	10	121 km
2	200	13	169 km
3	82	21	44 km
4	124	18	145 km
5	300	15	270 km
All	771	77	187 km

I have explored the distributed vehicle routing problem extensively [Sandholm, 1991, Sandholm, 1993, Sandholm and Lesser, 1995a, Sandholm and Lesser, 1997, Sandholm, 1992a, Sandholm, 1992b, Linnainmaa *et al.*, 1992, Sandholm and Lesser, 1995b], and lately other researchers have studied an almost identical distributed vehicle routing problem, yet with randomly generated instances and with a non-normative approach [Fischer *et al.*, 1995].

1.1.2 Production Planning and Scheduling in Multi-Enterprise Manufacturing

Although the experiments of this dissertation are based on the distributed vehicle routing problem, there are several other real world application domains that require the techniques developed in this dissertation. Among the most important ones is production planning and scheduling in multi-enterprise manufacturing.

In a production planning and scheduling problem, an agent has a set of tasks (manufacturing operations, setup operations, etc.) and a set of resources (machines, people, storage areas, guided carts etc.). The problem involves planning the assignments of tasks to resources and assigning times—i.e. scheduling—for the execution of the tasks on the resources. Many different objective functions—i.e. cost functions—have been used in such problems, for example minimizing tardiness of jobs (a job is an ordered set of operations), minimizing in-process inventory (as in the philosophy of Just On Time (JOT)-manufacturing), minimizing completion time of the last operation, etc. Any one of these objective functions and the constraints—operation order constraints, resource capacity limitations, restrictions on which operations can be assigned to which resources etc.—defines a constrained optimization problem. Because the $\mathcal{NP}$-complete job-shop scheduling problem [2] is a special case of many real world manufacturing problems, the latter are usually $\mathcal{NP}$-hard. Because the solution cost and feasibility of a complete solution can usually be checked in polynomial time, the manufacturing problems are usually in $\mathcal{NP}$. It follows that they are usually $\mathcal{NP}$-complete.

In multi-enterprise manufacturing, different enterprises can handle the same operation. The cost (and feasibility) of handling an operation may vary among enterprises—which can be represented by computational agents. Therefore it is sometimes beneficial to reassign operations from one agent to another. This provides for the potential savings achievable by negotiation-based coordination among agents.

There are three main types of distributed manufacturing environments based on the types of agents that constitute the system. I will call these agent types *cooperative*, *self-interested (SI)*, and *hostile*. Cooperative agents attempt to maximize social welfare, which is the sum of the agents' utilities. They are willing to take individual losses in service of the good of the society of agents. As an example, different production cells within an enterprise should act as cooperative agents. They should attempt to minimize production costs and maximize the revenues of the company as a whole—sometimes accepting local losses in order to facilitate production at other cells.[3] In *multi-enterprise agile manufacturing*, individual companies join together to form a so called *virtual enterprise*, which will take care of production tasks—usually

[2] Real-world manufacturing scheduling problems often differ from the basic job-shop scheduling problem for example by having sequence dependent setup times, and alternative resources (with different characteristics) for an operation.

[3] However, when cells involve humans, individual cells often need incentives to motivate them to perform efficiently locally. Such incentives can usually only be implemented through local evaluation, which often introduce a disparity between the cell's local goals and the company's goals.

more economically than the companies could when operating individually. In a virtual enterprise, each individual company is usually a self-interested agent: it wants to maximize its own profit while not caring about the other companies' profits within the virtual enterprise. In such cases, an agent is willing to accommodate other agents' tasks only for a (monetary) compensation. The third type of agent relationship that occurs in distributed manufacturing is hostile. As an example, one can consider companies that compete against each other in the same market. In such a setting the company agents can be viewed as maximizing their utility which increases with their own gains but also with other companies' losses. Even if an agent is self-interested on a strategic level, it may be in its interest to act in a hostile manner in operative level distributed production scheduling, e.g. to attempt to drive competitors out of business. Working together towards coordinated distributed scheduling seems least fruitful and least feasible in the hostile setting. In many situations, the distributed manufacturing environment actually comprises of agents from more than one of the classes. For example, while the setting between companies is self-interested, the cell-wise distribution within each company is simultaneously cooperative.

Up to date, automated distributed production planning and scheduling systems have been almost exclusively developed to operate among cooperative agents [Sycara *et al.*, 1991, Neiman *et al.*, 1994, Burke and Prosser, 1994, Sen, 1994, Sen, 1996]. Some reasons for distributing such systems have been presented. First there is the bandwidth argument: information is mostly sustained at the local level and this saves communication costs. Detailed information regarding tasks, resources and their dynamic state is usually not transmitted between agents; only abstractions (meta-information) such as load profiles of each resource class of each agent are sent [Sycara *et al.*, 1991, Neiman *et al.*, 1994]. This bandwidth argument originates mostly from the Cooperative Distributed Problem Solving work of Lesser et al. (see e.g. [Lesser and Erman, 1980]), which was done in an interpretation domain. In such a domain, large amounts of low level information arrive through sensors and it may really be infeasible to transmit that information to a central location to be processed. On the other hand, in planning and scheduling domains, the problem can often (not always because some schedulers can use large databases, for example of prior operation statistics) be relatively succinctly described and transmitted, and the real complexity mainly stems from inherent combinatorics. Another problem with this line of argument is that search efficiency is usually reduced as information of the global problem becomes less precise. Moreover, the negotiation communication that is required for distributed search may actually exceed the amount of communication that would occur if all the information were just gathered to a central problem solver: often distributed problem solving messages have to be sent multiple times, e.g. due to backtracking or recontracting out a task set that was previously contracted in. Distribution has also been argued for by claiming that parallelization increases problem solving efficiency. This holds for (nearly) decomposable hierarchical systems, but tightly coupled domains—including many scheduling domains—are not necessarily easily decomposable to independent subproblems that would facilitate efficient parallelization. Also, the real world natural distribution of a scheduling problem seldom equals the distribution that would be computationally most efficient. Furthermore, parallel asynchronous search among agents often (yet not always [Conry *et al.*, 1991]) involves giving up desirable algorithm properties such as completeness [Sycara *et al.*, 1991]. Another issue supporting distribution in cooperative settings is reactivity: agents can locally react to local changes faster than a centralized system could. However, in many domains, the communication time to and from a central site is often negligible compared to the rate of change in the physical domain. Distribution in cooperative domains has also been advocated using a decision autonomy argument: the agents prefer to maintain the possibility to make local decisions and not to submit to centrally made ones. The problem with this argument is that cooperative agents have the same goals and therefore they should be indifferent regarding the question of which one of them makes the decisions. Finally, distributed systems have been argued for by pointing out that they are more robust against failure at a single point than centralized systems. Some distributed systems have been implemented with enough agent autonomy so that this claim holds. But many of the arguments presented up to date in favor of distributing cooperative scheduling systems do not hold in all cooperative scheduling settings: in some cases the problem could be solved more efficiently by keeping a centralized problem solver up to date about the distributed events and having it do the planning and scheduling.

On the other hand, among self-interested agents the very self-interest introduces an inherent distribution into the domain. Agents do want to maintain local decision autonomy because they have private goals. Moreover, they do not necessarily pass information truthfully—such an agent lies whenever it benefits from doing so. An agent may reveal only some of its tasks or resources or it might untruthfully reveal tasks or resources that do not exist. Rosenschein and Zlotkin have formally analyzed the issue of task revelation [Rosenschein and Zlotkin, 1994]. Unfortunately their analysis of task oriented domains does not apply to most scheduling domains because it assumes that each agent has sufficient resources to potentially handle all of the tasks of all agents, and that the agents have symmetric costs for handling tasks. Furthermore, the work of Rosenschein and Zlotkin shows the negative result that even in simple two agent non-capacity-constrained symmetric cost contracting, truthful task revelation is usually not achieved. Since these simple problems are special cases of many real world problems, the negative results apply to the real world problems also. As to other forms of insincere behavior, an agent can lie about the cost that it is willing to pay to another agent for taking care of some of its tasks, or about the cost that it requires in order to accept some of the other agents' tasks. Ephrati [Ephrati, 1994] presented an initial approach to solving the problem of exaggerating agents using the Clarke tax voting mechanism in a meeting scheduling domain. Several unresolved issues remain concerning this problem, for example the applicability of the proposed mechanism to a sequence of decisions and to dynamic domains with different expectations of changing tasks and resources. Even further forms of lying exist, e.g. an agent can declare false load profiles on resources or otherwise lie about its dynamic status. In such cases the system designer cannot be certain of the efficiency of the negotiation network because it is not obvious how self-interested agents will behave. Because synergy savings from joint problem solving are often available among scheduling agents and virtual enterprises are becoming a reality, and because self-interested agents are inherently distributed, support mechanisms for distributed planning and scheduling among self-interested agents are clearly called for. The developing inter-enterprise communication infrastructure provides part of the appropriate technology push for distributed planning and scheduling applications. Such automated systems have to take into account the issues that arise from self-interest: an agent will take action only for a payment, and an agent may lie. Therefore most of the techniques developed for cooperative distributed scheduling are inappropriate for the self-interested setting.

Multi-enterprise manufacturing planning and scheduling problems provide a very promising application domain for the techniques of this dissertation: the agents are self-interested, and there is an underlying combinatorial problem that limits the rationality of the agents. For example, an agent is unable to exactly compute the cost associated with accepting tasks from other agents, and the savings from having other agents take care of some of the agent's tasks.

1.1.3 Meeting Scheduling

Another potential application for the techniques developed in this dissertation is distributed meeting scheduling [Ephrati, 1994, Sen, 1994, Sen, 1996]. Such problems involve the assignment of times (and locations) to meetings. Constraints can involve for example capacity limitations on locations and on attendants (usually at most one meeting per person at a time). The objective can be for example to maximize the sum of the attendees of the meetings, or to maximize the sum of the utilities that the agents assign to the schedule (different agents may have different time and place preferences). Again, the setting consists of self-interested agents (representing people), and there is an underlying combinatorial problem that limits the agents' rationalities. Embedding automated negotiation techniques for solving the problem into Personal Digital Assistants (PDAs) would clearly be useful.

1.2 Criteria for Mechanism Evaluation

Negotiation protocols—i.e. mechanisms—can be evaluated according to many types of criteria. These include social welfare, Pareto efficiency, individual rationality, stability, symmetry, computational efficiency, distribution, and communication efficiency. These are discussed shortly in the next sections.

1.2.1 Social Welfare

Social welfare [Fudenberg and Tirole, 1991, Kreps, 1990, Rasmusen, 1989, Rosenschein and Zlotkin, 1994] is the sum of all agents' payoffs—or utilities—in a given solution. It measures the global good of the agents. It can be used as a criterion for comparing alternative mechanisms by comparing the solutions that the mechanisms lead to. When measured in terms of utilities, the criterion is somewhat arbitrary, because it requires interagent utility comparisons, and really each agent's utility function can only be specified up to affine transformations.

1.2.2 Pareto Efficiency

Pareto efficiency [Fudenberg and Tirole, 1991, Kreps, 1990, Rasmusen, 1989, Rosenschein and Zlotkin, 1994] is another solution evaluation criterion that takes a global perspective. Again, alternative mechanisms can be evaluated according to Pareto efficiency by comparing the solutions that the mechanisms lead to. A solution x is Pareto efficient—i.e. Pareto optimal—if there is no other solution x' such that at least one agent is better off in x' than in x and no agent is worse off in x' than in x. So, Pareto efficiency measures global good, and it does not require questionable interagent utility comparisons.

Social welfare maximizing solutions are a subset of Pareto efficient ones. Once the sum of the payoffs is maximized, an agent's payoff can increase only if another agent's payoff decreases.

1.2.3 Individual Rationality

Participation in a negotiation is individually rational to an agent if the agent's payoff in the negotiated solution is no less than the payoff that the agent would get by not participating in the negotiation. A mechanism is individually rational if participation is individually rational for all agents. Only individually rational mechanisms are viable: If the negotiated solution is not individually rational for some agent, that self-interested agent would not participate in the negotiation.

1.2.4 Stability

Among self-interested agents, mechanism should be designed to be stable (non-manipulable), i.e. they should motivate each agent to behave in the desired manner. This is because if a self-interested agent is better off behaving in some other manner than desired, it will do so. Sometimes it is possible to design mechanisms with *dominant strategies*. This means that an agent is best off by using a specific strategy no matter what strategies the other agents use.

Often an agent's best strategy depends on what strategies other agents choose. In such settings, dominant strategy mechanisms are impossible to implement. For such settings, other stability criteria are needed. The most basic one is the *Nash equilibrium* [Nash, 1950b, Fudenberg and Tirole, 1991, Kreps, 1990, Rasmusen, 1989, Luce and Raiffa, 1957, Binmore, 1994, Friedman, 1990]. The strategy profile $S_A^* = \langle S_1^*, S_2^*, ..., S_{|A|}^* \rangle$ among agents A is in equilibrium if for each agent i, S_i^* is the agent's best strategy—i.e. best response—given that the other agents choose strategies $S^*_{A-i} = \langle S^*_1, S^*_2, ..., S^*_{i-1}, S^*_{i+1}, ..., S^*_{|A|}\rangle$. In other words, in Nash equilibrium, each agent chooses a strategy that is a best response to the other agents' strategies.

There are two main problems in applying Nash equilibrium. First, in some games there exist no strategies that form an equilibrium [Fudenberg and Tirole, 1991, Kreps, 1990, Rasmusen, 1989]. Second, some games have multiple Nash equilibria, and it is not obvious which one the agents should actually play [Kreps, 1990].

There are also limitations regarding what the Nash equilibrium guarantees even when it exists and is unique.

First, in sequential games it only guarantees stability in the beginning of the game. In a later stage—i.e. in a subgame—the strategies need not be in equilibrium any more. A refined solution concept called the *subgame perfect Nash equilibrium* is defined to be a Nash equilibrium that remains a Nash equilibrium in every subgame (even subgames that are not along the actual path of play and will thus never be reached) [Selten, 1965, Fudenberg and Tirole, 1991, Kreps, 1990]. This solution concept also suffers from existence and uniqueness problems.

Second, the Nash equilibrium only guarantees that no agent alone is better off by switching to another strategy. Agents might collude, and a coalition may be better off by jointly changing the strategies of the agents in the coalition. Coalition-proof refinements of the Nash equilibrium, and other solution concepts for coalitions are presented in Section 2.1.

Sometimes efficiency goals and stability goals conflict. A simple example of this is the two-person Prisoner's Dilemma game [Fudenberg and Tirole, 1991, Kreps, 1990, Rasmusen, 1989] where the unique welfare maximizing and Pareto efficient strategy profile is the one where both agents cooperate, Table 1.2. On the other hand the only dominant strategy equilibrium and Nash equilibrium is the one where both agents defect.

Table 1.2. Example payoff matrix of a two-person Prisoner's Dilemma game. The payoffs of the row player are listed first.

		column player	
		cooperate	defect
row	cooperate	3, 3	0, 5
player	defect	5, 0	1, 1

1.2.5 Symmetry

It is desirable for a mechanism to be symmetric. This means that the mechanism does not handle any agent in an *a priori* preferred or dissimilar manner. Secondly, it means that agents that actually act exactly alike should receive the same payoffs.

1.2.6 Computational Efficiency

Obviously, mechanisms should be designed so that when agents use them, as little computation is needed as possible. Classically, mechanisms have been designed so that they lead to domain solutions that satisfy some of the above evaluation criteria. Of these mechanisms, the ones with the lowest computational overhead have been preferred.

Because this dissertation explicitly deals with resource limited rationality, a slightly different and more detailed approach is taken. In Chapter 4 the cost of computation is explicitly factored into the agents' payoffs, and the above solution concepts are directly applied to the combination of domain actions and computation actions. According to this view, computational efficiency is not an additional orthogonal evaluation criterion. Instead it is incorporated in the other solution concepts. This allows the computation strategies to also fulfill desirable criteria such as social welfare maximization, Pareto efficiency, individual rationality, different forms of stability, and symmetry.

1.2.7 Distribution and Communication Efficiency

We want our mechanisms to be truly distributed in order to avoid a single point of failure among other reasons. Simultaneously we would like to minimize the amount of communication that is required to converge on a desirable global solution. In some cases these two goals conflict.

So far several desirable properties of negotiation mechanisms have been outlined. Some mechanisms fulfill many of these properties, but some of these desiderata are conflicting. In the next section the contributions of this dissertation are overviewed.

1.3 Overview of Contributions

This dissertation analyzes automated negotiation among self-interested computationally limited agents. It extends game theory, which has traditionally assumed perfect rationality. In combinatorially complex domains, it is unrealistic to assume perfect rationality, and deliberation costs have to be traded off against the advantages they bring. Some non-game theoretic DAI work has looked at such computational issues heuristically, but the dissertation takes a normative view: given an interaction protocol, what are the best strategies that agents can use—and therefore the strategies that self-motivated agents will use—and then what are the global outcomes. Domain independent methods for negotiation among such agents are presented.

In the approach of this dissertation, the issues are first analyzed formally, and then verified empirically when appropriate. The dissertation contributes to three interdependent subareas of automated negotiation: automated contracting, coalition formation, and contract execution. The contributions to automated contracting include the following.

- Contracting based on marginal costs. In the original contract net protocol the agents were cooperative, and they executed tasks for each other for free. This dissertation extends the protocol for self-interested agents by having agents pay each other for handling tasks. A formal model for making bidding and awarding decisions is presented, where the agents only accept individually rational contracts based on local marginal cost calculation. Approximation schemes for marginal cost calculation are presented that allow the agents to operate in combinatorial domains. The resulting negotiation scheme is an anytime task reallocation algorithm that can be terminated at any time, and is guaranteed to have a feasible solution for each agent that is no worse than the agent's initial solution: participation is individually rational. It is experimentally shown via a truly distributed asynchronous implementation that the technology scales to large scale problem instances.

- Leveled commitment contracting protocol. The protocol is first developed, and its practical advantages over contingency contracts are discussed. Then it is formally proven that the new protocol increases Pareto efficiency over full commitment contracts, and sometimes enables contracts that full commitment protocols would not allow based on individual rationality. The analysis is carried out formally in a number of contracting settings. The effects of agents' biased information are also discussed. Finally, practical implications of leveled commitment contracts to system builders are discussed.

- Choosing the stage of commitment dynamically in a contracting protocol. A new feature is introduced into contracting protocols in the sense that the stage of commitment is allowed to be decided dynamically based on the negotiation environment or even on a per negotiation item basis during the negotiation itself. This is a strict generalization of classical contracting where commitment takes place at the bidding phase. The varying stage of commitment allows a more efficient search for beneficial contracts.

- Richer negotiation protocol. The protocol also allows counterproposals, counter-counterproposals etc. *ad infinitum.* This also allows for a more efficient search for beneficial contracts among self-interested agents among which classical meta-information passing is not feasible. The protocol also allows the setting of specific parameters such as decommitment penalties and deadlines on a per negotiation issue basis.

- Necessary and sufficient contract types. It is first proven that contracts over one issue (task) at a time do not, in general, lead to a global optimum in the task allocation space—no matter how many task reallocation hill-climbing contract iterations are allowed. To alleviate the problem, three new contract types are introduced: *clustering contracts, swap contracts, and multiagent contracts.* It is shown that these do not suffice when used alone, in pairs, or even in threes when applied separately. However, they can be combined into a new contract type, *CSM-contract.* It is proven that this contract type is sufficient to guarantee that the globally optimal task allocation is reached with any hill-climbing algorithm in a finite number of steps.

- Analysis of contracting implications of limited computation. The necessity of local deliberation scheduling is substantiated. The deliberation controller has to decide which tasks sets to allocate computation on, how much, and in what order. Next the dissertation presents several previously unidentified tradeoffs regarding limited computation in a contracting setting. First, an agent can trade off complexity of marginal cost calculation against monetary risk. This occurs when an agent can bid or award while its previous bids are pending. The issue is formalized in different marginal cost approximation schemes. It is shown that under some of these schemes, the agent's local cost is guaranteed to decrease monotonically while under others it is not. Second, an agent has a tradeoff of getting more precise marginal cost estimates (and save computation) vs. being able to participate in multiple negotiations simultaneously. This is analyzed formally and experimentally. Again, only some of these alternatives guarantee monotonic decrease of local cost. Third, an agent has tradeoffs regarding sending messages early on vs. enlarging the bidding and awarding context by waiting for more incoming offers.

- Discussion of distributed asynchronous implementation of automated contracting. First, message congestion and agent saturation are discussed. The classical approaches to solving this problem are not viable among self-interested agents. Protocol related methods and local strategy related methods are presented for solving the problem. The latter alone were sufficient to solve it in practice. They are in concert with each agent's self-interest. Second, a method for terminating iterative refinement negotiations is presented that guarantees termination when a local optimum of the desired type has been reached. Third, the comparative advantages of reply-based protocols and timeout-based ones are discussed.

The second main subarea which this dissertation studies is coalition formation among computationally limited agents. A normative, domain-independent theory of coalitions in combinatorial domains is presented, where the rationality of self-interested agents is bounded by computational complexity. The computation limits are quantitatively modeled by a unit cost of computation and the performance profiles of the agents' problem solving algorithms. The agents are best off trading solution quality off against computation cost. As discussed, in contracting, the agents iteratively reallocate tasks among themselves to reach a globally more desirable solution. On the other hand, in the coalition formation part of the dissertation, the agents form coalitions, and task allocation and domain problem solving is pooled to occur centrally within each coalition. Yet, different coalitions solve their problems independently in a distributed manner. The contributions to coalition formation include the following.

- Domain classification for computationally bounded—i.e. bounded rational (BR)—agents. A Venn diagram-based analysis is carried out relating the domain classes for BR agents and rational ones. Two different rational agent domain classifications are included in the comparison, one from game theory and one by Rosenschein and Zlotkin from the multiagent research community. The domain classification carries with it information about the optimal coalition structure and its stability.

- Formal analysis of the social welfare maximizing coalition structure among BR agents. The presented formal model allows one to always determine the optimal coalition structure when the performance profiles and the computation unit cost are known. In addition, general theorems are proven that analytically state which types of performance profiles cause the best coalition structure to be the grand coalition irrespective of the computation unit cost (execution platform). Similar theorems are proven for the coalition structure where all agents work separately. It is shown that the optimal coalition structure of BR agents depends heavily on the computational limitations, and differs significantly from that of rational agents.

- Formal analysis of the stability of a coalition structure consisting of BR agents. The presented formal model allows one to always determine whether a coalition structure is stable among BR agents. Stability requires that no agent, subgroup, or the whole group of agents is motivated to switch to another coalition structure. These criteria correspond to the *core* solution concept, Sec. 2.1.2.1. There are games that have stable coalition structures for both rational and BR agents, for one but not the other, and for neither. Finally, general theorems are presented that relate the shape of the algorithms' performance profiles to the stability of the coalition structure.

- Experimental results in distributed vehicle routing. Real world one week vehicle and order data from five dispatch centers is used to see where this problem and these real problem instances fall in the domain classification. Although the grand coalition was always best for rational agents, the social welfare maximizing coalition structure varied widely for BR agents. Similarly, the stability of the coalition structure varied among BR agents. The experiments suggest that often—yet not always—with superlinear iterative refinement steps, low computation unit costs promote large coalitions while high computation unit costs suggests smaller ones. A plausible explanation for this phenomenon is presented. Another interesting observation is that the presented normative theory prescribes the bounded rational agents to choose coalition structures that agree closely with what human agents would select based on domain specific considerations. However, these coalition structures differ significantly from those which rational agents should choose.

The third main subarea to which the dissertation makes contributions is contract execution. Specifically, an enforcement free method for exchange is developed. Such methods are particularly important among computational agents that can vanish easily, and the connection between the agent and the real world party it represents is often hard to detect. The developed method is based on splitting the exchange into partial exchanges such that at any point in the exchange, the remaining benefits from completing the exchange outweigh the benefits from defecting for both agents. A formal equilibrium analysis is carried out proving the viability of the developed method. The analysis includes the following parts.

- Analysis of exchanging countable goods. Strategies that form an equilibrium and complete the exchange in a minimal number of steps are presented. The maximum size delivery that the supplier can safely make at any point in the exchange is shown as well as the maximum amount that the demander can safely pay. Necessary and sufficient conditions for the applicability of the method are formally stated. An inherent problem regarding the completion of the exchange is presented.

- Analysis of exchanging uncountable goods. Strategies that form an equilibrium and complete the exchange in a minimal number of steps are presented. Necessary and sufficient conditions for the applicability of the method are formally stated. The completion problem is less severe with uncountable goods than with countable ones.

- Analysis of non-isolated exchange. The first extension of the work includes studying settings where defection in an exchange can adversely affect an agent's future. Again, step minimizing equilibrium strategies are formally derived for countable and uncountable goods. The non-isolated nature of the exchange often solves the completion problem.

- Incorporating real time and discount functions. This second extension of the basic method presents sufficient conditions under which unenforced exchange is possible when agents discount payments and the value of goods in time. Step minimizing equilibrium strategies are presented. The analysis is carried out for both countable and uncountable goods. It is also theoretically established that under certain conditions, such an exchange will take infinite time to complete. This problem can often be solved by *unenforced* deadlines or lateness penalty schedules in practice.

- Delivery chunking and chunk sequencing. The third extension includes an analysis of settings where the order of delivery within an exchange can be varied. In such settings, two questions have to be answered. First, what chunks should the entire delivery be split into? Some chunkings allow unenforced exchange while others do not. A top down and a bottom up method for chunking are presented, and their relative advantages are compared. Second, in what order should the chunks be delivered? A non-obvious greedy quadratic algorithm for independent chunks is presented that guarantees finding a safe sequence if one exists. It is also proven that the general sequencing problem cannot be solved in polynomial time in the number of chunks when chunks are interdependent.

- Finally, the dissertation discusses how the unenforced exchange method helps avoid unfair renegotiation.

The rest of the dissertation is structured as follows. First, Chapter 2 discusses related research. Then, Chapter 3 presents our work on automated contracting. After that, Chapter 4 discusses our work on coalition formation. Finally, Chapter 5 presents our work on unenforced contract execution, and Chapter 6 concludes and presents future research directions.

CHAPTER 2

RELATED RESEARCH

This chapter reviews relevant research that has been done in related areas. The first sections discuss coalition formation research. Then mechanism design is reviewed—including auctions, social choice, principal-agent problems, and task revelation. After that, bargaining theory, market mechanisms, and non-economic approaches to DAI are overviewed. Finally, work in resource-bounded reasoning is addressed.

2.1 Coalition Formation

One of the three main areas of multiagent systems which this dissertation contributes to is coalition formation. That part of the work, Chapter 4, assumes knowledge of classical coalition formation concepts from Section 2.1.1, and the core solution concept from Section 2.1.2.1. Other solution concepts for coalition formation are presented for comparison in Sections 2.1.2.2—2.1.2.11.

2.1.1 Coalition Formation Settings

In many domains, self-interested real world parties—e.g. companies—can save costs by coordinating their activities with other parties. In negotiations among such agents, the question of coordination arises: what coalitions should the agents form, are they stable, and how should costs be divided among agents within each coalition? Coalition formation includes three activities. One is *coalition structure generation*: formation of coalitions by the agents such that agents within each coalition coordinate their activities, but agents do not coordinate between coalitions. Precisely this means partitioning the set of agents into exhaustive and disjoint coalitions. This partition is called a coalition structure (CS). The second is the *solving of the (optimization)* *problem* of each coalition. In task allocation problems this involves deciding how to distribute the tasks of the coalition among the member agents and solving the optimization problem of each agent given its resources and the tasks it was distributed. The coalition's objective is to maximize monetary value: money received from outside the system for accomplishing tasks minus the cost of using resources.[1] Third, agents within each coalition have to agree on how to *divide this value* of the generated solution. These activities interact. For example, the coalition that an agent wants to join depends on the portion of the value that the agent would be allocated in each potential coalition.

Coalition formation has been widely studied [Kahan and Rapoport, 1984, van der Linden and Verbeek, 1985, Raiffa, 1982, Shechory and Kraus, 1995a, Zlotkin and Rosenschein, 1994, Ketchpel, 1994], but to my knowledge, only among rational agents. Let us call the entire set of agents A. Say, that the lowest cost achievable by agents $S \subseteq A$ working together, but without any other agents, is c_S^R. This is the minimum cost to handle the tasks of agents S with the resources of agents S. A coalition game in characteristic function form—i.e. a characteristic function game (CFG)—is defined by a characteristic function v_S^R, which defines the *value* of each coalition S:

$$v_S^R = -c_S^R. \tag{2.1}$$

The superscript R emphasizes that we mean the *rational* value of the coalition, i.e. the maximum value that is reachable by the coalition given its optimization problem. A rational agent can solve this combinatorial problem optimally without any deliberation costs such as CPU time costs or time delay costs. In CFGs, the value of a coalition only depends on its members. The value is not affected by actions of nonmembers. Games where nonmembers' actions affect the value of a coalition are discussed later starting in Section 2.1.2.9.

[1] In some problems, not all tasks have to be handled. This can be incorporated by associating a cost with each omitted task. Then problem solving also involves the selection of tasks to handle.

In game theory, games where agents can make binding deals regarding who is going to coordinate with who and how, are called *cooperative games*. This does not mean that the agents are cooperative—they are still self-interested. The term is used as an antonym to *noncooperative games* where agents cannot make binding deals. Actually, cooperative games are a proper subset of noncooperative ones. The binding deals can be thought of as enlarging the strategy spaces of the agents in a noncooperative game. So even though the deals themselves are binding, the agents cannot bindingly coordinate their activities regarding what deals they are going to make.

Any outcome of a game can be analyzed with respect to *social welfare*, which is defined as the sum of the agents' payoffs. The payoff that agent i gets is denoted by $x_i \in \Re$. The sum of the agents' x_i's has to equal the sum of the values of the coalitions in the coalition structure (CS) that formed: no wealth is generated from nothing and no wealth disappears. In the setting of this dissertation, the agents can make side payments to each other.

A game is *superadditive* if the value of one coalition plus the value of another coalition is never more than the value of these coalitions joined into one coalition (Fig. 4.2):

Definition 2.1 *A game is* **superadditive** *if* $(\forall S, T \subseteq A, S \cap T = \emptyset), v^R_{S \cup T} \geq v^R_S + v^R_T$.

When computation cost is ignored, this is almost always the case, because at worst, the agents in the composite coalition can use the solutions that they had when they were in separate coalitions. A game can be non-superadditive only if the collusion process itself involves some cost, e.g. anti-trust penalties. All superadditive games are *grand coalition games*, i.e. the agents are best off—from a social welfare viewpoint—by forming the grand coalition ($CS^{R*} = \{A\}$). Some non-superadditive games are *subadditive*, Fig. 4.2:

Definition 2.2 *A game is* **subadditive** *if* $(\forall S, T \subseteq A, S \cap T = \emptyset), v^R_{S \cup T} < v^R_S + v^R_T$.

In subadditive games, the agents are best off by operating alone, i.e. $CS^{R*} = \{\{a_1\}, \{a_2\}, ..., \{a_{|A|}\}\}$. Some games are neither superadditive nor subadditive, because the characteristic function fulfills the condition of superadditivity for some coalitions and the condition of subadditivity for others. In such cases, the social welfare maximizing coalition structure varies.

There are many solution concepts for coalition games. Most of them are targeted towards guaranteeing some form of stability. Some of them also satisfy efficiency criteria. The next sections describe the most important solution concepts in more detail. First solution concepts for CFGs are discussed, and then ones for non-CFGs.

2.1.2 Solution Concepts for Coalition Games

2.1.2.1 Core

Is the current coalition structure one of the social welfare maximizing ones (group rationality)? Can the social good be distributed among the agents so that each agent is motivated to stay with the social welfare maximizing coalition structure CS^{R*} (individual rationality)? Furthermore, can it be distributed so that every subgroup of agents is better off with CS^{R*} than by forming a coalition of their own (coalition rationality)? The *core* (C) is the solution concept that satisfies all of these conditions [Kahan and Rapoport, 1984, van der Linden and Verbeek, 1985, Raiffa, 1982]. The core of a game is a set of payoff configurations $(\vec{x}, CS^{R*})$, where each $\vec{x}$ is a vector of payoffs to the agents in such a manner that no subgroup (individual agents and the group of all agents are also subgroups) is motivated to depart from a CS^{R*}. Given payoffs according to $\vec{x}$, the value of each subgroup is less than or equal to the sum of the payoffs that the agents of that subgroup get under CS^{R*}. Obviously, only CSs that maximize welfare can be stable in the sense of the core, because from any other CS the group of all agents would prefer to switch to a CS^{R*}. Formally,

Definition 2.3 Core $C = \{(\vec{x}, CS^{R*}) | \forall S \subset A, \sum_{i \in S} x_i \geq v_S^R$ *and* $\sum_{i \in A} x_i = \sum_{j \in CS^{R*}} v_{S_j}^R\}$.

The core is the strongest of the classical solution concepts in coalition formation. It is often too strong: in many cases it is empty, i.e. the social good cannot be divided so that the individual and coalition rationality conditions are satisfied [Kahan and Rapoport, 1984, van der Linden and Verbeek, 1985, Raiffa, 1982, Zlotkin and Rosenschein, 1994]. Games with non-empty cores are called *weak*, Fig. 4.2. In games where the core is empty, any solution is prone to deviation by some subgroup of agents. The new solution, that is acquired by the deviation, is again prone to deviation and so on. There will be an infinite sequence of steps from solution to solution. To avoid this, explicit mechanisms such as limits on negotiation rounds, contract costs or some social norms need to be in place in the negotiation setting.

A lesser problem is that the core may include multiple payoff vectors and the agents have to agree on one of them. An often used solution is to pick the *nucleolus* which is, intuitively speaking, the center of the core [Kahan and Rapoport, 1984, van der Linden and Verbeek, 1985, Raiffa, 1982]. The nucleolus is discussed later in Section 2.1.2.5.

The *strong* ϵ*-core* $C_{strong}(\epsilon)$ is a solution concept related to the core where the constraints $\sum_{i \in S} x_i \geq v_S^R$ are weakened to $\sum_{i \in S} x_i \geq v_S^R - \epsilon$. This provides a solution concept for coreless games by choosing a sufficiently large positive ϵ. It can be interpreted as there being a fixed cost ϵ for a coalition to complain about the solution. In another solution concept called the *weak* ϵ*-core* $C_{weak}(\epsilon)$, this complaining cost is assumed to be directly proportional to the number of agents in the coalition. Therefore the conditions $\sum_{i \in S} x_i \geq v_S^R - |S| \cdot \epsilon$ are used. Clearly, $C \subset C_{strong} \subset C_{weak}(\epsilon)$ for any positive ϵ. Both ϵ-cores can also be used to decide on a payoff distribution vector in games where the core includes many such vectors. This can be done by choosing a sufficiently small negative ϵ such that the ϵ-core becomes a point. Although the

ϵ-cores seem like solutions to the nonexistence and nonuniqueness of the core, they do not satisfy coalition rationality when the core itself is empty. Furthermore, they are not motivated by the dynamics of the negotiation process.

Another problem with the core and the ϵ-cores is that the constraints in the definitions become numerous as the number of agents increases. This is due to the subset operator in the definitions.

The core is used as a solution concept in Chapter 4 of this dissertation. Alternative solution concepts are discussed in the following sections.

2.1.2.2 Stable Sets

Another solution concept for CFGs that attempts to remedy the nonexistence problem of the core is *stable sets* [von Neumann and Morgenstein, 1947, Kahan and Rapoport, 1984]. This solution concept keeps individual and group rationality constraints but relaxes the requirement of coalition rationality. Let us say that a solution $(\vec{x}, CS)$ dominates $(\vec{x}', CS')$ if there exists some coalition $S \subseteq A$ such that $x_i > x_i'$ for all $i \in S$, and $v_S^R \geq \sum_{i \in S} x_i$. This binary relation, dominance, is not transitive. Now, V is defined to be a stable set iff

- if $(\vec{y}, CS_0)$ and $(\vec{y}', CS_1)$ are in V, then neither of them dominates the other, and
- if $(\vec{y}'', CS_2)$ is not in V, then there is some $(\vec{y}''', CS_3)$ in V that dominates $(\vec{y}'', CS_2)$.

There may be many stable sets for a given game instance. If the core is non-empty, it is a subset of every stable set. The number of stable sets and the number of constraints characterizing those stable sets increases rapidly with the number of agents. Another problem with this solution concept is that for some games the stable sets are empty. From the very motivation of the stable sets it also follows that this solution concept does not satisfy coalition rationality.

2.1.2.3 Bargaining Set

The *bargaining set* is a solution concept for CFGs that attempts to remedy the nonexistence problem of the core and the nonuniqueness and computational complexity of the stable sets. It assumes individual rationality but relaxes coalition and group rationality. The bargaining set is based on threat analysis. An individually rational payoff configuration $(\vec{x}, CS)$ is considered to be stable—i.e. in the bargaining set—if any threat to deviate from $(\vec{x}, CS)$ can be answered by a counterthreat [Kahan and Rapoport, 1984]. Say that agents k and l are members of the same coalition $S_j \in CS$. Agent k can launch an objection to S_j by pointing out a better coalition Y for itself (and a payoff distribution y within Y) whenever

- $k, l \in S_j$, $k \in Y$, $l \notin Y$, and
- $\sum_{i \in Y} y_i = v_Y^R$, and
- $y_k > x_k$, and $\forall i \in Y, y_i \geq x_i$.

On the other hand, agent l can launch a counterobjection by pointing out a better coalition Z for itself (and a payoff distribution z within Z) whenever

- $l \in Z$, $k \notin Z$, and
- $\sum_{i \in Z} z_i = v_Z^R$, and
- $\forall i \in Z, z_i \geq x_i$, and $\forall i \in (Y \cap Z), z_i \geq y_i$.

An objection is called justified if there is no possible counterobjection. The bargaining set is defined to be the set of individually rational payoff configurations $(\vec{x}, CS)$ in which no agent has a justified objection against another agent within the same coalition.

The bargaining set is always nonempty which makes it a potential solution concept for any CFG. Furthermore, in games where the core is nonempty, the core is always a subset of the bargaining set: in the core no justified objections exist because no objections exist at all.

Unfortunately there is a bargaining set for every coalition structure, so this solution concept cannot be used in coalition structure generation to decide among coalition structures: the negotiation is viewed as occurring over the payoff distribution $\vec{x}$ within a given CS. Furthermore, the bargaining set is often not a unique point even within a given coalition structure.

There have been multiple extensions to the bargaining set concept in order to restrict the space of outcomes within the solution concept [Kahan and Rapoport, 1984]. The first extension considers only payoff configurations that are rational for all subgroups within each coalition of the coalition structure instead of all individually rational payoff configurations. The second extension allows objections and counter-objections by subgroups—either subgroups of current coalitions or any subgroups. Furthermore, the first and the second extension can be combined. The extended solution concepts unfortunately do not guarantee nonemptyness.

The *competitive bargaining set* is an extension of the bargaining set where the objector can launch multiple alternative objections simultaneously, and the counter-objector has to satisfy the agents of each of the objections by a single counterobjection. Obviously the competitive bargaining set is a subset of the bargaining set. Unfortunately the competitive bargaining set is empty for some games.

The *ϵ-bargaining set* is another extension where the inequalities of the objection have to have a difference of at least ϵ. This corresponds to there being some bargaining cost in switching the coalition structure. The bargaining set is a subset of the ϵ-bargaining set.

Psychological extensions of the bargaining set include the *modified bargaining set* and the *power bargaining set* [Kahan and Rapoport, 1984].

2.1.2.4 Kernel

Unlike the previous solution concepts which considered stability based on different types of threats, the *kernel* is based on a notion of fairly distributing excesses [Kahan and Rapoport, 1984]. Let us define the excess of a coalition $S \in CS$ in a payoff configuration $(\vec{x}, CS)$ as

Definition 2.4 Excess $e(S, \vec{x}) = v_S^R - \sum_{i \in S} x_i$.

The kernel is based on balancing excesses within each coalition. Let the maximum surplus of player k over player l in $(\vec{x}, CS)$ be

Definition 2.5 Maximum surplus $s_{kl} = max_{S|k \in S, l \notin S} e(S, \vec{x})$.

The players k and l are said to be in a certain type of equilibrium when

- $s_{kl} = s_{lk}$, or
- $s_{kl} > s_{lk}$ and $x_l = v_{\{l\}}^R$, or
- $s_{lk} > s_{kl}$ and $x_k = v_{\{k\}}^R$.

The kernel is the set of payoff configurations where all players within each coalition are in this type of equilibrium. Formally,

Definition 2.6 Kernel $= \{(\vec{x}, CS) | \forall S \in CS, \forall k, l \in S, k \ \textit{and} \ l \ \textit{in equilibrium}\}$.

For any CFG, the kernel is nonempty for every coalition structure. Furthermore, it is always a subset of the bargaining set. The kernel fulfills the desirable characteristics of giving symmetric players equal payoff and more desirable players more payoff than to less desirable ones. The kernel is not a solution concept that models a bargaining process. It can rather be viewed as a criterion of fairness in an arbitration scheme.

2.1.2.5 Nucleolus

Another solution concept for CFGs based on a notion of fairness is the *nucleolus* [Kahan and Rapoport, 1984, Lundgren *et al.*, 1992]. Instead of pairwise comparisons of excesses of the agents, it considers the excesses of each of the $2^{|A|}$ possible coalitions for a given coalition structure CS and payoff vector $\vec{x}$. The nucleolus is defined to be the set of payoff configurations $(\vec{x}, CS)$ where the $2^{|A|}$ excesses are lexicographically minimal for $\vec{x}$ when sorted in nonincreasing order for a given CS.

The nucleolus is nonempty and unique for each coalition structure in any characteristic function game. It is a subset of the kernel as well as the core (whenever the latter is nonempty). Therefore, it inherits the desirable properties of symmetry and desirability from the kernel, and satisfies individual, coalition, and group rationality whenever the core is nonempty.

2.1.2.6 Equal Share Analysis

Yet another solution concept for CFGs based on fairness considerations more than stability is *equal share analysis* [Kahan and Rapoport, 1984]. A payoff configuration is within this solution concept if three conditions hold. First, no two coalitions in the coalition structure would be better off by joining. Second, no coalition can divide its value equally such that each member gets more than under the current payoff vector. Third, stronger members receive payoffs at least equal to those of weaker members. Every CFG has a nonempty set of payoff configurations satisfying this solution concept for some coalition structures. Unfortunately the solution of some CFGs may be empty for all Pareto efficient coalition structures.

2.1.2.7 Equal Excess Theory

Equal excess theory presents an iterative scheme for CFGs for arriving at a payoff configuration. The scheme is prescribed to the agents—there is no guarantee that a self-interested agent is best off using this particular scheme. The agents start with equal division of payoff and update their expectations of what they would get for payoff in each round. If at some round all agents for a given coalition view that coalition as their best alternative, then that coalition forms on the round. In games examined to date, the scheme has always converged although a convergence proof has not been presented [Kahan and Rapoport, 1984]. Equal excess theory addresses both the coalition structure formation problem and the payoff division problem.

2.1.2.8 Shapley Value

The *Shapley value* solution concept for CFGs is based on an axiomatic approach [Kahan and Rapoport, 1984, Raiffa, 1982, Zlotkin and Rosenschein, 1994]. Five desirable properties—axioms—for a solution concept are postulated, and then it is shown that the Shapley value is the only solution concept that satisfies them. This solution concept is not a model of stability in a bargaining process. Instead it can be used as a basis of fairness in an arbitration scheme.

The first axiom states that an agent's payoff depends on the characteristic function and the coalition structure only. The second axiom says that the agent's labeling in the characteristic function does not affect its payoff. The third axiom states that the sum of the agents' payoffs within a coalition equals the value of the coalition. The fourth axiom states that if an agent's participation does not change the value of any coalition, then that agent's payoff is zero. The fifth and final axiom states that a player's payoff in a composite game is the sum of its payoffs in component games of the composite game. These axioms define a unique payoff vector for a coalition structure. An element of this vector is an agent's Shapley value.

The Shapley value can be interpreted via a coalition accumulation argument. An agent's contribution to its coalition depends on which other agents have joined before it and which ones will join after it. An agent's Shapley value is the agent's contribution to its coalition averaged over all possible joining orders.

The Shapley value payoff vector is unique and existing for each coalition structure for any CFG. It is guaranteed to satisfy individual rationality in superadditive games. It does not necessarily satisfy coalition rationality, and need not be a member of the core even if the latter is nonempty. The solution concept satisfies the desirable property of symmetry among agents, and gives more desirable agents higher payoff than less desirable ones.

2.1.2.9 Nash Equilibrium

In general, the value of a coalition may depend on the actions of nonmember agents due to positive and negative interactions of the agents' solutions. Such settings can be modeled as normal form games (NFGs), Fig. 4.2. The coalition formation solution concepts presented so far in Section 2.1 are only applicable in characteristic function games (CFGs), where the value of each coalition S is given by the characteristic function v_S^R, and is thus not a function of the actions of nonmembers. CFGs are a strict subset of NFGs. Because superadditivity, subadditivity, core, and the other solution concepts are undefined in non-CFGs, different solution concepts are necessary. One alternative is the *Nash equilibrium* [Nash, 1950b, Kreps, 1990], which was discussed in Section 1.2.4. It guarantees stability in the sense that no agent alone is motivated to deviate from the solution by changing its strategy given that others in the game do not deviate. Note that the Nash equilibrium is a solution concept in the space of strategies while the solution concepts introduced in Sections 2.1.2.1— 2.1.2.8 where in the space of payoff configurations. The Nash equilibrium concept comes from noncooperative game theory while the latter come from cooperative game theory.

2.1.2.10 Strong Nash Equilibrium

Often the Nash equilibrium solution concept is too weak because subgroups of agents can deviate in a coordinated manner. The *Strong Nash equilibrium* [Aumann, 1959] is a solution concept for NFGs that guarantees more stability. It requires that there is no subgroup that can deviate by changing their strategies jointly in a manner that increases the payoff of all of its members given that nonmembers do not deviate from the original solution. The Strong Nash equilibrium is often too strong a solution concept because in many games no such equilibria exist.

2.1.2.11 Coalition-Proof Nash Equilibrium

Recently, the *Coalition-Proof Nash equilibrium* [Bernheim *et al.*, 1987, Bernheim and Whinston, 1987] solution concept for NFGs has been suggested as a partial remedy to the nonexistence problem of the Strong Nash equilibrium. This solution concept requires that there is no subgroup that can make a mutually beneficial deviation (keeping the strategies of nonmembers fixed) *in a way that the deviation itself is stable according to the same criterion.* A conceptual problem with this solution concept is that the deviation may be stable within the deviating group, but the solution concept ignores the possibility that some of the agents that deviated may prefer to deviate again with agents that did not originally deviate. Furthermore, even this kinds of solutions do not exist in all NFGs.

Clearly, there is room for further research on coalition formation solution concepts—even among agents that are not computationally limited. On the other hand, computational limitations significantly change the terrain. Existing computational coalition formation methods are discussed in the next section. In Chapter 4 of this dissertation, I present new normative results for coalition formation under costly computation.

2.1.3 Computational Coalition Formation

So far static solution concepts for coalition formation were presented. They address the question of how to divide the payoffs among agents. Some of them also address the question of which coalition structure should form. But being static in nature, they do not address the dynamics of the coalition formation process. In this section we will discuss work that has addressed the dynamics.

Friend [Friend, 1973, Kahan and Rapoport, 1984] has developed a program that simulates a 3-agent coalition formation situation where agents can make offers, acceptances, and rejections to each other regarding coalitions and payoffs. In the model, at most one offer regarding each agent can be active at a time. A new offer makes old offers regarding that agent void. Players consider only current proposals: there is no lookahead or memory. The negotiations terminate when two agents have reach a dyad and the third one has given up. Specifically, the termination criterion is based on a local threat-counterthreat examination: an agent does not necessarily accept a new better offer if that introduces a risk of being totally excluded in the next step. The model is purely descriptive. There is no guarantee that a self-interested agent would be best off by using the specified local strategies.

Transfer schemes are another dynamic approach to coalition formation [Kahan and Rapoport, 1984]. The agents stay within a given coalition structure and exchange payments within each coalition according to demands computed in a prespecified manner. So, transfer schemes address the payoff distribution problem but not coalition structure generation. A transfer scheme that provably converges to a payoff configuration in the bargaining set from any initial payoff configuration in any CFG has been developed. At each iteration, if an agent is faced with a justified objection from another agent, the former has to pay the latter the minimal amount required to make the objection unjustified. If an agent is faced with multiple objections, it caters only to the largest one on an iteration. When the bargaining set is not a point, the exact final payoff configuration depends on the starting point. A transfer scheme for the kernel has also been developed. There the agents within a coalition exchange payments at each iteration so as to diminish the difference of their mutual maximum surpluses. Furthermore, a transfer scheme for the core has been developed. This scheme alternates between two operators. In the first, an agent's payoff is incremented by its coalition's excess divided by the number of agents in the coalition.

In the second, every agent's payoff is decremented equally, and just enough to keep the total payoff vector feasible. The method can be implemented in a largest-excess-first fashion, or in a round-robin fashion among agents. Both schemes converge to the core if it is nonempty. All of the transfer schemes assume that the agents know the values of the characteristic function.

Zlotkin and Rosenschein [Zlotkin and Rosenschein, 1994] analyze coalition formation among rational agents that cannot make side payments, while our agents do. Their analysis is limited to "Subadditive Task Oriented Domains" (STODs), which are a strict subset of CFGs, Fig. 4.2. In their solution concept, one agent handles all the tasks. In STODs this is optimal because STODs never exhibit diseconomies of scale. We do not assume that one agent can take care of all the agents' tasks. Unlike our work, they also assume that all agents have the same capabilities (symmetric cost functions for task sets). Their method guarantees each agent an expected value that equals its Shapley value [Kahan and Rapoport, 1984, Raiffa, 1982]. The Shapley value motivates individual agents to stay with the coalition structure (individual rationality) and the group of all agents to stay (group rationality). Unlike the core, the Shapley value does not in general motivate every subgroup of agents to stay with the coalition structure (coalition rationality). In a subset of STODs, "Concave Task Oriented Domains" (Fig. 4.2), the Shapley value also satisfies coalition rationality, i.e. the vector of Shapley value payoffs is in the core.

A naive method that guarantees an expected value equal to the Shapley value has exponential complexity in the number of agents, but Zlotkin and Rosenschein present a novel cryptographic method for achieving this with linear complexity in the number of agents. Yet each one of these linearly many problems involving the agents' tasks needs to be solved optimally. In combinatorial problems such as the vehicle routing problem of this dissertation—and the Postmen Domain of Zlotkin and Rosenschein for that matter—this is clearly intractable if the problem instances are large.

Ketchpel [Ketchpel, 1994] presents a coalition formation method for rational agents which have different expectations of coalition values. The (computational) origin of these expectations is not addressed. His assumption of imperfect information differs from our setting, where the agents have perfect information, but cannot perfectly deduce. The method differs from methods presented so far in that it addresses coalition structure generation as well as payoff distribution. These two activities are handled simultaneously. Ketchpel's coalition formation algorithm runs in cubic time in the number of agents, but does not guarantee stability. His protocol is based on mutual offers. In practice it is hard to prevent out-of-protocol offers such as multiagent offers. Finally, his 2-agent auction is manipulable and computationally inefficient.

This is closely related to a contracting protocol of mine [Sandholm, 1993] (TRACONET), where agents construct the global solution by contracting a small number of tasks at a time, and payments are made regarding each contract before new contracts take place. An agent updates its approximate solution after each task transfer. My contributions to contracting schemes are discussed in detail in Chapter 3.

In general equilibrium market mechanisms (Section 2.4) such as WALRAS [Wellman, 1992], non-manipulative agents iterate over the allocation of resources and tasks, and payments are made only after a final solution is reached.

Shechory and Kraus [Shechory and Kraus, 1995a] analyze coalition formation among rational agents with perfect information in domains that are not necessarily superadditive. Their protocol guarantees that if agents follow it, a certain stability criterion (K-stability) is met. This requires the solution of an exponential number of optimization problems. Their other protocol guarantees a weaker form of stability (polynomial K-stability), but only requires the solution of a polynomial number of optimization problems. Unfortunately, each one of these may be intractable. Their algorithm switches from one coalition structure to another guaranteeing improvements at each step: coalition structure formation is an anytime algorithm, although each domain problem is solved optimally. On the other hand, in the approach of this dissertation, each domain problem is solved using an approximation (design-to-time) algorithm. A variant of a computational coalition formation scheme has been used in the database domain [Klusch and Shechory, 1996].

Recently Shechory and Kraus [Shechory and Kraus, 1995b] presented an algorithm for coalition structure generation among cooperative—social welfare maximizing, i.e. not self-interested—agents. Among such agents the payoff distribution is a non-issue and is thus not addressed. The distributed algorithm forms disjoint coalitions—which by their definition can only handle one task each—and allocates tasks to the coalitions. The complexity of the problem is reduced by restricting (possibly compromising optimality) the number of agents in a coalition. The greedy algorithm guarantees that the solution is within a ratio bound from the best solution that is possible given the restriction on the number of agents. The work assumes that the domain problem of each coalition can be solved optimally and without cost, which is not the case in the combinatorial problems of this dissertation. Second, in this dissertation, a coalition can handle more than one task.

2.2 Mechanism Design

In Section 2.1 solution concepts and dynamic schemes for coalition formation were discussed. Another area of game theory called *mechanism design* explores interaction mechanisms among agents. The goal is to generate protocols such that when agents use them according to some stability solution concept—e.g. dominant strategies [Muller and Satterwaithe, 1985, Fudenberg and Tirole, 1991, Kreps, 1990, Rasmusen, 1989] (Section 1.2.4), Nash equilibrium or its refinements [Maskin, 1985, Postlewaite, 1985, Fudenberg and Tirole, 1991, Kreps, 1990, Rasmusen, 1989], or other [Myerson, 1985, Fudenberg and Tirole, 1991, Kreps, 1990, Rasmusen, 1989]—then desirable social outcomes follow. The approach is normative. Local strategies are not externally chosen for the agents, but instead each agent uses the strategy that is best for itself.

The *revelation principle* is an important result within mechanism design. For Nash equilibrium implementation it states that any outcomes that can be achieved in Nash equilibrium using any mechanism that is representable as a finite game tree (with possibly moves by nature) can be achieved in Nash equilibrium using a direct revelation mechanism [Kreps, 1990, Maskin, 1985, Postlewaite, 1985]. A direct revelation mechanism is a mechanism where the agents simultaneously and at once reveal their types, i.e. all of their private information. Potential problems include the difficulty of modeling some free form negotiation situations as a game tree, the players' difficulty of computing the Nash equilibrium in a complex mechanism (leading to non-Nash play among computationally limited agents and human agents in practice), and the nonuniqueness of a Nash equilibrium [Kreps, 1990].

The revelation principle for dominant strategy mechanisms states that the outcome of any dominant strategy mechanism can be achieved via a direct revelation mechanism where participation and truthtelling are dominant strategies [Kreps, 1990, Muller and Satterwaithe, 1985].

In the following subsections, mechanism design is discussed in specific settings.

2.2.1 Auction Theory

Auction theory analyzes protocols and agents' strategies in auctions. An auction consists of an auctioneer and potential bidders. Auctions are usually discussed in situations where the auctioneer wants to sell an item and get the highest possible payment for it while the bidders want to acquire the item at the lowest possible price. The discussion of this section will pertain to the classical setting, although in the contracting setting of this dissertation, the auctioneer wants to subcontract out tasks at the lowest possible price while the bidders who handle the tasks want to receive the highest possible payment for doing so. The mechanisms for the latter setting are totally analogous to mechanisms for the former.

2.2.1.1 Auction Settings

There are three qualitatively different auction settings depending on how an agent's value (monetary equivalent of utility) of the item is formed.

In *private value* auctions, the value of the good depends only on the agent's own preferences. An example is auctioning off a cake that the winning bidder will eat. The key is that the winning bidder will not resell the item in which case the value would depend on other agents' valuations (a valuation is the monetary equivalent of expected utility). The agent is often assumed to know its value for the good exactly.

On the other hand, in *common value* auctions, an agent's value of an item depends entirely on other agents' values of it, which are identical to the agent's by symmetry of this criterion. For example, auctioning treasury bills fulfills this criterion. Nobody inherently prefers having the bills, and the value of the bill comes entirely from reselling possibilities.

In *correlated value* auctions, an agent's value depends partly on its own preferences and partly on others' values. For example, a negotiation within the contracting setting of this dissertation fulfills this criterion. An agent may handle a task itself in which case the agent's local concerns define the cost of handling the task. On the other hand, the agent can recontract out the task in which case the cost depends solely on other agents' valuations.

The next section discusses different auction protocols.

2.2.1.2 Auction Protocols

In the *English (first-price open-cry) auction*, each bidder is free to raise his bid. When no bidder is willing to raise anymore, the auction ends, and the highest bidder wins the item at the price of his bid. An agent's strategy is is a series of bids as a function of his private value, his prior estimates of other bidder's valuations, and the past bids of others. In private value English auctions, an agent's dominant strategy is to always bid a small amount more than the current highest bid, and stop when his private value price is reached. In correlated value auctions the rules are often varied to make the auctioneer increase the price at a constant rate or at a rate he thinks appropriate. Secondly, sometimes *open-exit* is used where a bidder has to openly declare exiting without a re-entering possibility. This provides the other bidders more information regarding the agent's valuation.

In the *first-price sealed-bid auction*, each bidder submits one bid without knowing the others' bids. The highest bidder wins the item and pays the amount of his bid. An agent's strategy is his bid as a function of his private value and prior beliefs of others' valuations. In general there is no dominant strategy for acting in this auction. With common knowledge assumptions regarding the probability distributions of the agents' values, it is possible to determine Nash equilibrium strategies for the agents [Rasmusen, 1989].

In the *Dutch (descending) auction*, the seller continuously lowers the price until one of the bidders takes the item at the current price. The Dutch auction is strategically equivalent to the first-price sealed-bid auction, because in both games, an agent's bid matters only if it is the highest, and no relevant information is revealed during the auction process.

In the *Vickrey (second-price sealed-bid) auction*, each bidder submits one bid without knowing the others' bids. The highest bidder wins, but at the price of the second highest bid [Vickrey, 1961, Milgrom, 1985, Rasmusen, 1989]. An agent's strategy is his bid as a function of his private value and prior beliefs of others' valuations. The dominant strategy in private value Vickrey auctions is to bid one's true valuation. If an agent bids more than that, and the increment made the difference between winning or not, he will end up with a loss if he wins. If he bids less, there is a smaller chance of winning, but the winning price is unaffected [2]. The dominant strategy result of Vickrey auctions means that an agent is best off bidding truthfully no matter what the other bidders are like: what are their capabilities, operating environments, bidding plans, etc. This has two desirable sides. First, the agents reveal their preferences truthfully which allows globally efficient decisions to be made. Second, the agents need not waste effort in counterspeculating other agents, because they do not matter in making the bidding decision. Vickrey auctions have been widely advocated and adopted for use in computational multiagent systems [Sun Microsystems, 1996, Agorics, Inc., 1996, Edelman, 1996, Huberman and Clearwater, 1995, Waldspurger *et al.*, 1992, Drexler and Miller, 1988, MacKie-Mason and Varian, 1993, MacKie-Mason and Varian, 1994, Smart Market, 1996, Rosenschein and Zlotkin, 1994]. For example, versions of the Vickrey auction have been used to allocate computation resources in operating systems [Waldspurger *et al.*, 1992, Drexler and Miller, 1988], to allocate bandwidth in computer networks [Sun Microsystems, 1996, MacKie-Mason and Varian, 1993, MacKie-Mason and Varian, 1994, Smart Market, 1996], and to computationally control building environments [Huberman and Clearwater, 1995]. On the other hand, Vickrey auctions have not been widely adopted in auctions among humans [Rothkopf and Harstad, 1995, Rothkopf *et al.*, 1990] even though the protocol was laid out 25 years ago [Vickrey, 1961]. Limitations of the Vickrey auction protocol—especially in computational multiagent systems—have been discussed by Sandholm in [Sandholm, 1996].

All-pay auctions are another family of auction protocols. In such mechanisms, all participating bidders have to pay something to the auctioneer. The schemes have been used in computational multiagent systems for tool reallocation [Lenting and

[2] In private value auctions, the Vickrey auction is strategically equivalent to the English auction. They will produce the same allocation at the same prices. On the other hand, in correlated value auction, the other agents' bids in the English auction provide information to the agent about its own valuation. Therefore English and Vickrey auctions are not strategically equivalent in general, and may lead to different results.

Braspenning, 1994]. These methods are often susceptible to infinite escalations of bids [Raiffa, 1982], and will not be discussed further in this dissertation.

2.2.1.3 Revenue Equivalence and Non-Equivalence

When considering one auction in isolation, each one of the four auction protocols (English, Dutch, first-price sealed-bid, and Vickrey) allocates the auctioned item Pareto efficiently to the bidder who values it the most. One would imagine that the first-price auctions give higher expected revenue to the auctioneer because in second-price auctions the auctioneer only gets the second price. This is not the case however, because in first-price auctions the bidders are motivated to lie by biasing their bids downward. The *revenue-equivalence theorem* [Vickrey, 1961, Milgrom, 1985, Rasmusen, 1989] states that all four auction protocols produce the same expected revenue to the auctioneer in private value auctions where the values are independently distributed. Although all four are Pareto efficient in the allocation, the ones with dominant strategies (Vickrey auction and English auction) are more efficient in the sense that no effort needs to be wasted in counterspeculating the other bidders.

However, most auctions are not pure private value auctions as discussed earlier. In correlated value auctions with at least three bidders, the open-exit English auction leads to higher revenue than the Vickrey auction. The reason is that other bidders willing to go high up in price causes a bidder to up its own valuation of the auctioned item. In this type of auctions, both English and Vickrey auction protocols produce greater revenue to the auctioneer than the first-price sealed-bid auction—or its equivalent, the Dutch auction.

2.2.1.4 Bidder Collusion

One problem with all four of the auction mechanisms (English auction, Dutch auction, first-price sealed-bid auction, and Vickrey auction) is that they are not collusion proof. The bidders could coordinate their bid prices so that the bids stay artificially low. In this manner, the bidders get the item at a lower price than they would without colluding.

The English auction and the Vickrey auction actually self-enforce some of the most likely collusion agreements. Therefore, from the perspective of deterring collusion, the first-price sealed-bid and the Dutch auctions are preferable. The following example from [Rasmusen, 1989] shows this. Let bidder Smith have value 20, and every other bidder have value 18 for the auctioned item. Say that the bidders collude by deciding that Smith will bid 6, and everyone else will bid 5. In an English auction this is self-enforcing, because if one of the other agents exceeds 5, Smith will observe this, and be willing to go all the way up to 20, and the cheater will not gain anything from breaking the coalition agreement. In the Vickrey auction, the collusion agreement can just as well be that Smith bids 20, because Smith will get the item for 5 anyway. Bidding 20 removes the incentive from any bidder to break the coalition agreement by bidding between 5 and 18, because no such bid would win the auction. On the other hand, in a first-price sealed-bid auction, if Smith bids anything below 18, the other agents have an incentive to bid higher than Smith's bid and to win the contract. The same holds for the Dutch auction.

However, for collusion to occur under the Vickrey auction, the first-price sealed-bid auction, or the Dutch auction, the bidders need to identify each other before the submission of bids—otherwise a non-member of the coalition could win the contract. On the other hand, in the English auction this is not necessary, because the bidders identify themselves by shouting bids. An auctioneer can organize a computerized English auction where the bidding process does not reveal the identities of the bidders.

2.2.1.5 Lying Auctioneer

The insincerity of the auctioneer may be a problem in the Vickrey auction. The auctioneer may overstate the second highest bid to the highest bidder unless that bidder can verify it. An overstated second offer would give the bidder a higher bill than it would receive if the contractor were truthful. In other words, the theory classically assumes a truthful auctioneer. Alternatively, cryptographic electronic signatures could perhaps be used by the bidders so that the auctioneer could actually present the second best bid to the winning bidder—and would not be able to alter it.

The other three auction protocols (English, Dutch, and first-price sealed-bid) do not suffer from lying by the auctioneer because the highest bidder gets the item at the price that it stated in the bid.

Cheating by the auctioneer has been suggested to be one of the main reasons why the Vickrey auction protocol has not been widely adopted in auctions among humans [Rothkopf *et al.*, 1990]. In another paper, two formal models of cheating by the auctioneer are discussed [Rothkopf and Harstad, 1995]. The first model is game theoretic. It analyses the situation where the auctioneer can choose to use a first-price sealed-bid protocol or a Vickrey protocol. The bidders' equilibrium behavior creates positive incentives for all auctioneers, except the type most prone to cheat, to choose standard first-price sealed-bid auctions. The second model assumes simple (not rational) bidders. They bid honestly as long as the auctioneer has not been caught cheating, but after catching a cheating auctioneer, the bidders will bid as if the auctioneer always cheats. The result is that a seller with probabilistic opportunities to cheat, and finite abilities to resist cheating, will cheat and be caught in finite time and thereafter have no reason to conduct Vickrey auctions.

2.2.1.6 Lying in Non-Private-Value Auctions

Most auctions are not pure private value auctions: an agent's valuation of a good depends at least in part on the other agents' valuations of that good. For example in contracting settings, a bidder's evaluation of a task is affected by the prices at which the agent can subcontract the task or parts of it out to other agents. This type of recontracting is commonly allowed in automated versions of the contract net protocol also [Sandholm, 1993, Smith, 1980].

Common value (and correlated value) auctions suffer from the *winner's curse*. If an agent bids its valuation and wins the auction, it will know that its valuation was too high because the other agents bid less. Therefore winning the auction amounts to a monetary loss. Knowing this in advance, agents should bid less than their valuations [Milgrom, 1985, Rasmusen, 1989]. This is the best strategy in this type of Vickrey auctions also. So, even though the Vickrey auction promotes truthful bidding in private-value auctions where an agent's valuation is totally determined locally, it fails to induce truthful bidding in most auctions.

2.2.1.7 Undesirable Private Information Revelation

Because the Vickrey auction has truthful bidding as the dominant strategy in private value auctions, agents often bid truthfully. This leads to the bidders revealing their true valuations. Sometimes this information is sensitive, and the bidders would prefer not to reveal it. For example, after winning a contract with a low bid, a company's subcontractors figure out that the company's production cost is low, and therefore the company is making larger profits than the subcontractors thought. It has been observed that when such auction results are revealed, the subcontractors will want to renegotiate their deals to get higher payoff [Rothkopf *et al.*, 1990]. This has been suggested—along with the problem of a lying auctioneer—as one of the main reasons why the Vickrey auction protocol has not been widely adopted in auctions among humans [Rothkopf *et al.*, 1990]. First-price auction protocols do not expose a bidder's valuation as clearly because the bid is subject to strategic lying. Therefore, these auction types may be more desirable than the Vickrey auction when valuations are sensitive.

2.2.2 Social Choice Mechanisms

In an auction—except in an all-pay auction—the mechanism usually leads to a binding contract among only two agents, the auctioneer and the winning bidder. On the other hand, *social choice theory* studies settings where all agents give input to a mechanism, and the outcome that the mechanism chooses based on these inputs is a solution for all of the agents. In most settings, this outcome is enforced so that all agents have to abide to the solution prescribed by the mechanisms. An example social choice rule is majority voting.

The classic goal has been to derive a social choice rule that ranks feasible social outcomes based on individuals' rankings of those outcomes. Let the set of agents be A, and let O be the set of feasible social outcomes. Furthermore, let each agent $i \in A$ have an asymmetric and negatively transitive strict preference relation $\succ_i$ on O. A social choice rule takes as input the agents' preference relations $(\succ_1, ..., \succ_{|A|})$ and produces as output the social preferences denoted by a relation $\succ^*$. Intuitively, the following four properties of a social choice rule seem desirable.

- A social preference ordering $\succ^*$ should exist, be asymmetric and negatively transitive over O for all individual preferences.

- The outcome should be Pareto efficient in the sense that if $\forall i \in A, o \succ_i o'$, then $o \succ^* o'$.

- The scheme should be independent of irrelevant alternatives. Specifically, if $\succ$ and $\succ'$ are arrays of consumer rankings that satisfy $o \succ_i o'$ iff $o \succ'_i o'$ for all i, then the social ranking of o and o' is the same in these two situations.

- No agent should be a dictator in the sense that $o \succ_i o'$ implies $o \succ^* o'$ for all preferences of the other agents.

Arrow's impossibility theorem states that there is no social choice rule that satisfies these four properties [Arrow, 1963, Kreps, 1990, Ordeshook, 1986]. So to design social choice rules, the desiderata have to be relaxed. Commonly the first property is relaxed in the sense that the domain (combinations of individual preferences) on which the rule works is restricted. The third desideratum can also be relaxed—as is done in the majority voting rule.

Due to the impossibility of designing desirable social choice rules, many existing social choice rules can lead to paradoxical results for some combinations of individuals' preferences. When the majority rule is used to pairwise compare alternative social outcomes so that the winner goes on to challenge further alternative outcomes and the loser is eliminated, the *agenda*, i.e. order of the pairings, can totally change the socially chosen outcome [Ordeshook, 1986]. Specifically, in most games the outcome can be any one of the feasible social outcomes, even a Pareto dominated one, i.e. one that is preferred less than some other alternative by all agents. The introduction of irrelevant alternatives also often changes the outcome. If the number of alternative outcomes is large, this type of repeated voting is slow, and an alternative method is often used. The *Borda count* assigns an alternative $|O|$ points whenever it is highest in some agent's preference list, $|O| - 1$ whenever it is second and so on. The alternative with the maximum points is the social choice. The Borda count can also lead to paradoxical results. For example removing the lowest ranked alternative from the set of possible social outcomes can lead to the second worst outcome turning best and the best outcome turning worst in a new Borda count [Ordeshook, 1986].

So far it was assumed that in executing the social choice method, all agents' preferences are known. In reality this is seldom the case. Agents often need to reveal, i.e. declare, their preferences. Assuming knowledge of the preferences is equivalent to assuming that the agents reveal their preferences truthfully. But if an agent can benefit from insincerely declaring its preferences, it will do so. This further complicates the design of social choice mechanisms. Specifically, the *Gibbard-Satterthwaite impossibility theorem* states that if the domain of a social choice function (like a social choice rule but only figures out the highest ranked social outcome instead of a preference ordering) is the space of all combinations of individuals' preferences over

O, $|O| \geq 3$, and the social choice function is implemented by a dominant strategy mechanism, then the function is dictatorial [Gibbard, 1973, Satterthwaite, 1975, Kreps, 1990].

The design of social choice functions is not as impossible as it may seem in the light of the two impossibility theorems. This is because in many settings, the individual preferences are restricted—thus invalidating the conditions of the impossibility theorems. For example, an agent's preferences are often linear in payoff and an agent often does not care what payoffs other agents receive. For this type of settings, it is possible to design non-dictatorial social choice mechanisms. For example the *Groves mechanisms* [Groves and Ledyard, 1977] levy a tax on the voters. The more the voter alters the outcome of the vote, the higher the tax. The taxation mechanism can be setup in a way that truthtelling about one's preferences is the dominant strategy. Therefore the agents need not waste effort in counterspeculating each other's preference declarations. Also, the outcome chosen by the mechanism maximizes social welfare, and therefore also Pareto efficiency. Furthermore, participation in the mechanism can only increase an agent's utility, which makes the mechanism individually rational. The mechanism can be set up so that total tax revenue is positive [Kreps, 1990] or negative [Raiffa, 1982, Rasmusen, 1989], but unfortunately it cannot equal zero. This means that the society's projects need to be funded from outside or that some of the collected tax revenue must be destroyed. It cannot be redistributed into the society of agents. Thus, without an external benefactor, the mechanism as a whole cannot be Pareto efficient or social welfare maximizing. The Groves mechanism that minimizes this surplus tax revenue is called the *Clarke tax mechanism* [Clarke, 1971]. Besides not being Pareto efficient, the Groves mechanisms are not coalition proof.

Traditionally, the Clarke tax mechanism has been used to solve a single isolated social choice problem. In (AI) planning for multiple agents, this would mean voting over all possible multiagent plans. This is clearly intractable. To reduce this complexity, Ephrati has used a variant of the method where the agents repeatedly use the Clarke tax mechanism to do planning one step at a time [Ephrati and Rosenschein, 1991, Ephrati and Rosenschein, 1993a, Ephrati, 1993, Ephrati, 1994]. Although truthtelling is a dominant strategy in the isolated Clarke tax mechanism, it has not, to my knowledge, been proven that truthtelling at every step of the repeated Clarke tax mechanism is dominant. Ephrati also addressed distributing the mechanism, partial solutions for redistributing the tax waste, and partial solutions for the collusion problem.

2.2.3 Principal-Agent Problems

In another field of mechanism design, *principal-agent problems*, there are two types of actors, principals and agents [Rasmusen, 1989, Kreps, 1990]. There may be one or more of each [Kraus, 1993]. The agents have more precise knowledge than the principals. In game theoretic terms, the principal has a coarser partition of information sets. The basic idea in these games is that the agent tries to maximize its utility by using its finer information, and the principal is trying to maximize its payoff by using indirect methods to extract the agent's hidden information. In game theory (and rarely also in multiagent systems [Kraus, 1993]), *contracting* refers to the contracts that the principal uses in principal-agent problems to get at the agent's information. This differs significantly from the common use of the word *contracting* in distributed artificial intelligence, where it usually refers to a process of finding and making mutually beneficial contracts, e.g. via the contract net protocol. Issues of lying and asymmetric information are usually not involved in such studies. In this dissertation, the word *contracting* will be used in the sense that is common to DAI. Nevertheless, for completeness, several different types of principal-agent problems are summarized below. Usually in all of these game types, bargaining issues—i.e. how to split profits among the principal(s) and the agent(s)—are avoided by assuming that either the agent or the principal faces perfect competition.

In games of *moral hazard with hidden action*, the actors start with symmetric information, and make a contract. Then the agent takes an action that is not observed by the principal. The principal can therefore not condition the contract on the action. It can be conditioned on events correlated to the action though. For example, an employer principal can pay a worker agent based on the amount of goods produced by the agent, but not on how hard the agent works, because this cannot be exactly measured.

In *moral hazard with hidden information*, the actors again start with symmetric information, and make a contract. Then some event occurs that is observed by the agent but not by the principal. After that, the agent takes an action—such as reporting the outcome of the event. I will return to a related issue in Section 3.2.

In *adverse selection* problems, the agent is of some type which it knows but the principal does not. The principal tries to tailor a menu of contracts so that each agent will self-select a contract—or non-participation—in a way that is desirable to the principal. For example, a principal employer can offer a flat rate salary, and an incentive contract so that low-ability worker agents choose the former but high ability workers pick the latter.

In a special case of adverse selection called *signaling*, the agent is again of some type which it knows but the principal does not. Before the contract, the agent can take actions observed by the principal which give indications of the agent's type to the principal. For example, a potential employee agent can go to school to prove that he is a hard worker.

Screening games are like signaling games except that the agent takes the observed actions after the contract has been made. The above education example works here if the agent knows the contracts—i.e. functions of education level—that will be offered before he chooses which education to acquire.

Ephrati and Rosenschein [Ephrati and Rosenschein, 1993b, Ephrati, 1993] have discussed computational multiagent planning in a situation where the agent sincerely wants to follow the principals plan. In their setting the agent has more precise information than the principal, and cannot physically communicate that information back to the principal. Because of this, the agent has to decide how to adapt the plan to the precise situation so as to attempt to maximize the principals utility. Specifically, metrics were discussed that measure how much the intended plan and the actual plan deviate.

2.2.4 Two-Agent Task Revelation and Allocation

Rosenschein and Zlotkin have applied game theoretic mechanism design concepts to computational multiagent systems [Rosenschein and Zlotkin, 1994]. They address the problem of making insincere self-interested agents reveal private information such as their goals or worths for states. Unlike the contracting protocols of this dissertation, which contain a series of negotiation iterations, their mechanisms assume that the negotiation takes place in a single-shot isolated manner. They also assume that the agents can optimally solve exponentially many $\mathcal{NP}$-complete problems without computation costs. This is intractable if the problem instances are not small. On the other hand, in this dissertation, limitations on the agents' computation capabilities are addressed as a key issue. Furthermore, the agents in this dissertation are allowed to make side payments to each other while those of Rosenschein and Zlotkin are not. Also, their work on negotiation is specific to 2-agents setting while the work in this dissertation applies to any number of agents. Finally, Rosenschein and Zlotkin assume that the agents have symmetric capabilities and equal costs for handling tasks or moving the world from one state to another. This dissertation does not make such assumptions, and these assumptions do not hold in the application domains discussed. This point will be discussed in detail later in this dissertation.

The methods constructed by Rosenschein and Zlotkin satisfy the criteria of individual rationality, stability, symmetry, Pareto efficiency, and independence of irrelevant alternatives, and they are invariant regarding the units in which utility is measured. The last four properties stem from using deals that maximize the product of the agents' utilities. This point will be discussed in conjunction with the Nash bargaining solution in Section 2.3.1.

Rosenschein and Zlotkin present a domain classification, which is included in Figure 4.2. Their most general class of games, *Worth Oriented Domains* (WODs), encompasses games where the agents have different numeric preferences—worths—for different states of the world, but all agents have the same capabilities and costs for moving the world from one state to another [Rosenschein and Zlotkin, 1994, Zlotkin and Rosenschein, 1993c]. A subset of WODs, *State Oriented Domains* (SODs), includes games where the agents' worth functions over states are binary, i.e. agents have some goal states, and no states partially satisfy the agents [Rosenschein and Zlotkin, 1994, Zlotkin and Rosenschein, 1993b]. In WODs, the mechanisms attempt to make the agents declare their worths for different states, and in SODs they attempt to make the agents declare which states are goals. In a subset of SODs, *Task Oriented Domains* (TODs), the agents have tasks to handle, and they try to redistribute the tasks so that they all get handled inexpensively [Rosenschein and Zlotkin, 1994, Zlotkin and Rosenschein, 1993a]. It is assumed that there is a symmetric monotonic cost function among the agents which assigns a real value to any subset of tasks. This is the cost that any single agent incurs in handling the tasks. The work in this dissertation does not assume that the costs (capabilities) of agents are symmetric. The assumption of assigning a real number implies that one agent is capable of handling all tasks of all agents. The work in this dissertation does not make this questionable assumption.

The research focus in TODs is how truthfully do agents reveal tasks to each other when each agent only knows about its own tasks. The domain class of TODs includes subclasses with very different properties regarding insincere task revelation. A subclass, *Subadditive TODs*, is defined as those TODs where the cost of handling any two disjoint subsets of tasks together never exceeds the sum of the costs of handling the subsets separately. A subclass of Subadditive TODs, *Concave TODs*, is defined as those TODs where the cost of adding an arbitrary set of tasks to an agents original tasks never exceeds the cost of adding that arbitrary set to a subset of the original tasks. Finally, a subclass of Concave TODs, *Modular TODs* is defined as those TODs where the cost of handling any two disjoint subsets of tasks together equals the sum of the costs of handling them separately.

Three alternative types of deals are analyzed. In *pure deals*, agents are deterministically allocated exhaustive, disjoint task sets. On the other hand, *mixed deals* specify a probability distribution over such partitions. *All-or-nothing* deals are mixed deals where the alternatives only include partitions where one agent handles the tasks of all agents.

Rosenschein and Zlotkin discuss three forms of lying in task revelation. First, an agent may *hide tasks* by not revealing them. Secondly, it may declare *phantom tasks* which do not exist and cannot be generated if another agent wants to see them. Finally, it may announce *decoy tasks*, which really did not exist, but which can be generated on demand. The forms of lying that are possible in different domain classes and with different deal types are summarized in Table 2.1. It can be seen that in the more general TODs, many different lying methods can be profitable. It also follows that, because TODs are a subclass of SODs and WODs, lying can be beneficial in SODs and WODs also.

Table 2.1. Rosenschein and Zlotkin's results on lying in task revelation. 'Hid' stands for hiding tasks, 'Pha' for declaring phantom tasks, and 'Dec' for decoy lies. An 'L' indicates that lying of the specified type is profitable in some problem instances within the given domain class using the deal type. In general TODs using all-or-nothing deals, the negotiation set (set of individually rational, Pareto efficient deals) may be empty.

	General TOD			Subadditive TOD			Concave TOD			Modular TOD		
	Hid	Pha	Dec	Hid	Pha	Dec	Hid	Pha	Dec	Hid	Pha	Dec
Pure deals	L	L	L	L	L	L	L	L	L	L		
Mixed deals	L		L	L		L	L			L		
All-or-nothing deals	-	-	-			L						

2.3 Bargaining Theory

In a *bargaining* setting, agents can make a mutually beneficial agreement, but have a conflict of interest about which agreement to make. Furthermore, as in other subfields of game theory, no outcome may be imposed on any agent without its approval. Bargaining theory is classically divided in two based on the approach taken: axiomatic or strategic.

2.3.1 Axiomatic Bargaining Theory

Unlike competitive game theory, *axiomatic bargaining theory* does not use the idea of a solution concept where the agents' strategies form some type of equilibrium. Instead, desirable properties for a solution, called axioms of the bargaining solution, are postulated, and then the solution concept that satisfies these axioms is sought [Osborne and Rubinstein, 1990, Kalai, 1985, Rasmusen, 1989, Osborne and Rubinstein, 1994].

The *Nash bargaining solution* is a historically early solution concept that uses this approach [Nash, 1950a]. Nash analyzed a 2-agent setting where the agents have to decide on an outcome, and they get their fallbacks if no agreement is reached. It is assumed that the space of feasible utility vectors (one element in the vector per agent) is compact and convex. This occurs, for example, if all possible lotteries over pure deals in a domain are allowed.

When many deals are individually rational—i.e. pay more than the fallback—to both agents, there may be many Nash equilibria. For example, if the agents are bargaining over how to split a dollar, all splits that give each agent more than zero are in equilibrium. If agent one's strategy is to offer ρ and no more, agent two's best response is to take the offer as opposed to the fallback which is zero. Now one's best response to this is to offer ρ and no more. Thus, a Nash equilibrium exists for any ρ that defines a contract that is individually rational for both agents, and feasible $(0 < \rho < 1)$. Due to the nonuniqueness of the equilibrium, a stronger (axiomatic) solution concept such as the Nash bargaining solution is needed to prescribe a unique solution.

The first axiom of the Nash bargaining solution is based on the view that the agents' numeric utility functions really only represent ordinal preferences among outcomes—the actual cardinalities of the utilities do not matter. Therefore, utility functions can be transformed affinely, and the resulting game should be equivalent to the original game. The second axiom requires symmetry: if the agents have symmetric bargaining positions, their outcome utilities should be equal. Third, independence of irrelevant alternatives is required. The fourth axiom requires Pareto efficiency. It turns out that there is a unique solution that satisfies these four axioms. This Nash bargaining solution selects the utility pair that maximizes the product of the players' gains in utility over their fallback utilities.

The Nash bargaining solution can be extended to more than two agents [Kalai, 1985]. Other bargaining solutions also exist. They postulate different desiderata as axioms and arrive at a different utility combination as the outcome [Kalai, 1985].

2.3.2 Strategic Bargaining Theory

Unlike axiomatic bargaining theory, *strategic bargaining theory* does not postulate desiderata as axioms on the solution concept. Instead, the bargaining situation is modeled as a game, and the solution concept is based on an analysis of which of the players' strategies are in equilibrium. It follows that for some games, the solution is not unique. On the other hand, strategic bargaining theory explains the behavior of rational utility maximizing agents better than axiomatic approaches. The latter are not based on what the agents can choose for strategies, but instead rely on the agents pertaining to axiomatic notions of fairness.

Strategic bargaining theory usually analyses sequential bargaining where the agents alternate in making offers to each other in a prespecified order [Osborne and Rubinstein, 1990, Rasmusen, 1989, Osborne and Rubinstein, 1994, Kreps, 1990]. As an example, one can again think about deciding how to split a dollar. With a finite number of offers and no time discount, the last offerer will get it all, and the other agent is indifferent between accepting a zero offer or getting a zero fallback payoff. With time discount, a finite game can be solved starting from the end. The agent that is to make the first and last offer gets a payoff that approaches $1/(1+\delta)$ as the number of negotiation rounds approaches infinity. The term δ is the discount factor.

When the protocol in a non-discounted setting allows an infinite number of bargaining rounds, the solution concept is powerless because any split of the dollar can be supported in subgame perfect Nash equilibrium. On the other hand, in a discounted infinite round setting, the subgame perfect Nash equilibrium is unique. Specifically, the first offerer gets $(1-\delta_2)/(1-\delta_1\delta_2)$, where the first offerer's discount factor is δ_1, and the other player's δ_2 [Rasmusen, 1989]. Agreement is reached up front.

Another model of sequential bargaining does not use discounts, but assumes that there is a fixed bargaining cost per negotiation round. If the agents have symmetric bargaining costs, the solution concept is again powerless because any split of the dollar can be supported in subgame perfect Nash equilibrium. On the other hand, if the first offerer's bargaining cost is smaller than the other agent's, the first agent gets the entire dollar. If the first agent's bargaining cost is greater than the second agent's, the first agent receives a payoff that equals the second agent's bargaining cost. The second agent receives one minus this. Agreement is reached on the first round.

In classical microeconomics, assumptions of monopoly (or monopsony) or perfect competition are often made. A monopolist gets all of the gains while an agent facing perfect competition can make no profit. Real world settings usually consist of a finite number of competing agents, so neither monopoly nor perfect competition assumptions strictly apply. Osborne and Rubinstein have analyzed how close these classical assumptions come to the solutions from bargaining theory, which explicitly considers the interactions of agents [Osborne and Rubinstein, 1990].

Within the multiagent systems research community, Kraus et al. have extended the work on sequential bargaining [Kraus *et al.*, 1992]. In addition to going over previous results, they contribute new theorems to the case with outside options. They also analyze the case where one agent gains and one loses over time. Finally, the paper discusses negotiation over time when agents do not know each others types.

Unlike this dissertation, all of the models discussed above assume perfect rationality from the agents. No computation is required in finding a mutually beneficial contract. The space of deals is assumed to be fully comprehended by the agents, and the value of each potential contract known.

2.4 Market Mechanisms

Market mechanisms are another microeconomic approach that has been adapted for use in computational multiagent systems. Unlike game theoretic approaches which are normative, market mechanisms are usually descriptive. They determine which kinds of social outcomes can follow if the agents use certain specified strategies. There is usually no analysis whether such a strategy is an agent's best response, or whether an agent could do better by deviating to using another strategy.

The most extensively studied market framework is *general equilibrium theory* [Varian, 1992, Kreps, 1990]. The market has two types of agents, *producers* and *consumers*. It has a finite number of *commodities*. The amount of each commodity is unrestricted, and each commodity is arbitrarily divisible. Different elements within a commodity are not distinguishable, while elements from different commodities are. Each consumer has a utility function which encodes the agent's preferences over different consumption bundles, i.e. over vectors. Each element of the vector describes how much of a given commodity the agent consumes. Each consumer also has an initial *endowment* of the different commodities. The producers can use some commodities to produce others. The production vector of a producer describes how much of each commodity the agent produces. Net usage of a commodity is denoted by a negative number. A producer's capabilities of turning inputs to outputs can be characterized by the *production possibilities set*, which is the set of feasible production vectors. The producer's profits are divided among the consumers according to predetermined proportions which need not be equal.

The market mechanism sets the global vector of prices for units of each commodity. In a general equilibrium of the market:

- Each producer uses the feasible production vector that maximizes its profits given the prices.

- Each consumer consumes a bundle of commodities such that the agent could not afford another bundle of higher utility given its initial endowments, the current prices, and the profits it receives from producers.

- Markets clear: for each commodity, production equals consumption.

Such an equilibrium is always Pareto efficient. If the production possibilities sets are convex, and the consumers' preferences continuous, non-decreasing and locally insatiable, then at least one such equilibrium exists. A sufficient condition for uniqueness of such an equilibrium is that the demand for each good is nondecreasing in the prices of other goods.

The general equilibrium is related to coalition formation in such market games. Specifically, the general equilibrium is within the core. Moreover, any other point in the core can be removed from the core by simply increasing the number of agents while maintaining the proportions of all types of agents.

General equilibria also exist in markets with no producers. In such *pure exchange markets*, the consumers just reallocate their initial endowments among themselves [Varian, 1992, Kreps, 1990].

To reach a general equilibrium, the *tatonnement process* is usually used. This is an iterative mechanism, and the trades, production, and consumption are assumed to occur only after the process has terminated. At each iteration, the *auctioneer* sets a vector of prices. Then all agents have to declare a vector of how much they are willing to buy and sell of each commodity at the current prices. Based on this information, the auctioneer updates the price vector for the next iteration. Under certain technical conditions, this process is guaranteed to converge to a general equilibrium.

Within computational multiagent systems, Wellman has developed a general equilibrium based software system called WALRAS. As example domains, he has used WALRAS in flow routing in a network [Wellman, 1992], and in configuration design [Wellman, 1994]. Wellman discusses how the latter application fails to meet the assumptions for existence of a general equilibrium. For example, because some design parameters are discrete, production possibilities sets cannot be convex. Wellman and Mullen have also applied WALRAS to a distributed information network example [Mullen and Wellman, 1995]. The iterative market process of WALRAS differs from tatonnement. Specifically, WALRAS uses asynchronous declarations by the agents, and the agents bid demand functions (of price) as opposed to just quantities. This process converges to a general equilibrium [Cheng and Wellman, 1995]. As in tatonnement, trades in WALRAS only occur after the market process has terminated.

Usually in the market mechanisms studied up to date, the agents are assumed to act truthfully in declaring the amounts that they are willing to buy or sell at the prices of the specific iteration. It has not been proven that these revelation strategies themselves are in equilibrium in the sense of competitive game theory. It can be the case that an agent could do better by misrepresenting at some iteration. For example in general equilibrium approaches, a more desirable equilibrium for that agent could be reached. Truthful declaration is reasonable if it is assumed that the market is so large that the impact of any single agent is totally negligible. In other words, if an agent is a price taker, no amount of manipulation on its behalf is going to change the final global price vector. Because the agents are assumed to reveal truthfully, they do not counterspeculate other agents' declarations either.

The problem domains of this dissertation do not fall within the scope of general equilibrium theory. First, tasks are usually countable (i.e. not infinitely divisible), which violates the assumptions for existence of a general equilibrium. Secondly, most tasks (commodities) are unique as opposed to there being many tasks of a given type—so there is usually only at most one item of each "commodity". Third, the approach of the dissertation is mostly normative as opposed to the descriptive studies of general equilibrium processes. Finally, in the dissertation, the agent's limited rationality precludes the exact computation of the whole production possibilities set.

Also, traditional market models do not account for *externalities*. In *consumption externalities*, one agent's consumption affects another agent's utility. In *production externalities*, one agent's production possibilities set is directly affected by another agent's actions. Glance and Hogg have presented examples of computational ecologies (not based on general equilibrium theory) where externalities are so dominant that, counterintuitively, adding resources to the system makes it operate less efficiently [Glance and Hogg, 1995]. Hogg has also shown that externality problems are likely to be common in computational ecosystems [Hogg, 1995]. Evolutionary aspects of such systems have also been discussed [Miller and Drexler, 1988a], and the behaviors under incomplete and delayed information analyzed [Huberman and Hogg, 1988]. Some mechanisms to attack externality problems include taxes and viewing some of the externality issues as commodities themselves [Varian, 1992].

Other price-based market mechanisms exist besides general equilibrium approaches. For example, Kuwabara et al. have studied computational societies where the agents are either buyers or sellers. In their approach, the sellers set the prices while the buyers decide how much of each commodity to purchase [Kuwabara and Ishida, 1992, Kuwabara *et al.*,]. Huberman and Clearwater have applied a price-based market mechanism to controlling building heating [Huberman and Clearwater, 1995]. In their mechanism, exchanges occur after each iteration. The agents submit buy and sell bids, and the auctioneer sets the price at each iteration so as to make supply equal demand. The buyers then pay based on the bids. Money is repeatedly dribbled to the agents so that they can continue bidding. The system works in practice although its formal properties have not been proven. The *Spawn* system for allocating independent computation jobs among heterogeneous processors in a network works on the same basis [Waldspurger *et al.*, 1992]. Every time a processor becomes idle, it runs a second-price sealed-bid auction in which computation jobs can bid for the right to execute for a duration specified in the bid. The agents use prespecified strategies: the approach is not normative. The formal properties of Spawn are unproven, but the experimental results suggest that Spawn achieves efficient usage of the processors on the net with a low coordination overhead (about 10%), and achieves relatively good fairness among jobs. In Spawn, virtual money is continuously dribbled to the jobs, but there is no mechanism against forging currency or against insincere agents.

Ferguson et al. [Ferguson *et al.*, 1988] have studied a similar market mechanism for allocating jobs to processors based on repeated auctions, but their focus was not so much on fairness as on global efficiency. Miller and Drexler have discussed computational price-based market mechanisms as a basis for software publishing and distribution [Miller and Drexler, 1988b] as well as a mechanism for implementing processor scheduling and garbage collection in computer systems [Drexler and Miller, 1988].

In addition to the price-based market mechanisms, resource-based mechanisms exist. For example, Kurose and Simha have developed a market mechanism—applied to file allocation—where at each iteration, the agents compute their marginal worths for resources [Kurose and Simha, 1989]. Based on these worths, resources are reallocated at every iteration. This differs from equilibrium approaches where trades only occur after all iterations have been completed. In their resource-based approach, the solution gets better at every iteration, and is guaranteed to finally converge to the optimum.

The *contract net protocol* (CNP) [Smith, 1980, Smith, 1978, Smith, 1979, Davis and Smith, 1981, Davis and Smith, 1983, Smith and Davis, 1981] for decentralized task allocation is one of the important market-like paradigms developed in distributed artificial intelligence. Its significance lies in that it was the first work to use a computational negotiation process involving a mutual selection by both *contractors* and *contractees*. The CNP mechanism resembles a directed government contracting scheme, where each involved party is allowed to make one bid for each announcement it receives, and the bids of the other parties are not revealed to it. The negotiations are directed in the sense that an announcement is not sent to all other agents—only to likely contractees [Van Dyke Parunak, 1987]. The CNP was initially applied to task allocation in a simulated distributed sensor network for acoustic interpretation. Later it has been applied to job dispatching among machines within a manufacturing plant [Van Dyke Parunak, 1987]. The *Enterprise* [Malone *et al.*, 1983, Malone *et al.*, 1988] system for allocating computation jobs among processors in a network used the CNP mechanism. The choice of a processor was based on expected completion time. Recently, Sen has studied the application of CNP principles to distributed meeting scheduling [Sen, 1994, Sen, 1996]. His work studied the efficiency of the global system under different local contracting strategies, such as counterproposing different meeting times, and decommitting from previously scheduled meetings. Other recent work on the CNP includes a restricted quantitative analysis [Gu and Ishida, 1995].

In these application, the agents were totally cooperative, and selection of a contractee was based on suitability, for example adjacency, processing capability, and current agent load. However, there was no formal model discussed for making task announcing, bidding and awarding decisions. The announcements, bids, and awards were not based on real microeconomic principles such as prices or quantitative demands. This dissertation presents such a formal model, where agents locally calculate their marginal costs for performing sets of tasks. The choice of a contractee is based solely on these costs.

This dissertation will also address several other previously unaddressed questions regarding the CNP. As evidenced by my recent research, the traditional CNP is not an off-the-shelf mature technology that can be applied to different domains as is. The protocol really includes enormous numbers of design alternatives. Many of these will be discussed in the dissertation (Chapter 3), and prescriptions will be made regarding choices among these alternatives. For example, previous work on the CNP has not addressed the risk attitude of an agent toward being committed to activities it may not be able to honor, or the honoring of which may turn out to be unbeneficial. Additionally, in previous CNP implementations, tasks have been negotiated one at a time. This is not sufficient, if the effort of carrying out a task depends on the carrying out of other tasks. The framework is extended to handle task interactions, among other methods by clustering tasks into sets to be negotiated over as atomic bargaining items. The practical problem of announcement message congestion is solved, and several other issues regarding real asynchronous implementation will be addressed. Finally, the question of local deliberation scheduling in the negotiations has not been discussed earlier, but is a key focus in this dissertation. The hypothesis is that distributed contracting can be developed into an efficient—in terms of results and computation complexity—interaction mechanisms for self-interested agents whose rationality is bounded by limited computation resources.

2.5 Non-Economic Approaches in DAI

Historically, the field of distributed artificial intelligence (DAI) [Bond and Gasser, 1988, Huhns, 1987, Gasser and Huhns, 1989] has been split into two branches which have classically been called *cooperative distributed problem solving (CDPS)* [Durfee *et al.*, 1989, Decker *et al.*, 1989, Lesser and Erman, 1980] and *multiagent systems (MAS)* [Rosenschein and Genesereth, 1985, Genesereth *et al.*, 1984]. The distinction has been fuzzy, but the main difference is that in CDPS, the agents are cooperative, e.g. because they are designed centrally, while in MAS, the agents are self-interested, e.g. because the agents are designed by separate parties. The focus of MAS has mainly been to solve problems related to self-interest while research in CDPS has mostly tackled computational questions. As can be seen, the term MAS does not clearly capture the difference. Lately the whole field of DAI has been called MAS, while CDPS has been viewed as a subfield of it. This corresponds well with the terms, but not with the historical progression. The field of CDPS began in the late 1970's and preceded the field of MAS which began in the mid 1980's. So far this document has reviewed MAS and microeconomic research relevant to it. In this section I will review CDPS and other related topics.

The work in CDPS began with the Hearsay II speech understanding system, which used a blackboard architecture to structure the computations [Lesser and Erman, 1977, Lesser and Erman, 1980]. Different knowledge sources worked together to solve the interpretation problem by posting partial results in a common data structure called the blackboard. The aim of CDPS agents is to maximize the global good of agents. This goal is usually approached heuristically. Some important questions include the scheduling of knowledge source executions and the adequate—yet not overwhelming—distribution of partial results and control knowledge. Although blackboard systems were originally used in a single agent setting where the agent had several knowledge sources, lately blackboard systems have been used to simulate cooperative problem solving systems consisting of multiple agents with many knowledge sources each [Lesser and Corkill, 1981, Lesser, 1991, Lesser and Corkill, 1983, Durfee and Lesser, 1989, Durfee and Lesser, 1991, Neiman *et al.*, 1994]. After the Hearsay II research, focus of the early CDPS research switched to the Distributed Vehicle Monitoring Testbed (DVMT) [Lesser and Corkill, 1983], which is a blackboard-based centralized simulation of geographically distributed acoustic sensor agents that coordinate their activities to form an interpretation of vehicle movements in the area. The initial paradigm was that of Funtionally Accurate/Cooperative (FA/C) agents [Lesser and Corkill, 1981, Lesser, 1991]. Later the agents were equipped with the ability to form and negotiate over partial global plans (PGP) of performing interpretation actions [Durfee and Lesser, 1989, Durfee and Lesser, 1991]. The latest work on this line of research has studied general coordination relationships and statistically analyzed how different coordination algorithms perform on the task [Decker and Lesser, 1993b, Decker and Lesser, 1993a, Decker and Lesser, 1995].

As discussed in Section 1.1.2, most work on distributed manufacturing scheduling also falls into the category of cooperative distributed problem solving. The CDPS approach has also been applied to other problems. In the *multi-stage negotiation* *protocol* [Conry *et al.*, 1991] it was used to solve distributed constraint satisfaction problems where the agents did not have a global view, e.g. due to communication bandwidth limitations. The algorithm allows the identification of local decision that are disadvantageous from the global perspective, and it incorporates goal relaxation. As an example, the method was applied to circuit restoration in a network. In the DENEGOT simulation [Moehlman *et al.*, 1992], CDPS was used to coordinate fire fighting bulldozers and firebosses. The distributed search was structured in a small number of levels of satisfaction. Search proceeded into a worse level whenever a solution on the previous, better level could not be found. In the TEAM simulation, CDPS was applied to a distributed design problem of configuring a small number of components into a desirable steam-condenser [Lander and Lesser, 1993].

Another non-economic approach to DAI is that of social laws [Tennenholtz and Moses, 1989, Moses and Tennenholtz, 1993, Shoham and Tennenholtz, 1993], where binding conventions for all agents are designed off-line. The idea is that this would increase efficiency of the agent society because the conventions reduce the need for on-line coordination.

Yet another body of non-economic DAI work focuses on constructing logic-based characterizations of rational behavior [Cohen and Levesque, 1987, Cohen and Levesque, 1990, Grosz and Kraus, 1993, Rao and Georgeff, 1991, Rao and Georgeff, 1995, Morgenstern, 1990, Fagin *et al.*, 1995]. The idea is that these formalisms could be used to make an automated agent act rationally. These characterizations are often called beliefs, desires, intentions (BDI) models. This line of research has grown from single agent logic-based approaches, and most often the subject of study is an individual agent within the agent society. They study for example when an agent should commit (to itself) to a plan, and how these commitments should be kept. Some of this research also focuses explicitly on commitments in multiagent plans.

The *recursive modeling method* (RMM) is a non-economic, but utility-theoretic approach to multiagent systems [Gmytrasiewicz and Durfee, 1992, Durfee *et al.*, 1993, Gmytrasiewicz and Durfee, 1995]. There, the agents have no *a priori* interaction protocol, but they model each other recursively. This enables them to evaluate the utility of their actions—both domain actions and communication actions. In some cases this leads to the emergence of interaction protocols. One problem with the method is that there is often no clear way to decide how the recursion should bottom out. Lately, Vidal and Durfee have used the approach of only extending the recursion further if it is expected to provide more gains than the cost of the deliberation [Vidal and Durfee, 1995].

2.6 Resource-Bounded Reasoning

Although the CDPS work has focused on computational issues, almost all of the work on multiagent systems consisting of self-interested agents has ignored the computational complexity that is assumed of the agents' local reasoning [Rosenschein and Zlotkin, 1994, Zlotkin and Rosenschein, 1993c, Zlotkin and Rosenschein, 1993b, Zlotkin and Rosenschein, 1993a, Ephrati and Rosenschein, 1991, Ephrati, 1994, Kraus *et al.*, 1992]. Often, computational limitations render the traditional mechanisms inapplicable when problem instances are not extremely small. This section first reviews different approaches to controlling single agent reasoning under computational limitations. After that, work on computational bounded rationality in multiagent contexts is reviewed. This dissertation makes use of some of the ideas of normative deliberation control in single agent settings, and extends them to multiagent systems.

Traditional AI systems work off-line. Their input is a complete problem description, and their output—acquired after some unknown, long time delay—is a complete answer. As problem solving systems scale up and longer execution times become common, many problem solving tasks become real-time [Garvey and Lesser, 1994], i.e. considerable changes in the real world take place during an agent's deliberation. The value of an agent's action in the real world may decrease with time, thus necessitating a tradeoff between deliberating to find a good action and performing some action early on. As an answer to the real-time requirements on AI, some research has focused on reactive systems [Brooks, 1986] that provide the needed responsiveness but are often unable to solve complex problems adequately. One alternative between these two extremes is to have the agent dynamically trade off solution quality against computation time. This can be done e.g. by reasoning about reasoning, i.e. *meta-reasoning*. For example, the agent can choose to deliberate longer when the answer is not needed soon, but it can provide crude answers quickly if required by its environment. The qualitative tradeoff between time and quality is intuitive, but its operationalization is challenging. Some approaches use ad hoc meta-level control policies, but it is more grounded to use probability theory and utility-theory in the choice of base-level actions. The latter approach is called *decision-theoretic deliberation control*.

Three questions have to be answered to control single agent reasoning: which deliberation tasks to execute, in what order, and how much time to allocate to each one. This research can be divided according to two main criteria: interruptibility and number of reasoning tasks. In *design-to-time algorithms* [D'Ambrosio, 1989, Garvey and Lesser, 1993] the meta-reasoner implicitly sets the run time and expected result quality by setting the reasoning algorithm parameters before execution. Design-to-time algorithms are not guaranteed to have any answer available if interrupted before the planned time, and in the simplest case they have only one computational task. Garvey and Lesser [Garvey and Lesser, 1993] study the combination of several computational tasks—some using the results of others—into a composite design-to-time algorithm. *Anytime algorithms* (also called flexible computations) are examples of interruptible processing. Their result quality usually increases with time, and some answer is available at any time. Newton's iteration for root finding is an example of an anytime algorithm with one computational task. Horvitz [Horvitz, 1987, Horvitz, 1988] addresses the choice of a single anytime algorithm and the number of steps to run it in a medical diagnosis domain. Dean and Boddy [Boddy and Dean, 1994, Dean and Boddy, 1988, Boddy and Dean, 1989] study the sequential time allocation to multiple anytime algorithms in order to create a deliberation schedule that maximizes the sum of the qualities of responses to multiple events that are known ahead of time. The combination method is optimal for deterministically known events if the anytime algorithms' (one for each event) performance profiles are increasing and convex, and the algorithms are independent: they do not use each others results. Zilberstein and Russell [Zilberstein and Russell, 1991, Zilberstein and Russell, 1995, Zilberstein, 1993] analyze the time allocation among multiple anytime algorithms that form a composite design-to-time algorithm and present a method by which this type of design-to-time algorithm can be made interruptible by executing and reexecuting the algorithm with repeatedly doubling time allocations. In the best case, this method uses twice the amount of computation that would be optimal in case an oracle were available to reveal the actual time of interrupt. In the worst case, the method uses four times that amount. Russell and Wefald [Russell and Wefald, 1991b, Russell and Wefald, 1991a] study an agent that has to repeatedly choose between several deliberation actions and the currently highest ranked real world action. This can be viewed as a multitask anytime algorithm control problem, where time is allocated to deliberation actions in chunks, i.e. deliberation actions and real world actions are non-interruptible. The method has been used in adversary search and in single agent search. Sandholm and Lesser have presented an optimal method for terminating an anytime decision algorithm when the performance profile is conditioned on execution so far [Sandholm and Lesser, 1994]. Progress monitoring with similar conditioning has later been studied in optimization problems, although in that work, optimality was sacrificed by making simplifying assumptions [Hansen and Zilberstein, 1996]. Specifically, the remaining performance profile was conditioned on the quality of the solution so far, but not on the path that the quality of the solution had taken.

Decision-theoretic control architectures need information value estimates. Russell and Wefald [Russell and Wefald, 1991b, Russell and Wefald, 1991a] estimate the value of a possible computation as the difference of the change in intrinsic utility and time cost. They compute the change in intrinsic utility as the change in the utility that the agent gets by not executing the real world action perceived best before the computation, but executing the action perceived best after it (either action assumed to occur after the computation). Horvitz [Horvitz, 1987] computes the comprehensive value of a computation as the product of a non-time-dependent object-related value and a time-dependent discount factor. As mentioned in both of these papers, the separation of the information value function into two parts—one for taking time into account and one for other aspects—is not always possible in practice. In general, information value is a function of both the answer quality and time. These two factors can be reasonably separated only if they are independent. The problem of finding the exact form for the time-dependent part is crucial because algorithm choice and optimal time allocation are very sensitive to it [Horvitz, 1987]. Sandholm and Lesser present a method for computing the time-dependent value of probabilistic information [Sandholm and Lesser, 1994]. This method takes into account both the object-related and the time-related part simultaneously.

Lesser et al. [Lesser *et al.*, 1988] identify three general ways to reduce solution quality for approximate processing. Their work focused on interpretation domains but the same three ways apply to planning domains also [Zilberstein and Russell, 1991, Zilberstein and Russell, 1995, Zilberstein, 1993]. One way is to *ignore some solution aspects*. Analogously, one could ignore some details of a solution to an optimization problem. A second way to reduce solution quality is to compromise precision. This is analogous to *relaxing optimality* in optimization problems. This type of approximation has been widely studied for $\mathcal{NP}$-complete optimization problems. Often, a fast algorithm exists that guarantees a solution within an error bound from optimum.

To solve a troublesome problem in $\mathcal{NP}$, we could reduce it in polynomial time to an $\mathcal{NP}$-complete problem for which an approximation scheme of this type is known. There is a complication, however. In general, reductions do not preserve this type of approximation, i.e. the fact that the new reduced problem can be solved approximately within a certain error bound does not guarantee that the acquired solution is an approximate solution (within any reasonable error bound) to the original problem. The search for approximation preserving reductions, called L-reductions, is an active field of research in theoretical computer science [Papadimitriou and Yannakakis, 1991]. A third way of reducing solution quality for approximate processing is to *decrease certainty*. This dissertation focuses on the second type of approximation: relaxing optimality. This is done via an added component that trades off solution quality against deliberation time.

Less work has focused on resource-bounded reasoning in multiagent systems than in single agent settings. Within economics it has been realized that models that assume perfect rationality on behalf of the actors often do not accurately predict interaction outcomes among humans. Therefore, the concept of bounded rationality has been discussed, where cognitive limitations of the interacting humans are also considered [Simon, 1982, Good, 1971]. Because the rationalities associated with humans are hard to quantify, these theories have mostly been qualitative. Secondly, they have been descriptive in that they exemplify how human rationality might be limited, but do not prescribe deliberation control methods that humans ought to adopt. On the other hand, this dissertation focuses on constructing normative and quantitative theories of multiagent systems involving bounded rationality: agents try to maximize utility *given their limited reasoning capability*. Heiner has made some first steps towards quantifying the original descriptive models of bounded rationality [Heiner, 1983, Heiner, 1985]. He constructs an agent's action repertoire based on the characteristics of the environment that the agent resides in. He presents the *reliability condition*, which states that an action should be added to an agent's action repertoire if the reliability in selecting the action exceeds the minimum required reliability necessary to improve performance, which in turn depends on how common the correct environmental conditions for the new action are and how much an agent gains by correctly choosing the new action and how much an agent looses by incorrectly choosing it. In other words, the bound on rationality is the fact that the agent cannot always distinguish which action is the correct one to choose in a given situation. To choose the action repertoire, an agent greedily adds actions starting with an empty action set, or greedily removes actions starting with the set of all possible actions. One problem with this approach is that in reality, the gain of choosing the new action in the right situation and the loss of choosing it in the wrong situation depend on what alternative actions already exist in the repertoire. Moreover, the probability of choosing the right action should decrease and the probability of choosing the wrong action should increase as the number of actions in the repertoire increases. Therefore, a greedy algorithm is not sufficient in generating an optimal action repertoire. Also, the internal deliberation process of perceiving the environment and choosing an action is not described, and no cost is associated with deliberation itself once the action repertoire has been chosen.

Bounded rationality in a multiagent context has also been rigorously studied in repeated games [Rubinstein, 1986, Abreu and Rubinstein, 1988, Neyman, 1985, Papadimitriou and Yannakakis, 1994, Kalai and Stanford, 1988, Gilboa and Samet, 1989]. In this work, two agents meet repeatedly in a structurally simple game such as the prisoner's dilemma (Table 1.2). Each agent is represented by a deterministic finite automaton (DFA), and the bound on an agent's rationality stems from a limit on the number of states in its DFA. So the rationality is restricted in terms of what histories an agent can remember, not in terms of how well it is able to solve combinatorial problems. Some of the results are counterintuitive, since in some cases, the less rational agent is better off [Gilboa and Samet, 1989]. Secondly, the model has been used to explain how cooperation occurs even in a finitely repeated prisoner's dilemma game, where rational agents would defect at each iteration [Neyman, 1985, Papadimitriou and Yannakakis, 1994]. As a model of a real computerized agent, a DFA is unrealistically weak. The abstract computing device that is usually used to model the capabilities of a computational agent is the Turing machine. Because any real digital computer (and any analog computer with finite resolution) has finite storage capacity (unlike the Turing machine), theoretically any computer can be modeled by a DFA. But even a computer with only eight million bits (about one megabyte) of memory—virtual included—would need a DFA of $2^{8000000}$ states to model it. Moreover, some of the results on DFAs playing repeated games hinge on the assumption that one player can identify the opponent's DFA. This is clearly infeasible, since even with the small example computer above, this would require more than $2^{8000000}$ repetitions of the game. Recently, Mor and Rosenschein showed how time can be used to enable cooperation in a finitely repeated prisoner's dilemma among bounded rational agents [Mor and Rosenschein, 1995]. Their idea is that an agent uses time whenever it checks from its memory the number of the iteration that is occurring. This delays its response, and the other agent can observe the delay. This fact allows for cooperative equilibria, where neither agent has time to check the iteration number (in fear of a punishment from the opponent), and thus the finitely repeated game becomes equivalent to an infinite one because the agents do not know when the last iteration will occur.

This dissertation uses a different model of bounded rationality. The bound on rationality stems from the agent's limited capability to solve combinatorial problems. In Chapter 4, an explicit model of this is used, where an agent can solve combinatorial problems with iterative refinement so that the domain cost of the solution decreases with allocated computation time. This rationality bound is quantitatively characterized by the algorithm's performance profile. This allows optimal trading off of solution quality against computation time. This normative model of resource-bounded reasoning enables the construction of normative theories of interactions—e.g. coalition formation—among computationally limited agents.

The next chapters discuss the contributions of this dissertation. Chapter 3 analyzes automated contracting, Chapter 4 discusses coalition formation, and Chapter 5 presents methods for unenforced contract execution.

CHAPTER 3

AUTOMATED CONTRACTING

This chapter presents extensions to the Contract Net Protocol (CNP), Section 2.4, that allow it to operate among self-interested, computationally limited agents. In automated contracting, the agents make deals to iteratively reallocate tasks among themselves in order to be able to handle the tasks more efficiently than they could with the original distribution of tasks among agents.

We cast such negotiations in the following framework. Each agent has a (possibly empty) set of tasks and a (possibly empty) set of resources it can use to handle tasks. These sets can change due to domain events, e.g. new tasks arriving or resources breaking down. The agents can subcontract tasks to other agents by paying a compensation. This subcontracting process can involve breaking a task into a number of subtasks handled by different agents, or clustering a number of tasks into a supertask. A task transfer is profitable from the global perspective if the contractee can handle the task less expensively than the contractor, or if the contractor cannot handle it at all, but the contractee can. So, the problem has two levels: a *global task allocation problem*, and each agent's *local combinatorial optimization problem* defined by the agent's current tasks and resources. The goal of each agent is to maximize its payoff which is defined as its income minus its costs. Income is received for handling tasks, and costs are incurred by using resources to handle the tasks. The model allows resources to have idling costs, so an agent's local cost can be positive even with no tasks. However, this is not the case in our vehicle routing problem.

We restrict ourselves to domains where the feasibility and cost of handling a task do not depend on what other agents do with their resources (the control of no resource is shared) or how they divide tasks among themselves, but do depend on the other tasks that the agent has—as is the case in our distributed vehicle routing problem, and usually also in scheduling. The global solution can be evaluated from a social welfare viewpoint according to the sum of the agents' payoffs.

This dissertation concentrates on negotiation in hard multiagent domains where a task can increase or decrease the *marginal cost* [1] of another task that the agent has. This corresponds to having both diseconomies and economies of scale in the domain. If only economies of scale were present, the task allocation problem would be trivial from the social welfare perspective: the optimal solution would be one where one agent handles all agents' tasks [Rosenschein and Zlotkin, 1994, Zlotkin and Rosenschein, 1994]. On the other hand, in our setting that realistically also allows diseconomies of scale, the distribution of tasks among agents would not be trivial even if it were done centrally or among totally cooperative truthful agents. The best solution may be any partition of tasks among agents. Optimal task allocation is itself a hard combinatorial problem.

Also, solving an agent's local optimization problem for any task set that is currently allocated to the agent is a hard combinatorial problem. Therefore, as opposed to the bulk of the work in game theory and the formal work on computational multiagent systems, it will not be assumed that agents have an exact view of their marginal costs for tasks. Instead these marginal costs can be only found out by involved computations that can usually not be carried out optimally due to computational resource bounds and time constraints. Secondly, it will not be assumed that the agents have similar capabilities, i.e. the marginal cost of a task may vary between agents due to for example their static resources or the dynamic state of their resources.

[1] The marginal cost of a task to an agent is the cost of the agent's solution with the task minus the cost of the agent's solution without it. Thus, to calculate the exact marginal cost, two different (combinatorial) problems of the agent would need to be solved optimally.

The next sections detail the contributions that the dissertation makes to automated contracting.

3.1 Contracting Based on Marginal Costs

Previous contract net protocol (CNP) implementations (presented in Section 2.4) have assumed totally cooperative agents, and the mutual selection of contractors and contractees has been made heuristically based on suitability, for example adjacency, processing capability, and current agent load. However, no formal model has been discussed for making task announcing, bidding and awarding decisions.

On the other hand, the Transportation Cooperation Net (TRACONET) system, developed as an initial part of this thesis research, introduced such a formal model into automated contracting [Sandholm, 1993, Sandholm, 1992a, Sandholm, 1991, Sandholm, 1992b]. In that model, agents locally calculate their marginal costs for performing sets of tasks. The choice of contractors and contractees is based solely on these costs. The pricing mechanism generalizes the CNP to work for both cooperative and competitive agents. The TRACONET system operates under market assumptions: it is not game theoretic. For example, the agents truthfully reveal their marginal costs, and gains are always divided in half between the two parties of each contract. Also, agents do not counterspeculate into the future of the contracting process. On the other hand, later sections of this chapter, and the other chapters of this dissertation use fully game theoretic analysis.

TRACONET is a running implementation of automated contracting in the distributed vehicle routing domain. In TRACONET, each computational agent represents one dispatch center. While the original CNP implementation was a centralized simulation, TRACONET is a truly distributed system. Each agent is implemented as one Unix process, and they pass negotiation messages asynchronously over the network.[2]

[2]The system is implemented in an object-oriented manner using the C++ language and the X11 window system.

In solving the vehicle routing problem, each agent first solves its local routing problem. After that, an agent can potentially negotiate with other dispatch agents to take on some of their deliveries or to let them take on some of its deliveries for a dynamically constructed charge. In the negotiations the agents exchange sets of deliveries whenever this is profitable, i.e., whenever a contractor is able to carry out the task set with less costs than the manager agent. The negotiations can be viewed as an iterative way of enhancing the global routing solution by traversing a sequence of task allocations among agents. At every step of this iteration, each agent keeps a feasible local routing solution for the tasks that it has been allocated. Here 'feasible' means that the agent can take care of all of its deliveries with its vehicles. With this iterative task reallocation scheme, a global solution closer to the global optimum is reached although no global optimization run is performed.

The negotiation is real-time since after each contract is made the exchange of tasks and payments is made immediately. There is no iteration among the agents until an equilibrium is reached unlike in general equilibrium market approaches (Section 2.4). The scheme also works for dynamic domains where, between individual negotiations, some delivery orders may have been dispatched, new orders may have arrived, and the available vehicles may have changed. The iterative task allocation scheme is an anytime algorithm that can be stopped at any time, and still have a feasible solution. This solution is guaranteed to be no worse than the initial solution where the agents operated individually because agents only make profitable task transfers.

An agent can act both as a manager and a contractor of delivery sets, but it does not have to take both roles, nor is it required to negotiate with all other agents. Further, each agent can reallocate deliveries received from other agents. When announcing, an agent tries to buy some other agent's transportation services at a price, the maximum of which it specifies in the announcement. When bidding, an agent tries to sell its own services at a price, the minimum of which it specifies in the bid. Awarding means actually buying the services of some other center and award taking means actually selling one's services.

3.1.1 Announcing Tasks

In announcing, an agent chooses a set of deliveries from the deliveries of the center and announces them to other centers in order to get bids from them.[3] The announcing methods differ from each other for example in whether a delivery set that has already been announced can be reannounced. Reannouncing leads to better results because some tasks that cannot be reallocated profitably can be reallocated profitably later when other contracts have taken place and changed the existing task allocations. On the other hand, reannouncing negotiations are longer than announce-once negotiations. This is not a serious problem, if we assume that actual deliveries are being done during the negotiations and reannouncing is not done immediately.

When announcing a set of tasks, the agent includes a number $\rho^{announce}$ which is the maximum that any bidder may require the agent to pay for having the tasks handled. This allows unprofitably expensive bidders to not waste negotiation resources of the system such as computation and communication. Because an agent (call it i) is willing to make any individually rational contract, it announces

$$\rho^{announce} = c_i^{remove}(T^{announce}|T_i) \tag{3.1}$$

where $c_i^{remove}(T^{announce}|T_i)$ is its marginal cost for removing the task set $T^{announce}$ from its routing solution which incorporates all of agent i's tasks T_i:

$$c_i^{remove}(T^{announce}|T_i) = c_i(T_i) - c_i(T_i \cap T^{announce}) \tag{3.2}$$

Here $c_i(T)$ is the cost of the optimal routing solution (sum of route lengths) for tasks T with the vehicles of agent i. Because the vehicle routing problem is $\mathcal{NP}$-complete,

[3]In the implementation the announcements focus on deliveries ending in the geographical main operation areas of the potential contractors, because these deliveries are most likely to lead to contracts. This efficiency improving heuristic is not necessary for the contracting approach to work, however.

computing even one of the two c_i's is intractable. Therefore, an approximation scheme is used for computing $c_i^{remove}(T^{announce}|T_i)$ directly.

Note that even though the negotiation methods are domain independent, this approximation algorithm is domain specific, and has to be rewritten when the contracting software is transported across problem domains.

If the value of this approximation is too low, other agents will not bid even though that might be beneficial. On the other hand, if the estimate is too high, the agent may receive unbeneficial bids also. The actual value is not as crucial here as it is in the awarding phase, because announcements are not binding in TRACONET—unlike bids and awards. Therefore, even an incorrect approximation will not lead to unbeneficial contracting.

3.1.2 Bidding

In bidding, an agent reads the announcements sent by other agents. If the maximum price $\rho^{announce}$ mentioned in the announcement is higher than the price that the deliveries would cost if done by this agent, a bid is sent with the latter price ρ^{bid} (otherwise, no bid is sent for the specified announcement). Formally, a bidder j bids

$$\rho^{bid} = c_j^{add}(T^{announce}|T_j) \tag{3.3}$$

where $c_j^{add}(T^{announce}|T_j)$ is its marginal cost for adding the task set $T^{announce}$ to its routing solution which incorporates all of agent j's tasks T_j:

$$c_j^{add}(T^{announce}|T_j) = c_j(T_j \cup T^{announce}) - c_j(T_j) \tag{3.4}$$

Here again $c_j(T)$ is the cost of the optimal routing solution for tasks T with the vehicles of agent j. Because the vehicle routing problem is $\mathcal{NP}$-complete, computing even one of the two c_j's is intractable. Therefore, an approximation scheme is used for computing $c_j^{add}(T^{announce}|T_j)$ directly.

If the value of this approximation is lower than the true value, the agent may have to accept an unbeneficial contract. This holds when beneficiality is defined with respect to the optimal solutions. However, because the agents cannot compute the optimal solutions, the approximated marginal cost is the more relevant criterion of beneficiality—at least as long as the approximation algorithm is the same as the algorithm that generates the new solution when the tasks are actually awarded to the bidder. In this case, the approximated marginal cost is really the cost that the bidder incurs by handling the announced delivery tasks. Similarly, if the value of the approximation is too high, an agent may miss some contracts that would be beneficial if it could solve combinatorial problems optimally, but that are not beneficial when it uses the approximation scheme to actually incorporate tasks into its routing solution.

3.1.3 Awarding

In awarding tasks, an agent reads the bids of other agents. Before handling the bids concerning a certain announcement, it checks that a fixed time has passed since the sending of the announcement, so that many potential contractees have had time to bid. At this point, the awarder agent can recheck that the contract is still beneficial to it. This is not obvious, because its other contracting activities or domain events may have changed its other tasks. Therefore the awarder's current set of tasks T_i' need not equal the set T_i that it had at the time of announcing. Now, the awarding price

$$\rho^{award} = c_i^{remove}(T^{announce}|T_i') \tag{3.5}$$

where $c_i^{remove}(T^{announce}|T_i')$ is its marginal cost for removing the task set $T^{announce}$ from its routing solution which incorporates all of agent i's tasks T_i':

$$c_i^{remove}(T^{announce}|T_i') = c_i(T_i') - c_i(T_i' \cap T^{announce}) \tag{3.6}$$

Here $c_i(T)$ is the cost of the optimal routing solution for tasks T with the vehicles of agent i. Again, computing the c_i's is intractable. An approximation algorithm is used for estimating $c_i^{remove}(T^{announce}|T_i')$ directly with similar considerations regarding over and under estimation as in the announcing and bidding stages.

Then, if ρ^{award} is greater than the lowest bid, the awarder agent sends an award message to the agent with the least expensive bid. By convention, the contract takes place at price $\frac{\rho^{bid}+\rho^{award}}{2}$, which the awarder pays to the bidder. Finally, the awarder removes the set of deliveries $T^{announce}$ from its routing solution.

3.1.4 Receiving Awards

When an agent gets awarded a task set, it has to accept it because bids are binding in TRACONET. Therefore, no new marginal cost calculation is needed. Instead, the agent just incorporates the awarded tasks into its routing solution. This is done via the same algorithm that was used for marginal cost approximation in the bidding phase.

Some contracts may have sneaked in between the bidding for a certain set of tasks and taking the corresponding award. These contracts have altered the routing solution: taking the award might no longer be profitable for the agent. Because bids are binding, the center is committed to taking the award anyway. Making bids non-binding would not solve the problem, because the contractee, after receiving an award, would have to inform the contractor that it has taken the award or that it will not take it. This would require the contractor to keep the delivery set in its routing solution until award taking is confirmed, during which, some changes may have sneaked into its routing solution and the problem rearises.

3.1.5 Scaling Up

The presented marginal cost based task reallocation negotiation works efficiently even for large scale instances of combinatorial problems. For example, we ran the TRACONET system with the real-world data presented in Section 1.1.1. The purpose of this experiment was to validate the approach in reducing the total transportation costs among autonomous dispatch centers. Each one of the five agents executed on its own HP 9000 s300 workstation. In 30 minutes, each agent goes through the announce-bid-award loop between 100 and 200 times. Table 3.1 presents results of an example run. The percentual savings are measured compared to the initial routing solutions where each agent operates individually. These local solutions were acquired using a polynomial heuristic algorithm. They also acted as the solutions that the agents used as local solutions when the negotiations began. As can be seen, the negotiations led to considerable transportation cost savings in reasonable time even with such a large problem instance. The results should not be taken as a quantitative indication of how large the savings are because this depends on the problem, the problem instances, the quality of the heuristic algorithm that generates the initial solutions, and even the order in which contracts happen to occur in the asynchronous implementation. Instead, the results should be interpreted as a proof of concept that automated contracting can actually be implemented to operate in the large. Furthermore, it is guaranteed that the solution acquired after any amount of negotiation time is no worse for any agent than its initial solution: participation in the negotiations is individually rational to each agent.

Table 3.1. Experimental results of automated contracting in vehicle routing with real-world data.

Dispatch agent	Cost savings in 15 minutes of negotiation	Cost savings in 30 minutes of negotiation
1	5 %	6 %
2	12 %	18 %
3	31 %	34 %
4	11 %	23 %
5	9 %	15 %
Total	11 %	17 %

In addition to being a proof of concept, the TRACONET system has served to introduce several previously unaddressed questions in automated contracting. We have analyzed many of these in detail in our more recent work. These issues are addressed in the next sections.

3.2 Levels of Commitment

In traditional multiagent negotiation protocols among self-interested agents, once a contract is made, it is binding, i.e. neither party can back out [Rosenschein and Zlotkin, 1994, Sandholm, 1993, Ephrati and Rosenschein, 1991, Kraus, 1993]. Once an agent agrees to a contract, it has to follow through with it no matter how future events unravel. Although a contract may be profitable to an agent when viewed *ex ante*, it need not be profitable when viewed after some future events have occurred, i.e. *ex post*. Similarly, a contract may have too low expected payoff *ex ante*, but in some realizations of the future events, the same contract may be desirable when viewed *ex post*. Normal full commitment contracts are unable to efficiently take advantage of the possibilities that such—probabilistically known—future events provide.

On the other hand, many multiagent systems consisting of cooperative agents incorporate some form of decommitment possibility in order to allow the agents to accommodate new events. For example, in the original Contract Net Protocol [Smith, 1980], the agent that had contracted out a task could send a termination message to cancel the contract even when the contractee had already partially fulfilled the contract. This was possible because the agents were not self-interested: the contractee did not mind losing part of its effort without a monetary compensation. Similarly, the role of decommitment possibilities among cooperative agents has been studied in meeting scheduling using a contracting approach [Sen, 1994, Sen, 1996] and in cooperative coordination protocols [Decker and Lesser, 1995]. Again, the agents did not require a monetary compensation for their efforts: an agent agreed to cancel a contract merely based on the fact that some other agent wanted to decommit. This research was *descriptive*: what will happen if agents use certain externally specified strategies.

Unlike the descriptive approach that is viable among cooperative agents, systems consisting of self-interested agents require that we consider the case where agents do not follow externally specified strategies, but choose their own strategies. Thus the interaction protocols need to be considered from a *normative* perspective: given a protocol, what is the best strategy from a self-interested viewpoint that each agent can choose, and then what social outcomes follow.

Some normative research in game theory has focused on utilizing the potential provided by probabilistically known future events by *contingency contracts* among self-interested agents [Raiffa, 1982]. The obligations of the contract are made contingent on future events. There are games in which this method provides an expected payoff increase to both parties of the contract compared to any full commitment contract. Also, some deals are enabled by contingency contracts in the sense that there is no full commitment contract that both agents prefer over their fallback positions, but there is a contingency contract that each agent prefers over its fallback.

There are at least three problems regarding the use of contingency contracts in automated negotiation among self-interested agents. Though useful in anticipating a small number of key events, contingency contracts get cumbersome as the number of relevant events to monitor from the future increases. In the limit, all domain events (changes in the domain problem, e.g. new tasks arriving or resources breaking down) and all negotiation events—contracts from other negotiations—can affect the value of the obligations of the original contract, and should therefore be conditioned on. Furthermore, these future events may not only affect the value of the original contract independently: the value of the original contract may depend on combinations of the future events [Sandholm and Lesser, 1995c, Sandholm, 1993, Rosenschein and Zlotkin, 1994]. Thus there is a potential combinatorial explosion of events to be conditioned on. Second, even if it were feasible to use such cumbersome contingency contracts among the computerized agents, it is often impossible to enumerate all possible relevant future events in advance. The third problem is that of verifying the unraveling of the events. Sometimes an event is only observable by one of the agents. This agent may have an incentive to lie to the other party of the contract about the event in case the event is associated with an unadvantageous contingency to the directly observing agent. Thus, to be viable, contingency contracts would require an event verification mechanism that is not manipulable and not prohibitively complicated.

We propose another method for taking advantage of the possibilities provided by probabilistically known future events. Instead of conditioning the contract on future events, a mechanism is built into the contract that allows unilateral decommitting at any point in time. This is achieved by specifying in the contract decommitment penalties, one for each agent. If an agent wants to decommit—i.e. to be freed from the obligations of the contract—it can do so simply by paying the decommitment penalty to the other party. We will call such contracts *leveled commitment contracts* because the decommitment penalties can be used to choose a level of commitment. The method requires no explicit conditioning on future events: each agent can do its own conditioning dynamically. Therefore no event verification mechanism is required either. This section of the dissertation presents formal justifications for adding this decommitment feature into a contracting protocol.

Principles for assessing decommitment penalties have been studied in the economics of law [Calamari and Perillo, 1977, Posner, 1977], but the purpose has been to assess a penalty on the agent that has breached the contract *after the breach has occurred.* Similarly, penalty clauses for partial failure—such as not meeting a deadline—are commonly used in contracts, but the purpose is usually to motivate the agents to follow the contract. To my knowledge, the possibility of explicitly allowing decommitment from the whole contract for a predetermined price has not been studied as an active method for utilizing the potential provided by an uncertain future. Somewhat unintuitively, it turns out that the mere existence of a decommitment possibility in a contract can increase each agent's expected payoff.

Intuitively speaking, the goal of our leveled commitment contracting protocol is to allow for some flexibility for future negotiation as in the case with no commitment while guaranteeing agents some level of security as in the total commitment case. The commitment breaking cost can also increase with time, decrease as a function of acceptance time of the offer, or be conditioned on events in other negotiations or the environment. Using the suggested message types (Figure 3.10), the level of commitment can also be dynamically negotiated over on a per contract or per task set basis. There are several qualitative reasons why the use of multiple levels of commitment is a desirable approach:

- It allows agents to profitably accommodate new domain events such as new tasks arriving or resources breaking down by allowing an agent to back out of old contracts that these new events have made undeneficial.

- It allows agents to profitably accommodate new negotiation events such as new offers or acceptance messages. If these events make some old contracts unbeneficial to an agent, that agent can decommit from the old contracts.

- It allows more controlled profitable risk taking. In terms of search this means moving a low commitment search focus around in the global task allocation space (because decommitting is not unreasonably expensive), so that more of that space can be explored among self-interested agents which would otherwise avoid risky commitments. For example, an agent can accept a task set and later try to contract the tasks in that set further separately. With full commitment, an agent needs to have standing offers from the agents it will contract the tasks to, or it has to be able to handle them itself. With the leveled commitment protocol, the agent can accept the task set even if it is not sure about its chances of getting it handled, because in the worst case it can decommit.

- It enhances the negotiations computationally. For example, an agent can make the same low-commitment offer (or offers that overlap in task sets) to multiple agents. In case more than one accepts, the agent has to pay the penalty to all but one of them, but the speedup of being able to address multiple agents in committal mode may outweigh this risk.

- It allows flexibility to the agent's local deliberation control, because marginal cost calculation of a contract can go on even after the contract has already been agreed upon. If the contract turns out unbeneficial in the extended deliberation, the agent can decommit.

- It allows profitable construction of composite contracts from basic contracts. In Section 3.5 we present three types of composite operators that are desirable in contracting: clustering, swaps and multiagent contracts. All of these involve contracting over more than one task at a time atomically. On the other hand, the leveled commitment protocol allows any of these composite contracts to be constructed from a sequence of basic contracts. For example, say that the only profitable contract is one where agent A gives task t_1 to agent B, and agent C gives task t_2 to agent A. Now A can make the unprofitable contract with B first in anticipation of the contract with C which will make the combination profitable. Then, if C does not agree to the contract with A, agent A can back out of the contract that it made with B.

- It allows the agents with a lesser risk aversion to carry a greater portion of the risk. The more risk averse agent can trade off paying a higher price to its contractee (or get paid a lower price as a contractee) for being allowed to have a lower decommitting penalty. This increases social welfare.

- It allows beneficial contingency contracts by conditioning the payments and commitment functions on future negotiation events or domain events. These enlarge the set of mutually beneficial contracts, when agents have different expectations of future events or different risk attitudes [Raiffa, 1982].

In the rest of this section, we substantiate the advantages of leveled commitment contracts more formally.

We analyze contracting situations from the perspective of two agents: the *contractor* who pays to get a task done, and the *contractee* who gets paid for handling the task. Handling a task can mean taking on any types of constraints, i.e. the method is not specific to classical task allocation domains. The contractor tries to minimize the contract price ρ that it has to pay. The contractee tries to maximize the payoff ρ that it receives from the contractor. Outside offers from third parties will be explicitly discussed. The contracting setting consists of two games. First, the *contracting game* involves the agents choosing a contract—or no contract, i.e. the *null deal*—before any future events have unraveled. Secondly, the *decommitting game* involves the agents deciding on whether to decommit or to carry out the obligations of the contract—after the future events have unraveled. The decommitment game is a subgame of the contracting game: the expected outcomes of the decommitting game affect the agents' preferences over contracts in the contracting game. The decommitting game will be analyzed using the Nash equilibrium and the dominant strategy concepts. The contracting game will be analyzed with respect to *individual rationality* (IR): is the contract better for an agent than the null deal?

Often there is either no contract that is IR for both agents or then there are many such contracts. When there are many IR contracts to choose from, there are infinitely many Nash equilibria in the contracting game. If the contractor's strategy is to offer a contract for price ρ and no more (and that contract is IR for both agents), the contractee's best response is to take the offer as opposed to the null deal. Now the contractor's best response to this is to offer ρ and no more. Thus, a Nash equilibrium exists for any ρ that defines a contract that is IR for both agents. Actually, *axiomatic bargaining theory* [Nash, 1950a, Osborne and Rubinstein, 1990]studies the question of choosing among these IR contracts by asserting desirable properties that the chosen contract should fulfill compared to the other contracts.

In Sections 3.2.1 and 3.2.2 we analyze the advantage of the leveled commitment contracting protocol compared to the full commitment one using different contracting settings. These sections and their subsections are ordered so that simpler settings precede the more complex ones. Section 3.2.1 describes settings where only one agent's future involves uncertainty, while Section 3.2.2 describes settings where both agents' futures involve uncertainty. The symbols used in the upcoming sections are summarized in Table 3.2. In Section 3.2.3 some practical prescriptions are given to builders of automated negotiation systems.

Table 3.2. Symbols used in this section. We restrict our analysis to contracts where $a \geq 0$ and $b \geq 0$, i.e. we rule out contracts that specify that the decommitting agent receives a payment from the victim of the decommitment.

ρ	Contract price.
$a \geq 0$	Contractor's decommitment penalty.
$b \geq 0$	Contractee's decommitment penalty.
$\breve{a}$	Price of the contractor's best (full commitment) outside offer.
$\breve{b}$	Price of the contractee's best (full commitment) outside offer.
$f(\breve{a})$	*Ex ante* probability density function of $\breve{a}$.
$g(\breve{b})$	*Ex ante* probability density function of $\breve{b}$.
p_a	Probability that the contractor decommits.
p_b	Probability that the contractee decommits.
$\bar{b}$	Contractee's fallback, i.e. payoff that it gets if no contract is made.
B	Big positive number.

3.2.1 Uncertainty about One Agent's Outside Offer

This section presents games where one agent's outside offer is fixed and known to both agents at contract time but uncertainty prevails about the price of the other agent's outside offer. Let the contractee have the deterministically known outside offer $\breve{b}$. Let the contractor's (best) outside offer be only known probabilistically by the agents via a probability density function $f(\breve{a})$. In case the contractor receives no outside offer, $\breve{a}$ is the best of its outstanding outside offers and its fallback payoff. The case where the contractor's outside offer is deterministically known but the contractee's outside offer is only probabilistically known is analogous. The contract is made when the contractee's outside offer is known but the contractor's is not. On the other hand, the decommitting game takes place when the contractor has found out the value of $\breve{a}$. We assume that at this point the contractee does not know the contractor's outside offer $\breve{a}$. This seems realistic in automated contracting systems. Now there are two cases depending on whether the contractee's outside offer stays valid up to the point when the contractor finds out about its outside offer—i.e. up to when the contractor decides between decommitting and following through with the obligations of the contract.

3.2.1.1 Deterministic Offer Prevails (DOP)

In this case, the contractee's outside offer stays valid up to the point when the contractor finds out about its outside offer. So if the contractor decides to decommit, the contractee can still accept the outside offer. We will call this situation the DOP game, Figure 3.1.

In a sequential decommitting DOP game where the contractee reveals decommitment first (Figure 3.1), because the contractee gains no information between the beginning of the contracting game and the decommitting game, it will not find decommitting beneficial (for any $b \geq 0$) if it found the original contract beneficial (better than its outside offer $\breve{b}$) and thus agreed to it.

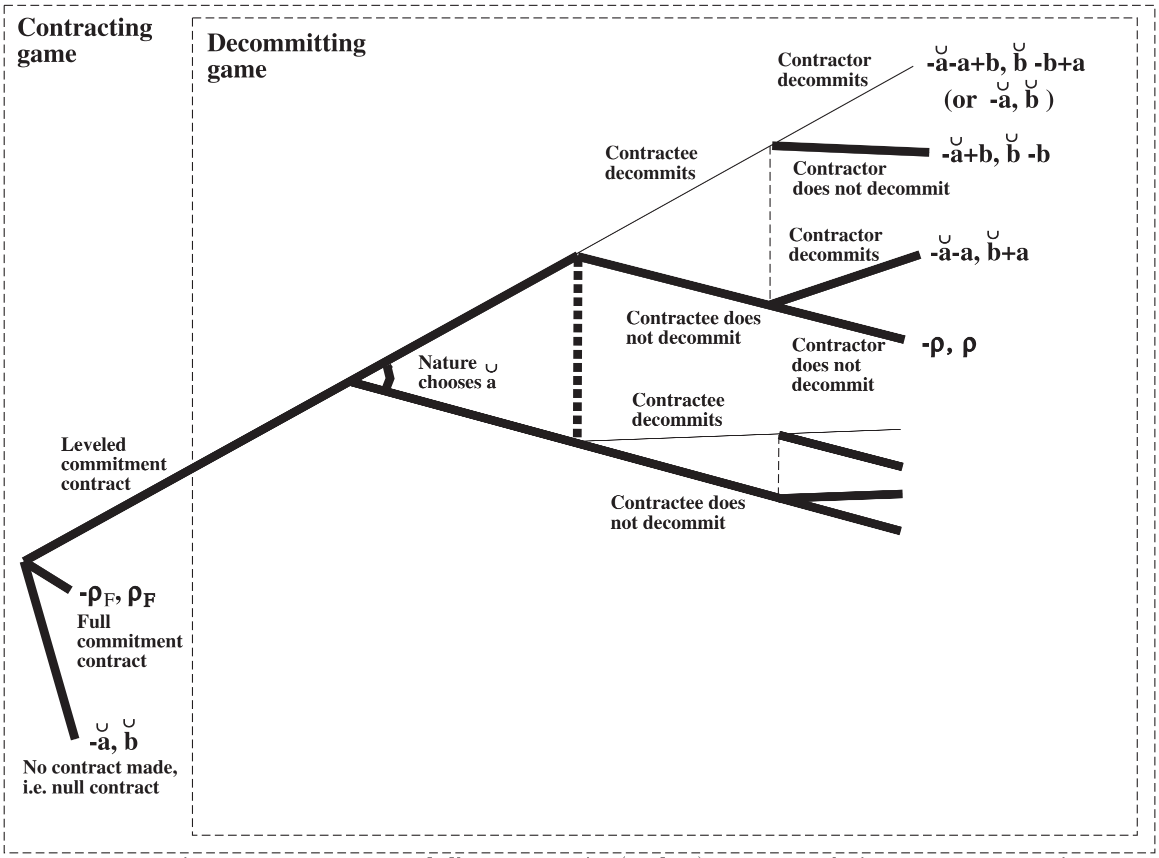

Figure 3.1. The Deterministic Offer Prevails (DOP) game. If the contractor decommits, the contractee can still accept the outside offer. The contractor's payoff is listed before the contractee's. The bold solid lines show choices that may actually occur in any subgame. The bold dashed line represents the contractee's information set: it does not know $\breve{a}$ in the decommitting game. The thin dashed lines represent the alternative situation where both agents reveal decommitment simultaneously: when deciding on decommitting, the contractor has not observed whether the contractee decommitted.

This holds even for a game where the agents reveal decommitting simultaneously as opposed to contractee first (this game is depicted by the information sets denoted by thin dashed lines in Figure 3.1).

Even in a sequential decommitting game where the contractor moves first, the contractee will not want to decommit. If the contractor decommits, the contractee can save its decommitment penalty by not declaring decommitment and the contract becomes void anyway. Now let us discuss the branches where the contractor did not decommit. In these branches the contractee's payoff is independent of the contractor's outside offer, and thus all of these branches are equivalent. Now, if the contractee would be better off decommitting in such a branch, the contractor would know that. Therefore the contractor would never decommit (i.e. no matter what its outside offer turns out to be). Thus the decommitting game would always be played by the contractor not decommitting, and the contractee decommitting. Clearly this kind of a contracting game is not IR for the contractee. Therefore the contractee never decommits.

Even if a protocol is used that specifies that neither agent has to pay the decommitment penalty if both decommit (payoffs in parentheses in Figure 3.1), the contractee wants to decommit in none of the three cases above. Thus the only agent to possibly make a move in the decommitting game is the contractor. In any one of the above three settings, the contractor can reason that the contractee will not decommit. Therefore the three cases become equivalent. This holds for the protocol that specifies that both have to pay if both decommit and for the protocol that specifies that neither has to pay if both decommit.

The contractor's cost is ρ if it does not decommit, and $\breve{a} + a$ if it does. In other words the contractor's payoff is $-\rho$ or $-\breve{a} - a$. Therefore, the contractor will decommit if $\breve{a} + a < \rho$. Thus the probability that the contractor will decommit is

$$p_a = \int_{-\infty}^{\rho - a} f(\breve{a}) d\breve{a}$$

The contractee's individual rationality (IR) constraint states that the contract has to have higher expected payoff than the fixed outside offer:

$$\breve{b} \leq [1 - p_a]\rho + p_a[\breve{b} + a]$$

The contractor can choose—*ex post*—whether it wants to decommit or stay with the contract. Therefore the contractor's *ex ante* IR constraint is based on the idea that $E[-\breve{a}] \leq E[max[-\breve{a} - a, -\rho]]$:

$$\int_{-\infty}^{\infty} f(\breve{a})[-\breve{a}] d\breve{a} \leq \int_{-\infty}^{\rho - a} f(\breve{a})[-\breve{a} - a] d\breve{a} + \int_{\rho - a}^{\infty} f(\breve{a})[-\rho] d\breve{a}$$

Now that the game has been specified, the advantages of a leveled commitment contract can be analyzed. Obviously in this game the full commitment contracts are a subset of leveled commitment ones because a leveled commitment contract can emulate a full commitment one by choosing a high decommitment penalty $a = B$ that motivates the contractor to surely not decommit (assuming that $\breve{a}$ is bounded from below). Therefore, for any full commitment contract, there exists a leveled commitment contract that has no worse payoff to either agent. Furthermore, the following two theorems state the strict superiority of leveled commitment contracts over full commitment ones in this game. The first theorem states that in some DOP games the agents cannot make a beneficial contract under the full commitment protocol, but can under the leveled commitment one.

Theorem 3.1 (Enabling in DOP games) *There are DOP games (defined by $\breve{b}$ and $f(\breve{a})$) where no full commitment contract satisfies both IR constraints but where a leveled commitment contract satisfies both IR constraints.*

Proofs of all but the most intuition enhancing theorems are postponed until an appendix at the end of this dissertation.

The next theorem states that even if both protocols would allow a beneficial contract, the leveled commitment protocol may allow higher expected payoffs for both agents than the full commitment protocol. This holds as long as there is some chance that the contractor's outside offer will be lower than the contractee's.

Theorem 3.2 (Pareto efficiency improvement in DOP games) *Let F be an arbitrary full commitment contract that satisfies both IR constraints, i.e. $\breve{b} \leq \rho_F \leq E[\breve{a}]$. Let f be bounded, and $\int_{-\infty}^{\breve{b}} f(\breve{a})d\breve{a} > 0$. Now there exists a leveled commitment contract that increases the contractor's payoff and the contractee's payoff (and thus also satisfies the IR constraints).*

It follows that under the conditions of the theorem, no full commitment contract is Pareto efficient.

The following theorem shows the desirable property that an agent cannot be hurt by its negotiation partner's biased beliefs in a DOP game. For any specific contract, an agent with precise information has an expected payoff of what it thinks it has independent of the other agent's reasoning process or information sources. Thus an agent need not counterspeculate its negotiation partner's beliefs. In the theorem, let $f_a(\breve{a})$ be the contractor's belief of $f(\breve{a})$, and let $f_b(\breve{a})$ be the contractee's belief of $f(\breve{a})$.

Theorem 3.3 (Payoff unaffected by opponent's beliefs in DOP games) *Say that one agent's information is unbiased, i.e. either $f_a(\breve{a}) = f(\breve{a})$ or $f_b(\breve{a}) = f(\breve{a})$. Now that agent's expected payoffs for contracts are unaffected by the possible biases of the other agent's information. Thus the former agent's preference ordering over contracts is unaffected.*

3.2.1.2 Certain Offer Becomes Void (COBV)

This section discusses the setting where the contractee has a fixed outside offer $\breve{b}$, but this offer has to be accepted before the contractor finds out the price of its (best) outside offer $\breve{a}$ (in case the contractor receives no outside offer, $\breve{a}$ is its fallback payoff). Otherwise the $\breve{b}$-offer becomes void. Thus, to agree to the contract, the contractee has to have a higher expected payoff when passing on the $\breve{b}$-offer and agreeing to the risky contract than by accepting the $\breve{b}$-offer. If the contract is made, decommitment occurs—if at all—when the contractor's outside offer is valid (and known to the contractor but not to the contractee) but the contractee's is not anymore. In this case the contractee gets its fallback payoff $\bar{b}$ plus the contractor's decommitment penalty payment a. The fallback $\bar{b}$ can be interpreted for example as the contractee's second best outside offer (best that is still available) or—in case no outside offers are outstanding—as the contractee's payoff without any contracts. We will call the setting the COBV game, Figure 3.2.

In a sequential decommitting COBV game where the contractee reveals decommitment first (Figure 3.2), because the contractee gains no information between the

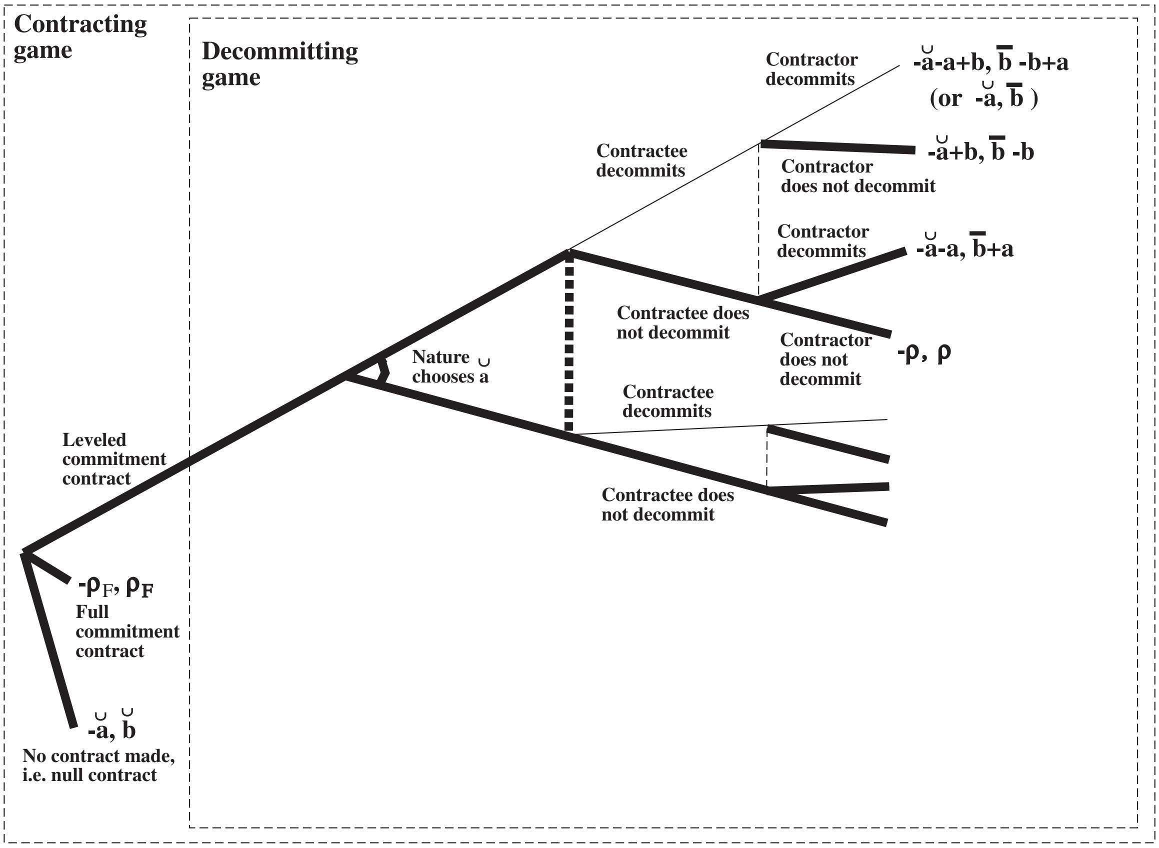

Figure 3.2. The "Certain Offer Becomes Void" (COBV) game. If at all, the contractee's outside offer $\breve{b}$ has to be accepted before the contractor's outside offer $\breve{a}$ becomes known. The bold solid lines show choices that may actually occur in any subgame. The bold dashed line represents the contractee's information set: it does not know $\breve{a}$ in the decommitting game. The thin dashed lines represent the alternative situation where both agents reveal decommitment simultaneously: when deciding on decommitting, the contractor has not observed whether the contractee decommitted.

beginning of the contracting game and the decommitting game, it will not find decommitting beneficial (for any $b \geq 0$) if it found the original contract beneficial (better than its outside offer $\breve{b}$) and thus agreed to it.

This holds even for a game where the agents reveal decommitting simultaneously as opposed to contractee first (this game is depicted by the information sets denoted by thin dashed lines in Figure 3.2).

Even in a sequential decommitting game where the contractor moves first, the contractee will not want to decommit. If the contractor decommits, the contractee can save its decommitment penalty by not declaring decommitment and the contract becomes void anyway. Now let us discuss the branches where the contractor did not decommit. In these branches the contractee's payoff is independent of the contractor's outside offer, and thus all of these branches are equivalent. Now, if the contractee would be better off decommitting in such a branch, the contractor would know that. Therefore the contractor would never decommit (i.e. no matter what its outside offer turns out to be). Thus the decommitting game would always be played by the contractor not decommitting, and the contractee decommitting. Clearly this kind of a contracting game is not IR for the contractee. Therefore the contractee never decommits.

Even if a protocol is used that specifies that neither agent has to pay the decommitment penalty if both decommit (payoffs in parentheses in Figure 3.2), the contractee wants to decommit in none of the three cases above. Thus the only agent to possibly make a move in the decommitting game is the contractor. In any one of the above three settings, the contractor can reason that the contractee will not decommit. Therefore the three cases become equivalent. This holds for the protocol that specifies that both have to pay if both decommit and for the protocol that specifies that neither has to pay if both decommit.

The contractor's cost is ρ if it does not decommit, and $\breve{a} + a$ if it does. Therefore, the contractor will decommit if $\breve{a} + a < \rho$. Thus,

$$p_a = \int_{-\infty}^{\rho - a} f(\breve{a}) d\breve{a}$$

The contractee's IR constraint is

$$\breve{b} \leq [1 - p_a]\rho + p_a[\bar{b} + a]$$

where $\bar{b}$ is the contractee's fallback position, i.e. the payoff it gets if it does not get its outside offer $\breve{b}$ or the contract with the contractor. Note that if $\bar{b} = \breve{b}$, this situation reduces to a DOP game (Section 3.2.1.1), and thus the DOP results in favor of leveled commitment hold. Naturally they also hold if $\bar{b} > \breve{b}$ because this can only make the risky leveled commitment contract more desirable to the contractee—without affecting the desirability to the contractor.

The contractor's IR constraint is based on the idea that *ex post*, the contractor can choose whether it wants to decommit or stay with the contract. *Ex post*, the contractor finds the contract individually rational if $-\breve{a} \leq max[-\breve{a} - a, -\rho] \Leftrightarrow \breve{a} \geq \rho$. Thus the (*ex ante*) IR constraint is

$$\int_{-\infty}^{\infty} f(\breve{a})[-\breve{a}]d\breve{a} \leq \int_{-\infty}^{\rho-a} f(\breve{a})[-\breve{a} - a]d\breve{a} + \int_{\rho-a}^{\infty} f(\breve{a})[-\rho]d\breve{a}$$

Full commitment contracts are a subset of leveled commitment ones because the contractor's decommitment penalty can be chosen so high ($a = B$) that the contractor will surely not decommit (assuming that $\breve{a}$ is bounded from below). As discussed earlier, the contractee will not decommit either for any $b \geq 0$. Thus, the class of leveled commitment contracts is no worse than the class of full commitment ones. Although there are games where a leveled commitment contract is Pareto superior to any full commitment contract if the contractee's fallback is sufficiently high (a COBV game reduces to a DOP game if $\bar{b} = \breve{b}$), the following two theorems show that if the contractee's fallback is too low, leveled commitment contracts are not helpful in COBV games.

Theorem 3.4 (No enabling for low fallbacks in COBV games) *Let us restrict to COBV games where* $\bar{b} \leq \frac{\int_{-\infty}^{\rho-a} f(\breve{a})\breve{a}d\breve{a}}{\int_{-\infty}^{\rho-a} f(\breve{a})d\breve{a}}$. *In such a game (defined by* $\breve{b}$, $f(\breve{a})$, *and* $\bar{b}$*), if no full commitment contract satisfies the IR constraints, no leveled commitment contract satisfies them either.*

The constraint $\bar{b} \leq \frac{\int_{-\infty}^{\rho-a} f(\breve{a})\breve{a}d\breve{a}}{\int_{-\infty}^{\rho-a} f(\breve{a})d\breve{a}}$ is satisfied for example if $\forall \breve{a} \leq 0, f(\breve{a}) = 0$, and $\bar{b} \leq 0$. This means that the contractor's outside offer will require some nonnegative payment to do the contractor's task, and that the contractee has a nonpositive fallback. The former requirement does not seem very restrictive, but the latter does. This suggests that these two theorems with negative results may have limited scope.

Theorem 3.5 (No Pareto improvement for low fallbacks in COBV games) *Let us restrict to COBV games where* $\bar{b} \leq \frac{\int_{-\infty}^{\rho-a} f(\breve{a})\breve{a}d\breve{a}}{\int_{-\infty}^{\rho-a} f(\breve{a})d\breve{a}}$. *Let* F *be an arbitrary full commitment contract that satisfies both IR constraints, i.e.* $\breve{b} \leq \rho_F \leq E[\breve{a}]$. *Now there exists no leveled commitment contract that increases (over* F*) at least one agent's expected payoff without decreasing the other agent's expected payoff.*

Now we will discuss COBV games where the agents' beliefs differ. Specifically, let $f_a(\breve{a})$ be the contractor's belief of $f(\breve{a})$, and let $f_b(\breve{a})$ be the contractee's belief of $f(\breve{a})$, and let everything else be common knowledge. Now the contractee's *perceived individual rationality* (PIR) constraint is

$$\breve{b} \leq [1 - (\int_{-\infty}^{\rho-a} f_b(\breve{a})d\breve{a})]\rho + (\int_{-\infty}^{\rho-a} f_b(\breve{a})d\breve{a})[\bar{b} + a]$$

Similarly, the contractor's PIR constraint is

$$\int_{-\infty}^{\infty} f_a(\breve{a})[-\breve{a}]d\breve{a} \leq \int_{-\infty}^{\rho-a} f_a(\breve{a})[-\breve{a} - a]d\breve{a} + \int_{\rho-a}^{\infty} f_a(\breve{a})[-\rho]d\breve{a}$$

The following theorem shows that even though no contract is beneficial to the agents, and no full commitment contract seems beneficial, both agents may perceive that some leveled commitment contract is beneficial.

Theorem 3.6 (Perceived enabling in COBV games) *There are games (defined by* $f(\breve{a})$, $f_a(\breve{a})$, $f_b(\breve{a})$, $\bar{b}$, *and* $\breve{b}$*) where no full commitment contract satisfies both IR constraints, no leveled commitment contract satisfies both IR constraints, no full commitment contract satisfies both PIR constraints, but some leveled commitment contract satisfies both PIR constraints.*

So the agents only *perceive* that this leveled commitment contract satisfies their individual rationality constraints. This is due to the fact that at least one agent's estimate of the distribution of the contractor's outside offer is biased. On the other hand, if the contractee's fallback payoff is sufficiently low, both agents know (by

Theorem 3.4) that the contract cannot really be IR for both. Now which agent is going to incur the loss if the agents agree to the contract that is perceived IR by both? The following positive result states that an agent with unbiased beliefs has an expected payoff of what it thinks it has independent of the other agents beliefs (stemming from a reasoning process or information sources). Thus the unbiased agent will not enter an unprofitable (non-IR) contract due to the other agent's biases. It also means that agents need not counterspeculate their negotiation partner's beliefs.

Theorem 3.7 (Payoff unaffected by opponent's beliefs in COBV games) *Say that one agent's information is unbiased, i.e. either* $f_a(\breve{a}) = f(\breve{a})$ *or* $f_b(\breve{a}) = f(\breve{a})$*. Now that agent's expected payoffs for contracts are unaffected by the possible biases of the other agent's information. Thus the former agent's preference ordering over contracts and the null deal is unaffected.*

Corollary 3.1 (Perceived IR is IR for unbiased agent in COBV games) *Say that at most one agent's information is biased, i.e. either* $f_a(\breve{a}) = f(\breve{a})$ *or* $f_b(\breve{a}) = f(\breve{a})$*. Say that the contract is perceived IR by the agent* x *for which* $f_x(\breve{a}) = f(\breve{a})$*. Now, the contract really is IR for that agent.*

It follows that if a contract is perceived IR by both agents, but really is not, the contract is really IR for the agent with unbiased beliefs but not for the agent with biased beliefs about $f(\breve{a})$.

3.2.2 Uncertainty about Both Agents' Outside Offers

This section discusses a contracting setting where the future of both agents involves uncertainty. Specifically, both agents—contractor and contractee—might receive outside offers. The contractor's best outside offer $\breve{a}$ is only probabilistically known *ex ante* by both agents, and is characterized by a probability density function $f(\breve{a})$. If the contractor does not receive an outside offer, $\breve{a}$ corresponds to its best outstanding outside offer or its fallback payoff, i.e. payoff that it receives if no contract is made. The contractee's best outside offer $\breve{b}$ is also only probabilistically known *ex ante*, and is characterized by a probability density function $g(\breve{b})$. If the contractee does not receive an outside offer, $\breve{b}$ corresponds to its best outstanding outside offer or its fallback payoff. The variables $\breve{a}$ and $\breve{b}$ are assumed statistically independent. The contractor's options are either to make a contract with the contractee or to wait for $\breve{a}$. Similarly, the contractee's options are either to make a contract with the contractor or to wait for $\breve{b}$. The two agents have many mutual contracts to choose from. A leveled commitment contract is specified by the contract price ρ, the contractor's decommitment penalty a, and the contractee's decommitment penalty b. The agents also have the possibility to make a full commitment contract. The contractor has to decide on decommitting when it knows its outside offer $\breve{a}$ but does not know the contractee's outside offer $\breve{b}$. Similarly, the contractee has to decide on decommitting when it knows its outside offer $\breve{b}$ but does not know the contractor's outside offer $\breve{a}$. This seems realistic from a practical automated contracting perspective.

The DOP game that was discussed in Section 3.2.1.1 is a special case of this type of games. In the DOP game, all of the probability mass of $g(\breve{b})$ is on one value $\breve{b}$. The DOP game is also a special case of the COBV game described in Section 3.2.1.2—when $\bar{b} = \breve{b}$. On the other hand, COBV games are not a subset of this type of games: in COBV games some opportunity (outside offer) may be missed due to waiting for the unraveling of the new outside offers.

We do not assume that the agents decommit truthfully. For example, an agent may not decommit although its outside offer is better for itself than the contract because the agent believes that there is a high probability that the opponent will decommit. This would save the agent its decommitment penalty and in fact make the agent receive a decommitment penalty from the opponent. Games of this type differ significantly based on whether the agents have to decommit sequentially or simultaneously. The next two sections (3.2.2.1 and 3.2.2.2) detail these cases.

3.2.2.1 Sequential Decommitting (SEQD)

In our sequential decommitting (SEQD) game, one agent has to declare decommitment before the other agent. We will study the case where the contractee has to decommit first. The case where the contractor has to decommit first is analogous. The game tree is presented in Figure 3.3. There are two alternative types of leveled commitment contracts that differ on what happens if both agents decommit. In the first, both agents have to pay the decommitment penalties to each other if both decommit. In the second, neither agent has to pay if both decommit.

Let us now analyze the decommitting game using dominance in subgames as the solution concept. Specifically, we start reasoning about the agents' actions at the leaves of the game tree and proceed backwards to the beginning of the game. In the subgame where the contractee has decommitted, the contractor's best move is not to decommit because $-\breve{a} - a + b \leq -\breve{a} + b$ (because $a \geq 0$). This also holds for a contract where neither agent has to pay a decommitment penalty if both decommit—because $-\breve{a} \leq -\breve{a} + b$, (Fig. 3.3 parenthesized payoffs). In the subgame where the contractee has not decommitted, the contractor's best move is to decommit if $-\breve{a} - a > -\rho$. This happens with probability $\int_{-\infty}^{\rho-a} f(\breve{a})d\breve{a}$. Put together, the contractee gets $\breve{b} - b$ if it decommits, $\breve{b} + a$ if it does not but the contractor does, and ρ if neither decommits. Thus the contractee decommits if

$$\breve{b} - b > \int_{-\infty}^{\rho-a} f(\breve{a})d\breve{a}[\breve{b} + a] + \int_{\rho-a}^{\infty} f(\breve{a})d\breve{a}[\rho]$$

If $\int_{\rho-a}^{\infty} f(\breve{a})d\breve{a} = 0$, this is equivalent to $-b > a$ which is false because a and b are nonnegative. In other words, if the contractee surely decommits, the contractor does not. On the other hand, the above is equivalent to

$$\breve{b} > \rho + \frac{b + \int_{-\infty}^{\rho-a} f(\breve{a})d\breve{a}[a]}{\int_{\rho-a}^{\infty} f(\breve{a})d\breve{a}} \stackrel{\text{def}}{=} \breve{b}^*(\rho, a, b) \text{ when } \int_{\rho-a}^{\infty} f(\breve{a})d\breve{a} > 0 \qquad (3.7)$$

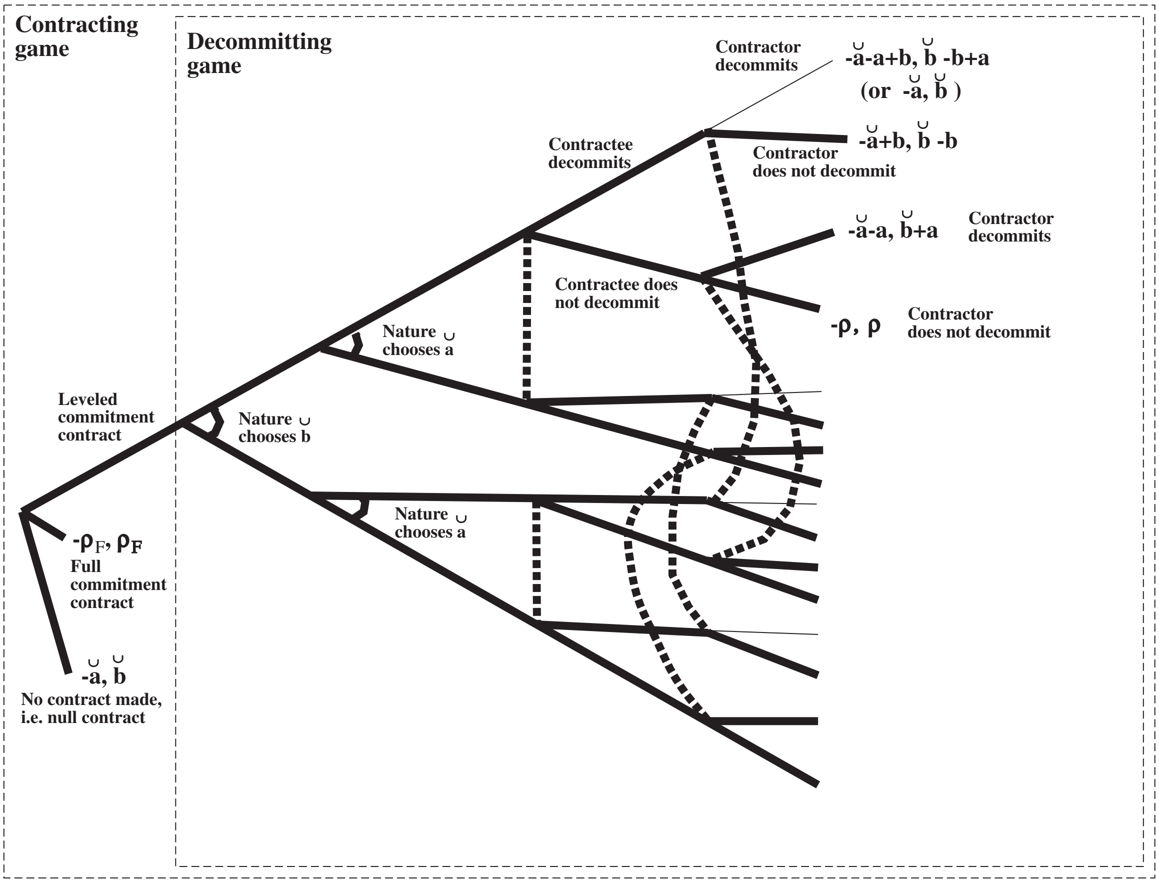

Figure 3.3. The "SEQuential Decommitting" (SEQD) game. The game tree represents two alternative protocols, i.e. two different games. In the first, both agents have to pay the decommitment penalties to each other if both decommit. In the second, neither agent has to pay if both decommit. The payoffs of the latter protocol are in parentheses when they differ from the former. Dotted lines show information sets: the contractor does not know the contractee's outside offer and vice versa. The contractor's payoffs are usually negative because it has to pay for having the task handled.

Now the contractee's IR constraint states that the expected payoff from the contract is no less than the expected payoff from the outside offer:

$$\begin{aligned} \pi_b &= \int_{\breve{b}^*(\rho,a,b)}^{\infty} g(\breve{b})[\breve{b}-b]d\breve{b} + \int_{-\infty}^{\breve{b}^*(\rho,a,b)} g(\breve{b})[\int_{-\infty}^{\rho-a} f(\breve{a})[\breve{b}+a]d\breve{a} + \int_{\rho-a}^{\infty} f(\breve{a})\rho d\breve{a}]d\breve{b} \\ &\geq E[\breve{b}] = \int_{-\infty}^{\infty} g(\breve{b})\breve{b}d\breve{b} \end{aligned} \tag{3.8}$$

Similarly, the contractor's IR constraint states that the expected payoff from the contract is no less than that from the outside offer:

$$\begin{aligned}\pi_a &= \int_{\breve{b}^*(\rho,a,b)}^{\infty} g(\breve{b}) \int_{-\infty}^{\infty} f(\breve{a})[-\breve{a}+b]d\breve{a}d\breve{b} \\ &+ \int_{-\infty}^{\breve{b}^*(\rho,a,b)} g(\breve{b})[\int_{-\infty}^{\rho-a} f(\breve{a})[-\breve{a}-a]d\breve{a} + \int_{\rho-a}^{\infty} f(\breve{a})[-\rho]d\breve{a}]d\breve{b} \\ &\geq E[-\breve{a}] = \int_{-\infty}^{\infty} f(\breve{a})[-\breve{a}]d\breve{a} \qquad (3.9)\end{aligned}$$

Now, do SEQD games exist where some full commitment contract is possible but no leveled commitment contract is? Because the contractor can want to decommit only if $-\breve{a} - a > -\rho$, its decommitment penalty can be chosen so high ($a = B$) that it will surely not decommit (assuming that $\breve{a}$ is bounded from below). In this case the contractee will decommit whenever $\rho < \breve{b} - b$. If $\breve{b}$ is bounded from above, the contractee's decommitment penalty can be chosen so high ($b = B$) that it will surely not decommit. Thus, full commitment contracts are a subset of leveled commitment ones. This reasoning holds for contracts where both agents have to pay the penalties if both decommit, and for contracts where neither agent has to pay a penalty if both decommit. Because full commitment contracts are a subset of leveled commitment contracts, the former can be no better in the sense of Pareto efficiency or social welfare than the latter. It follows that if there exists an IR full commitment contract, then there also exist IR leveled commitment contracts.

In addition to leveled commitment contracts never being worse than full commitment ones in SEQD games, they can enable a deal that is impossible via full commitment contracts:

Theorem 3.8 (Enabling in SEQD games) *There are SEQD games (defined by $f(\breve{a})$ and $g(\breve{b})$) where no full commitment contract satisfies the IR constraints but a leveled commitment contract does.*

The proof of Theorem 3.8 is constructive and uses the following example game: $f(\breve{a}) = \begin{cases} \frac{1}{100} & \text{if } 0 \leq \breve{a} \leq 100 \\ 0 & \text{otherwise} \end{cases}$ and $g(\breve{b}) = \begin{cases} \frac{1}{110} & \text{if } 0 \leq \breve{b} \leq 110 \\ 0 & \text{otherwise.} \end{cases}$ Actually, in this game, both IR constraints are satisfied by a wide range of leveled commitment contracts—and for no full commitment contract. Which leveled commitment contracts defined by ρ, a, and b satisfy the IR constraints? There are many values of ρ for which some a and b exist such that the constraints are satisfied. As in the proof of Theorem 3.8, let us analyze contracts where $\rho = 52.5$ as an example. Now which values of a and b satisfy both IR constraints? There are three qualitatively different cases.

Case 1. Some chance that either agent is going to decommit. In the case where $a < \rho$ there is some chance that the contractor will decommit (it may happen that $-\breve{a} > -\rho + a$). Now $\breve{b}^*(\rho, a, b) = \rho + \frac{b + \int_{-\infty}^{\rho-a} f(\breve{a}) d\breve{a}[a]}{\int_{\rho-a}^{\infty} f(\breve{a}) d\breve{a}} = \rho + \frac{b + \frac{1}{100}[\rho-a]a}{\frac{1}{100}[100-(\rho-a)]}$. If $\breve{b}^*(\rho, a, b) < 110$ (i.e. less than the maximum possible $\breve{b}$), there is some chance that the contractee will decommit. This occurs if $\breve{b} > \rho + b$. We programmed a model of the IR constraints (Equations 3.9 and 3.8) for this case. To make the algebra tractable (constant $f(\breve{a})$ and $g(\breve{b})$), versions of these IR constraint equations were used that assumed $0 \leq a < \rho$, and $0 < \breve{b}^* < 110$, without loss of generality. The corresponding decommitment penalties a and b that satisfy the IR constraints are plotted in Figure 3.4 left. Furthermore, the boundaries of the programmed model need to be checked. The boundaries $a = 0$, $a = \rho$, and $\breve{b}^* = 110$ are plotted in Figure 3.4 left. The constraint $\breve{b}^* > 0$ is always satisfied in this case and is thus not plotted. To summarize, in the gray area of Figure 3.4 left, the contracts are IR for both agents, given that the agents decommit optimally based on self-interest—not necessarily truthfully.

Case 2, Contractor will surely not decommit. When $a \geq \rho$, the contractor will surely not decommit because its best possible outside offer is $\breve{a} = 0$. Note that a can be arbitrarily high. The corresponding $\breve{b}^*(\rho, a, b) = \rho + \frac{b + \int_{-\infty}^{\rho-a} f(\breve{a}) d\breve{a}[a]}{\int_{\rho-a}^{\infty} f(\breve{a}) d\breve{a}} = \rho + b$, i.e. the contractee decommits truthfully. Now the contractor's IR constraint (Eq. 3.9) becomes

$$\int_{\rho+b}^{110} g(\breve{b}) \int_0^{100} f(\breve{a})[-\breve{a} + b] d\breve{a} d\breve{b} + \int_0^{\rho+b} g(\breve{b}) \int_0^{100} f(\breve{a})[-\rho] d\breve{a} d\breve{b} \geq E[-\breve{a}] \qquad (3.10)$$

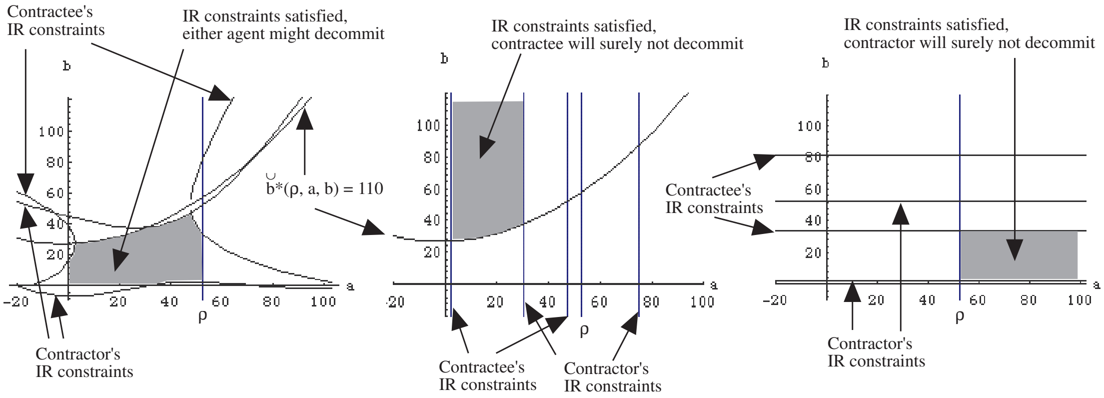

Figure 3.4. The decommitment penalties a and b that satisfy both agents' IR constraints in the example SEQD game. Right: case where either agent might decommit ($a < \rho$, and $\breve{b}^*(\rho, a, b) < 110$). Middle: case where the contractor might decommit but the contractee will not ($a < \rho$, and $\breve{b}^*(\rho, a, b) \geq 110$). Left: case where $a \geq \rho$, i.e. the contractor will surely not decommit but the contractee might.

If $\rho + b \geq 110$, this is equivalent to $-\rho \geq E[-\breve{a}]$ which is false. If $0 < \rho + b < 110$, this is equivalent to

$$\begin{aligned} & \frac{1}{110}\frac{1}{100}[(110 - (\rho + b)) \cdot (\frac{-(100)^2}{2} + 100b) + (\rho + b) \cdot (-100\rho)] \geq E[-\breve{a}] \\ \Leftrightarrow \quad & \frac{1}{110}\frac{1}{100}[(57.5 - b) \cdot (-5000 + 100b) + (52.5 + b) \cdot (-5250)] \geq -50 \\ \Leftrightarrow \quad & 2.5 \leq b \leq 52.5 \end{aligned}$$

by the quadratic equation solution formula.

Similarly, the contractee's IR constraint (Eq. 3.8) becomes

$$\int_{\rho+b}^{110} g(\breve{b}) \int_0^{100} f(\breve{a})[\breve{b} - b] d\breve{a} d\breve{b} + \int_0^{\rho+b} g(\breve{b}) \int_0^{100} f(\breve{a})[\rho] d\breve{a} d\breve{b} \geq E[\breve{b}] \tag{3.11}$$

If $\rho + b \geq 110$, this is equivalent to $\rho \geq E[\breve{b}]$ which is false. If $0 < \rho + b < 110$, this is equivalent to

$$\begin{aligned} & \int_{\rho+b}^{110} g(\breve{b})[\breve{b} - b] \int_0^{100} f(\breve{a}) d\breve{a} d\breve{b} + \int_0^{\rho+b} g(\breve{b})[\rho] \int_0^{100} f(\breve{a}) d\breve{a} d\breve{b} \geq E[\breve{b}] \\ \Leftrightarrow \quad & \frac{1}{110}\frac{1}{100}[(\frac{110^2}{2} - 110b - (\frac{(\rho + b)^2}{2} - (\rho + b)b)) \cdot 100 + (\rho + b)\rho \cdot 100] \geq 55 \\ \Leftrightarrow \quad & b \leq \text{ approximately } 34.05 \text{ or } b \geq \text{ approximately } 80.95 \end{aligned}$$

by the quadratic equation solution formula. The latter violates $\rho + b < 110$.

Put together, the open region $2.5 \leq b \leq 34.05$, $a \geq \rho$ is where this type of contracts are IR for both agents—given that the agents decommit optimally (not necessarily truthfully). This region is colored gray in Figure 3.4 right.

Case 3, Contractee will surely not decommit. If b is so high that $\breve{b}^*(\rho, a, b) \geq 110$, the contractee will surely not decommit. Now the contractor will decommit whenever $-\breve{a} - a > -\rho \Leftrightarrow \breve{a} < \rho - a$. In other words the decommitting threshold $\breve{a}^* = \rho - a$. The contractor's IR constraint becomes

$$\int_{-\infty}^{\breve{b}^*(\rho,a,b)} g(\breve{b})[\int_{-\infty}^{\rho-a} f(\breve{a})[-\breve{a} - a]d\breve{a} + \int_{\rho-a}^{\infty} f(\breve{a})[-\rho]d\breve{a}]d\breve{b} \geq E[-\breve{a}] \quad (3.12)$$

$$\Leftrightarrow \quad \int_{0}^{110} \frac{1}{110}[\int_{0}^{\rho-a} f(\breve{a})[-\breve{a} - a]d\breve{a} + \int_{\rho-a}^{100} f(\breve{a})[-\rho]d\breve{a}]d\breve{b} \geq -50$$

$$\Leftrightarrow \quad \int_{0}^{\rho-a} f(\breve{a})[-\breve{a} - a]d\breve{a} + \int_{\rho-a}^{100} f(\breve{a})[-\rho]d\breve{a} \geq -50$$

If $a \geq \rho$, this is equivalent to $-\rho \geq -50$ which is false. If $0 \leq a < \rho$, this is equivalent to

$$\frac{1}{100}[\int_{0}^{\rho-a} [-\breve{a} - a]d\breve{a} + \int_{\rho-a}^{100} [-\rho]d\breve{a}] \geq -50$$

$$\Leftrightarrow \quad \frac{1}{100}[(\frac{-(\rho-a)^2}{2} + (\rho - a)(-a)) + ((100 - (\rho - a)) \cdot (-\rho)) \geq -50$$

$$\Leftrightarrow \quad a \leq \text{ approximately } 30.14 \text{ or } a \geq \text{ approximately } 74.86$$

by the quadratic equation solution formula. The latter violates $a < \rho$.

Similarly, the contractee's IR constraint becomes

$$\int_{-\infty}^{\breve{b}^*(\rho,a,b)} g(\breve{b})[\int_{-\infty}^{\rho-a} f(\breve{a})[\breve{b} + a]d\breve{a} + \int_{\rho-a}^{\infty} f(\breve{a})[\rho]d\breve{a}]d\breve{b} \geq E[\breve{b}] \quad (3.13)$$

$$\Leftrightarrow \quad \int_{0}^{110} \frac{1}{110}[[\breve{b} + a] \int_{0}^{\rho-a} f(\breve{a})d\breve{a} + [\rho] \int_{\rho-a}^{100} f(\breve{a})d\breve{a}]d\breve{b} \geq 55$$

$$\Leftrightarrow \quad \frac{1}{110}[[\frac{110^2}{2} + 110a] \int_{0}^{\rho-a} f(\breve{a})d\breve{a} + 110\rho \int_{\rho-a}^{100} f(\breve{a})d\breve{a}] \geq 55$$

$$\Leftrightarrow \quad [55 + a] \int_{0}^{\rho-a} f(\breve{a})d\breve{a} + \rho \int_{\rho-a}^{100} f(\breve{a})d\breve{a} \geq 55$$

If $a \geq \rho$, this is equivalent to $\rho \geq 55$ which is false. If $0 \leq a < \rho$, this is equivalent to

$$[55 + a](\rho - a)\frac{1}{100} + \rho[100 - (\rho - a)]\frac{1}{100} \geq 55$$

$$\Leftrightarrow \quad 2.5 \leq a \leq 47.5$$

by the quadratic equation solution formula. Thus the open region $2.5 \leq a \leq 30.14$, $\breve{b}^* \geq 110$ is where this type of contracts are IR for both agents—given that the agents decommit optimally (not necessarily truthfully). This region is colored gray in Figure 3.4 middle.

In addition to enabling deals that are impossible using full commitment contracts, leveled commitment contracts can increase the efficiency of a deal even if a full commitment contract were possible (the reverse cannot occur because leveled commitment contracts subsume full commitment ones). This occurs always if there is some chance that the contractor's outside offer is lower than the contractee's expected outside offer, or some chance that the contractee's outside offer is higher than the contractor's expected outside offer.

Theorem 3.9 (Pareto efficiency improvement in SEQD games) *If a SEQD game has at least one IR full commitment contract F and*

1. *$\breve{b}$ is bounded from above, f is bounded, and $\int_{-\infty}^{E[\breve{b}]} f(\breve{a})d\breve{a} > 0$, or*

2. *$\breve{a}$ is bounded from below, g is bounded, and $\int_{E[\breve{a}]}^{\infty} g(\breve{b})d\breve{b} > 0$,*

then that game has a leveled commitment contract that increases the contractor's expected payoff as well as the contractee's expected payoff over any full commitment contract. Therefore, the leveled commitment contract is Pareto superior and IR.

It follows that under the conditions of the theorem, no full commitment contract is Pareto efficient.

Unlike in the DOP and COBV games of Sections 3.2.1.1 and 3.2.1.2, in SEQD games, one agent's expected payoff for a given contract may depend on the other agent's—possibly biased—beliefs. For example, the contractee's decision of whether to decommit depends on its belief $f_b(\breve{a})$ of the contractor's upcoming outside offer:

$$\breve{b}^*(\rho, a, b) = \rho + \frac{b + \int_{-\infty}^{\rho-a} f_b(\breve{a}) d\breve{a}[a]}{\int_{\rho-a}^{\infty} f_b(\breve{a}) d\breve{a}}$$

That decommitting decision affects the contractor's expected payoff, which really is (the contractor could perceive it differently):

$$\begin{aligned} \pi_a &= \int_{\breve{b}^*(\rho,a,b)}^{\infty} g(\breve{b}) \int_{-\infty}^{\infty} f(\breve{a})[-\breve{a} + b] d\breve{a} d\breve{b} \\ &+ \int_{-\infty}^{\breve{b}^*(\rho,a,b)} g(\breve{b}) [\int_{-\infty}^{\rho-a} f(\breve{a})[-\breve{a} - a] d\breve{a} + \int_{\rho-a}^{\infty} f(\breve{a})[-\rho] d\breve{a}] d\breve{b} \end{aligned}$$

3.2.2.2 Simultaneous Decommitting

In our simultaneous decommitting games, both agents have to declare decommitment simultaneously. Again, at decommitting time, the contractor knows its outside offer $\breve{a}$ but not the contractee's outside offer $\breve{b}$. Similarly, the contractee knows its outside offer $\breve{b}$ but not the contractor's outside offer $\breve{a}$. There are two alternative types of leveled commitment contracts that differ on what happens if both agents decommit. In the first, both agents have to pay the decommitment penalties to each other if both decommit. In the second, neither agent has to pay if both decommit. Figure 3.5 presents the games induced by both of these contract types. Next, these two game types are discussed separately.

3.2.2.2.1 Both Pay if Both Decommit (SIMUDBP)

This section discusses simultaneous decommitting games where a protocol is used where both agents have to pay the decommitting penalties if both decommit. Such settings will be called SIMUDBP games, Figure 3.5. Let p_b be the probability that

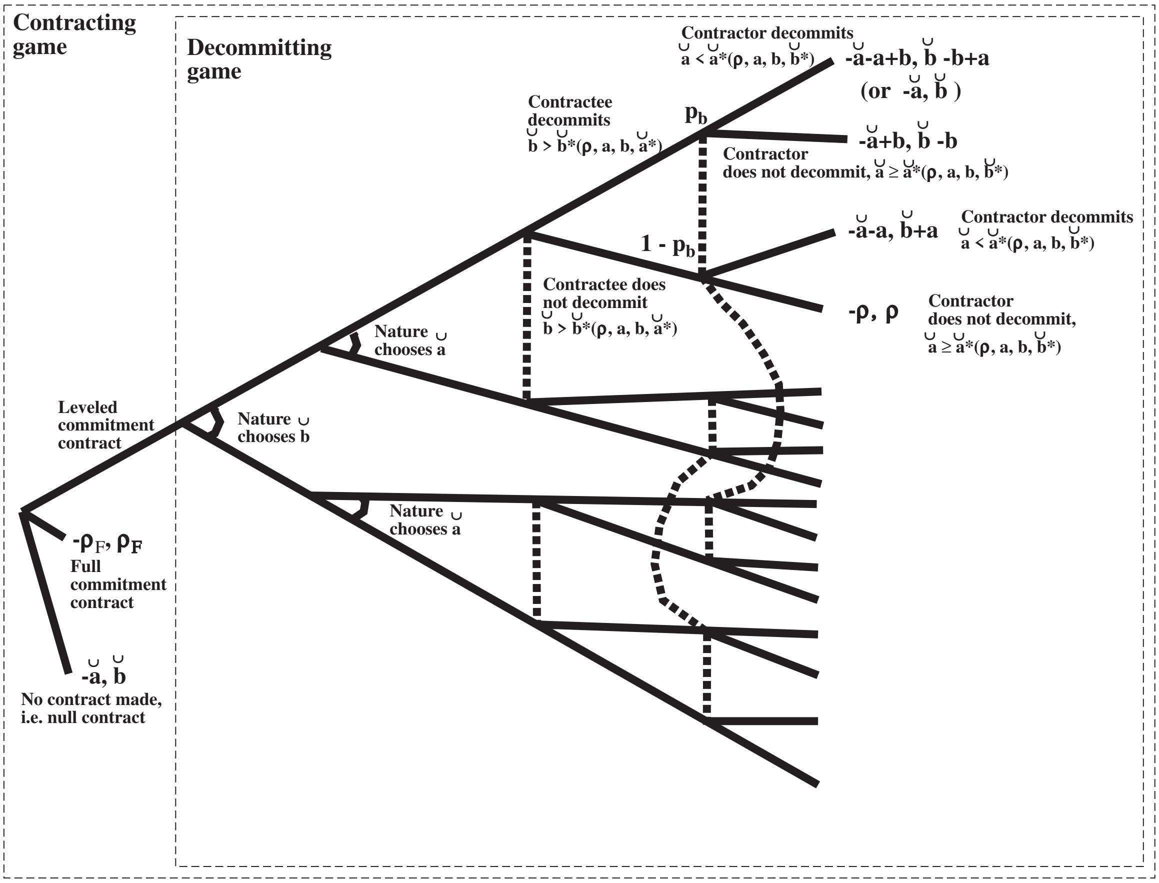

Figure 3.5. The "SIMUltaneous Decommit - Both Pay if both decommit" (SIMUDBP) game. The parenthesized payoffs represent the "SIMUltaneous Decommit - Neither Pays if both decommit" (SIMUDNP) game. The dashed lines represent the agents' information sets. When decommitting, the contractor does not know the contractee's outside offer and vice versa. Furthermore, the contractor has to decide on decommitting before it has observed the contractee's decommitting decision, and vice versa.

the contractee decommits. The value of this variable depends on $f(\breve{a})$, $g(\breve{b})$, ρ, a, and b. The contractor will decommit if

$$p_b \cdot (-\breve{a} + b - a) + (1 - p_b)(-\breve{a} - a) > p_b \cdot (-\breve{a} + b) + (1 - p_b)(-\rho)$$

If $p_b = 1$, this is equivalent to $a < 0$. But we already ruled out this type of contracts where either one of the agents gets paid for decommitting. On the other hand, the above inequality is equivalent to

$$\breve{a} < \rho - \frac{a}{1 - p_b} \stackrel{\text{def}}{=} \breve{a}^*(\rho, a, b, \breve{b}^*) \text{ when } p_b < 1 \tag{3.14}$$

Thus we have characterized a decommitting threshold $\breve{a}^*$ for the contractor. If the contractor's outside offer $\breve{a}$ is less than $\breve{a}^*$, the contractor is best off by decommitting. The contractee decommits if

$$\int_{\breve{a}^*(\rho,a,b,\breve{b}^*)}^{\infty} f(\breve{a})d\breve{a}[\breve{b}-b] + \int_{-\infty}^{\breve{a}^*(\rho,a,b,\breve{b}^*)} f(\breve{a})d\breve{a}[\breve{b}-b+a]$$
$$> \int_{\breve{a}^*(\rho,a,b,\breve{b}^*)}^{\infty} f(\breve{a})d\breve{a}[\rho] + \int_{-\infty}^{\breve{a}^*(\rho,a,b,\breve{b}^*)} f(\breve{a})d\breve{a}[\breve{b}+a]$$

If $\int_{\breve{a}^*(\rho,a,b,\breve{b}^*)}^{\infty} f(\breve{a})d\breve{a} = 0$, this is equivalent to $b < 0$. But we already ruled out this type of contracts where either one of the agents gets paid for decommitting. On the other hand, the above inequality is equivalent to

$$\breve{b} > \rho + \frac{b}{\int_{\breve{a}^*}^{\infty} f(\breve{a})d\breve{a}} \stackrel{\text{def}}{=} \breve{b}^*(\rho, a, b, \breve{a}^*) \text{ when } \int_{\breve{a}^*(\rho,a,b,\breve{b}^*)}^{\infty} f(\breve{a})d\breve{a} > 0 \qquad (3.15)$$

Now we have characterized a decommitting threshold $\breve{b}^*$ for the contractee. If the contractee's outside offer $\breve{b} > \breve{b}^*$, the contractee is best off by decommitting. The probability that the contractee will decommit is

$$p_b = \int_{\breve{b}^*(\rho,a,b,\breve{a}^*)}^{\infty} g(\breve{b})d\breve{b} \qquad (3.16)$$

Condition 3.14 states the contractor's best response (defined by $\breve{a}^*$) to the contractee's strategy that is defined by $\breve{b}^*$. Condition 3.15 states the contractee's best response $\breve{b}^*$ to the contractor's strategy that is defined by $\breve{a}^*$. Condition 3.14 uses the variable p_b which is defined by Equation 3.16. So together, Equations 3.14, 3.15, and 3.16 define the Nash equilibria of the decommitting game.

Now the contractor's IR constraint becomes

$$\int_{\breve{b}^*(\rho,a,b,\breve{a}^*)}^{\infty} g(\breve{b})[\int_{-\infty}^{\breve{a}^*(\rho,a,b,\breve{b}^*)} f(\breve{a})[-\breve{a}+b-a]d\breve{a} + \int_{\breve{a}^*(\rho,a,b,\breve{b}^*)}^{\infty} f(\breve{a})[-\breve{a}+b]d\breve{a}]d\breve{b}$$
$$+ \int_{-\infty}^{\breve{b}^*(\rho,a,b,\breve{a}^*)} g(\breve{b})[\int_{-\infty}^{\breve{a}^*(\rho,a,b,\breve{b}^*)} f(\breve{a})[-\breve{a}-a]d\breve{a} + \int_{\breve{a}^*(\rho,a,b,\breve{b}^*)}^{\infty} f(\breve{a})[-\rho]d\breve{a}]d\breve{b} \geq E[-\breve{a}]$$

The first row corresponds to the contractee decommitting, while the second corresponds to the contractee not decommitting. The second integral in each row corresponds to the contractor decommitting, while the third integral corresponds to the contractor not decommitting. Using the same logic, the contractee's IR constraint becomes

$$\int_{\breve{b}^*(\rho,a,b,\breve{a}^*)}^{\infty} g(\breve{b})[\int_{-\infty}^{\breve{a}^*(\rho,a,b,\breve{b}^*)} f(\breve{a})[\breve{b}-b+a]d\breve{a} + \int_{\breve{a}^*(\rho,a,b,\breve{b}^*)}^{\infty} f(\breve{a})[\breve{b}-b]d\breve{a}]d\breve{b}$$
$$+\int_{-\infty}^{\breve{b}^*(\rho,a,b,\breve{a}^*)} g(\breve{b})[\int_{-\infty}^{\breve{a}^*(\rho,a,b,\breve{b}^*)} f(\breve{a})[\breve{b}+a]d\breve{a} + \int_{\breve{a}^*(\rho,a,b,\breve{b}^*)}^{\infty} f(\breve{a})[\rho]d\breve{a}]d\breve{b} \geq E[\breve{b}]$$

If $\breve{a}$ is bounded from below, the contractor's decommitment penalty a can be chosen so high that the contractor's decommitment threshold $\breve{a}^*(\rho, a, b, \breve{b}^*)$ becomes lower than any $\breve{a}$. In that case the contractor will surely not decommit. Similarly, if $\breve{b}$ is bounded from above, the contractee's decommitment penalty b can be chosen so high that the contractee's decommitment threshold $\breve{b}^*(\rho, a, b, \breve{a}^*)$ is greater than any $\breve{b}$. In that case the contractee will surely not decommit. Thus, full commitment contracts are a subset of leveled commitment ones. Therefore, the former can be no better in the sense of Pareto efficiency or social welfare than the latter.

In addition to leveled commitment contracts never being worse than full commitment ones, the following theorem states the positive result that in SIMUDBP games, they can enable—through increased efficiency—a deal that is impossible via full commitment contracts.

Theorem 3.10 (Enabling in SIMUDBP games) *There are SIMUDBP games (defined by $f(\breve{a})$ and $g(\breve{b})$) where no full commitment contract satisfies the IR constraints but a leveled commitment contract does.*

Proof. Let $f(\breve{a}) = \begin{cases} \frac{1}{100} & \text{if } 0 \leq \breve{a} \leq 100 \\ 0 & \text{otherwise} \end{cases}$ and $g(\breve{b}) = \begin{cases} \frac{1}{110} & \text{if } 0 \leq \breve{b} \leq 110 \\ 0 & \text{otherwise} \end{cases}$. No full commitment contract F satisfies both IR constraints because that would require $E[\breve{b}] \leq \rho_F \leq E[\breve{a}]$ which is impossible because $55 = E[\breve{b}] > E[\breve{a}] = 50$. Let us analyze a leveled commitment contract where $\rho = 52.5$. There are four qualitatively different cases.

Case 1. Some chance that either agent is going to decommit. If $0 < \breve{a}^* < 100$, and $0 < \breve{b}^* < 110$, there is a nonzero probability for each agent to decommit. The unique Nash equilibrium is plotted out for different values of a and b in Figure 3.6. The Nash equilibrium decommitment thresholds $\breve{a}^*$ and $\breve{b}^*$ differ from the truthful

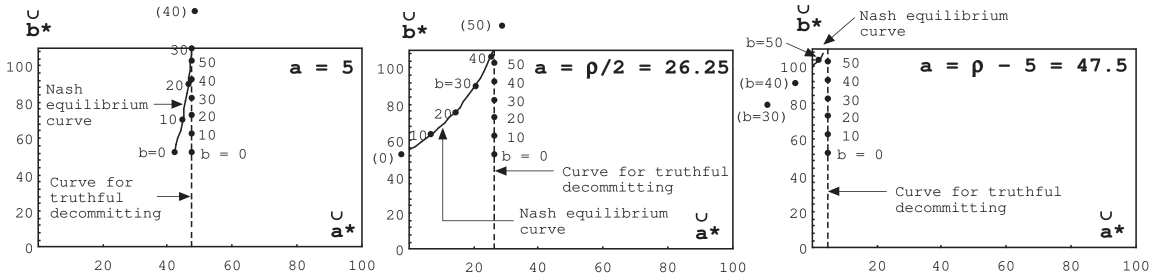

Figure 3.6. The Nash equilibrium decommitment thresholds $\breve{a}^*$ and $\breve{b}^*$ of our example SIMUDBP game for different values of the decommitment penalties a and b. The Nash equilibrium deviates from truthful decommitting. If $0 < \breve{a}^* < 100$, and $0 < \breve{b}^* < 110$, there is some chance that either agent will decommit.

ones. Yet there exist Nash equilibria within the proper range of $\breve{a}^*$ and $\breve{b}^*$. It is not guaranteed that all of these Nash equilibria satisfy the agents' IR constraints however.

We programmed a model of Equations 3.14, 3.15, and 3.16 and the IR constraints. To make the algebra tractable (constant $f(\breve{a})$ and $g(\breve{b})$), versions of these equations were used that assumed $0 < \breve{a}^* < 100$, and $0 < \breve{b}^* < 110$, without loss of generality. Therefore the first task was to check the boundaries of the validity of the model. The boundaries $\breve{a}^* = 0$ and $\breve{b}^* = 110$ are plotted in Figure 3.7. The boundary $\breve{a}^* = 100$ turns out to be the line $b = 0$. There exists no boundary $\breve{b}^* = 0$ because $\breve{b}^*$ was always greater than zero.

After plotting the validity boundaries of the model, the curves at which the IR constraints held with equality were plotted, Fig. 3.7. Each agent's IR constraint induced three curves, two of which actually bound the IR region. The third one is just a root of the IR constraint, but at both sides of that curve, the IR constraint

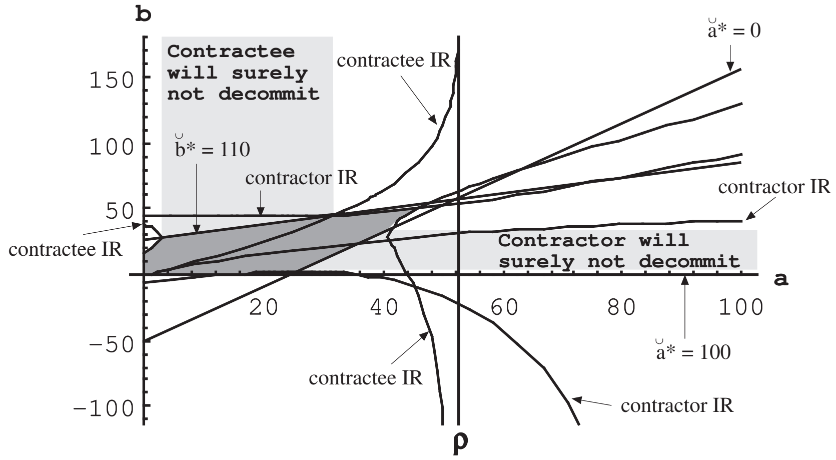

Figure 3.7. IR regions in the SIMUDBP decommitting game. The gray areas are three qualitatively different regions of contracts that are IR for both agents and allow an equilibrium in the SIMUDBP decommitting game. In the dark gray area either agent might decommit while in the light gray areas only one agent might decommit. The curves represent the IR constraints and validity constraints of the programmed model that requires $0 < \breve{a}^* < 100$, and $0 < \breve{b}^* < 110$. Both agents have one curve from their IR constraint that is just a root of the constraint but is satisfied on both sides.

is satisfied. Now, the dark gray area of Figure 3.7 represents the values of the decommitment penalties a and b for which the validity constraints of the programmed model and the IR constraints are satisfied. In other words, for any such a and b, there exists decommitment thresholds $\breve{a}^*$ and $\breve{b}^*$ such that these form a Nash equilibrium, and there is a nonzero probability for either agent to decommit, and each agent has higher expected payoff with the contract than without it.

As a numeric example, pick a contract where $a = \frac{\rho}{2} = 26.25$, and $b = 30$. Now in Nash equilibrium, the decommitment thresholds are $\breve{a}^* \approx 20.50$, and $\breve{b}^* \approx 90.24$, Figure 3.6. The contractor's expected payoff is approximately $-44.94 > E[-\breve{a}] = -50$, and the contractee's is approximately $59.81 > E[\breve{b}] = 55$. Thus both agents' expected payoffs are higher than without the contract, i.e. the contract is IR for both agents. This suffices to prove the theorem. Nevertheless, we present the other types of equilibria that can occur.

Case 2, Contractor will surely not decommit. If $\breve{a}^* \leq 0$, the contractor will surely not decommit. Now $\breve{b}^*(\rho, a, b, \breve{a}^*) = \rho + \frac{b}{\int_{\breve{a}^*}^{\infty} f(\breve{a})d\breve{a}} = \rho + b$, i.e. the contractee decommits truthfully. The contractor's IR constraint becomes exactly the same as in case 2 of the example SEQD game (Eq. 3.10). This constraint was proven equivalent to $2.5 \leq b \leq 52.5$. Similarly, the contractee's IR constraint becomes exactly the same as in the SEQD game (Eq. 3.11). It was proven equivalent to $b \leq$ approximately 34.05. Thus the open region $2.5 \leq b \leq 34.05$, $\breve{a}^* \leq 0$ is where this type of contracts are IR for both agents and in equilibrium. This region is colored light gray in Figure 3.7.

Case 3, Contractee will surely not decommit. If $\breve{b}^* \geq 110$, the contractee will surely not decommit ($p_b = 0$). Now $\breve{a}^*(\rho, a, b, \breve{b}^*) = \rho - \frac{a}{1-p_b} = \rho - a$, i.e. the contractor decommits truthfully. The contractor's IR constraint becomes exactly the same as in case 3 of the example SEQD game (Eq. 3.13). This constraint was proven equivalent to $a \leq$ approximately 30.14. Similarly, the contractee's IR constraint becomes the same as in the SEQD game (Eq. 3.14). It was proven equivalent to $2.5 \leq a \leq 47.5$. Thus the open region $2.5 \leq a \leq 30.14$, $\breve{b}^* \geq 110$ is where this type of contracts are IR for both agents, and in equilibrium. This region is colored light gray in Figure 3.7.

Case 4, Trivial case. A contract where at least one agent will surely decommit, i.e. $\breve{a}^* \geq 100$ or $\breve{b}^* \leq 0$ can be IR. For such a contract to be IR for the decommitting agent, its decommitment penalty would have to be zero. Thus the decommitting agent gets the same payoff as without the contract. Similarly, the other agent gets the same payoff as it would get without the contract. Though this contract is IR for both agents (barely because it does not increase either agent's payoff), it is equivalent to no contract at all: decommitment occurs and no payment is transferred. □

In addition to enabling deals that are impossible using full commitment contracts, leveled commitment contracts can increase the efficiency of a deal even if a full commitment contract were possible (the reverse cannot occur because leveled commitment contracts can emulate full commitment ones). This occurs always if there is some chance that the contractor's outside offer is lower than the contractee's expected outside offer, or some chance that the contractee's outside offer is higher than the contractor's expected outside offer.

Theorem 3.11 (Pareto efficiency improvement in SIMUDBP games) *If a SIMUDBP game has at least one IR full commitment contract F and*

1. *$\breve{b}$ is bounded from above, f is bounded, and $\int_{-\infty}^{E[\breve{b}]} f(\breve{a})d\breve{a} > 0$, or*

2. *$\breve{a}$ is bounded from below, g is bounded, and $\int_{E[\breve{a}]}^{\infty} g(\breve{b})d\breve{b} > 0$,*

then that game has a leveled commitment contract that increases the contractor's expected payoff as well as the contractee's expected payoff over any full commitment contract. Therefore, the leveled commitment contract is Pareto superior and IR.

It follows that under the conditions of the theorem, no full commitment contract is Pareto efficient.

In SIMUDBP games—like SEQD games but unlike DOP and COBV games—one agent's expected payoff for a given contract may depend on the other agent's—possibly biased—beliefs. For example, the contractee's decision of whether to decommit depends on its belief $f_b(\breve{a})$ of the contractor's upcoming outside offer. For example, if the contractee receives a good outside offer, it would decommit if it acted truthfully. But if the contractee believes—according to $f_b(\breve{a})$—that the contractor is likely to get a good outside offer and decommit, then the contractee can save the decommitment penalty by not decommitting. On the other hand, the contractee's decommitting decision affects the contractor's expected payoff because in case the contractee decommits, the contractor's payoff is either $-\breve{a} + b$ or $-\breve{a} + b - a$, and in case the contractee does not decommit, the contractor's payoff is either $-\rho$ or $-\breve{a} - a$. Because of such dependencies, an agent's preference order over potential contracts may depend on the other agent's beliefs. Therefore, in SIMUDBP games with asymmetric biased information, an agent may need to counterspeculate the other agent's beliefs in order to determine a preference order over contracts.

3.2.2.2.2 Neither Pays if Both Decommit (SIMUDNP)

This section discusses simultaneous decommitting games where a protocol is used where neither agent has to pay a decommitting penalty if both agents decommit. In such a SIMUDNP game (Figure 3.5), the contractor will decommit if

$$p_b \cdot (-\breve{a}) + (1-p_b)(-\breve{a}-a) > p_b \cdot (-\breve{a}+b) + (1-p_b)(-\rho)$$

If $p_b = 1$, this is equivalent to $0 > b$. But we already ruled out this type of contracts where either one of the agents gets paid for decommitting. On the other hand, the above inequality is equivalent to

$$\breve{a} < \rho - a - \frac{bp_b}{1-p_b} \stackrel{\text{def}}{=} \breve{a}^*(\rho, a, b, \breve{b}^*) \text{ when } p_b < 1 \tag{3.17}$$

The contractee decommits if

$$\begin{aligned}&\int_{\breve{a}^*(\rho,a,b,\breve{b}^*)}^{\infty} f(\breve{a})d\breve{a}[\breve{b}-b] + \int_{-\infty}^{\breve{a}^*(\rho,a,b,\breve{b}^*)} f(\breve{a})d\breve{a}[\breve{b}] \\ &> \int_{\breve{a}^*(\rho,a,b,\breve{b}^*)}^{\infty} f(\breve{a})d\breve{a}[\rho] + \int_{-\infty}^{\breve{a}^*(\rho,a,b,\breve{b}^*)} f(\breve{a})d\breve{a}[\breve{b}+a]\end{aligned}$$

If $\int_{\breve{a}^*(\rho,a,b,\breve{b}^*)}^{\infty} f(\breve{a})d\breve{a} = 0$, this is equivalent to $0 > a$. But we already ruled out this type of contracts where either one of the agents gets paid for decommitting. On the other hand, the above inequality is equivalent to

$$\breve{b} > \rho + b - \frac{a\int_{-\infty}^{\breve{a}^*(\rho,a,b,\breve{b}^*)} f(\breve{a})d\breve{a}}{\int_{\breve{a}^*(\rho,a,b,\breve{b}^*)}^{\infty} f(\breve{a})d\breve{a}} \stackrel{\text{def}}{=} \breve{b}^*(\rho, a, b, \breve{a}^*) \text{ when } \int_{\breve{a}^*(\rho,a,b,\breve{b}^*)}^{\infty} f(\breve{a})d\breve{a} > 0 \tag{3.18}$$

The probability that the contractee will decommit is

$$p_b = \int_{\breve{b}^*(\rho,a,b,\breve{a}^*)}^{\infty} g(\breve{b})d\breve{b} \tag{3.19}$$

Condition 3.17 states the contractor's best response (defined by $\breve{a}^*$) to the contractee's strategy that is defined by $\breve{b}^*$. Condition 3.18 states the contractee's best response $\breve{b}^*$

to the contractor's strategy that is defined by $\breve{a}^*$. Condition 3.17 uses the variable p_b which is defined by Equation 3.19. So together, Equations 3.17, 3.18, and 3.19 define the Nash equilibria of the decommitting game.

Now the contractor's IR constraint becomes

$$\int_{\breve{b}^*(\rho,a,b,\breve{a}^*)}^{\infty} g(\breve{b})[\int_{-\infty}^{\breve{a}^*(\rho,a,b,\breve{b}^*)} f(\breve{a})[-\breve{a}]d\breve{a} + \int_{\breve{a}^*(\rho,a,b,\breve{b}^*)}^{\infty} f(\breve{a})[-\breve{a}+b]d\breve{a}]d\breve{b}$$
$$+\int_{-\infty}^{\breve{b}^*(\rho,a,b,\breve{a}^*)} g(\breve{b})[\int_{-\infty}^{\breve{a}^*(\rho,a,b,\breve{b}^*)} f(\breve{a})[-\breve{a}-a]d\breve{a} + \int_{\breve{a}^*(\rho,a,b,\breve{b}^*)}^{\infty} f(\breve{a})[-\rho]d\breve{a}]d\breve{b} \geq E[-\breve{a}]$$

The first row corresponds to the contractee decommitting, while the second corresponds to the contractee not decommitting. The second integral in each row corresponds to the contractor decommitting, while the third integral corresponds to the contractor not decommitting. Using the same logic, the contractee's IR constraint becomes

$$\int_{\breve{b}^*(\rho,a,b,\breve{a}^*)}^{\infty} g(\breve{b})[\int_{-\infty}^{\breve{a}^*(\rho,a,b,\breve{b}^*)} f(\breve{a})[\breve{b}]d\breve{a} + \int_{\breve{a}^*(\rho,a,b,\breve{b}^*)}^{\infty} f(\breve{a})[\breve{b}-b]d\breve{a}]d\breve{b}$$
$$+\int_{-\infty}^{\breve{b}^*(\rho,a,b,\breve{a}^*)} g(\breve{b})[\int_{-\infty}^{\breve{a}^*(\rho,a,b,\breve{b}^*)} f(\breve{a})[\breve{b}+a]d\breve{a} + \int_{\breve{a}^*(\rho,a,b,\breve{b}^*)}^{\infty} f(\breve{a})[\rho]d\breve{a}]d\breve{b} \geq E[\breve{b}]$$

Now, can some full commitment contract be more efficient than any leveled commitment contract in a SIMUDNP game? If $\breve{a}$ is bounded from below, and $\breve{b}$ from above, a can be chosen so high that the contractor will surely not decommit, and b so high that the contractee will not. Thus, full commitment contracts are a subset of leveled commitment ones. Therefore, the former cannot enable a deal whenever the latter cannot.

In addition, leveled commitment contracts can enable—through increased efficiency—a deal that is impossible via full commitment contracts:

Theorem 3.12 (Enabling in SIMUDNP games) *There exist SIMUDNP games (defined by $f(\breve{a})$ and $g(\breve{b})$) where no full commitment contract satisfies the IR constraints but a leveled commitment contract does.*

Proof. Let $f(\breve{a}) = \begin{cases} \frac{1}{100} & \text{if } 0 \leq \breve{a} \leq 100 \\ 0 & \text{otherwise} \end{cases}$ and $g(\breve{b}) = \begin{cases} \frac{1}{110} & \text{if } 0 \leq \breve{b} \leq 110 \\ 0 & \text{otherwise} \end{cases}$. No full commitment contract F satisfies both IR constraints because that would require $E[\breve{b}] \leq \rho_F \leq E[\breve{a}]$ which is impossible because $55 = E[\breve{b}] > E[\breve{a}] = 50$. Let us analyze a leveled commitment contract where $\rho = 52.5$. There are four qualitatively different cases.

Case 1. Some chance that either agent is going to decommit. If $0 < \breve{a}^* < 100$, and $0 < \breve{b}^* < 110$, there is a nonzero probability for each agent to decommit. The unique Nash equilibrium is plotted out for different values of a and b in Figure 3.8. Note that the Nash equilibrium decommitment thresholds $\breve{a}^*$ and $\breve{b}^*$ really do differ

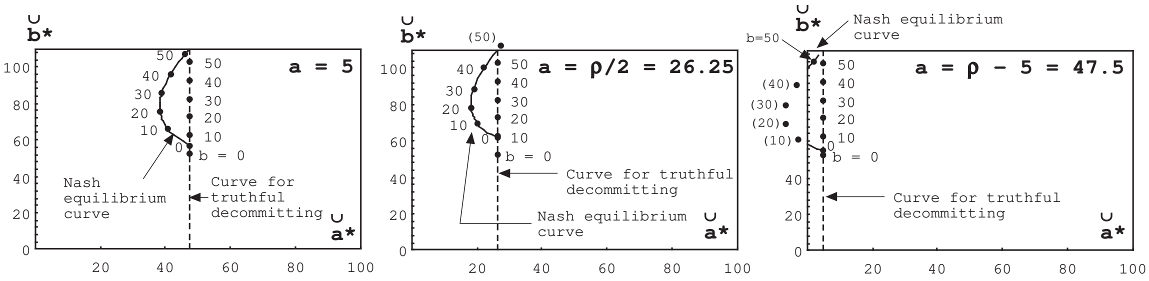

Figure 3.8. The Nash equilibrium decommitment thresholds $\breve{a}^*$ and $\breve{b}^*$ of our example SIMUDNP game for different values of the decommitment penalties a and b. The Nash equilibrium deviates from truthful decommitting. If $0 < \breve{a}^* < 100$, and $0 < \breve{b}^* < 110$, there is some chance that either agent will decommit.

from the truthful ones. They also differ from—are closer to the truthful ones than—what they were when a protocol where both agents pay if both decommit was used, Figure 3.6. The shapes of the curves using these two protocols also differ significantly. Yet there exist Nash equilibria within the proper range of $\breve{a}^*$ and $\breve{b}^*$. These Nash equilibria do not necessarily satisfy the agents' IR constraints however.

We programmed a model of Equations 3.17, 3.18, and 3.19 and the IR constraints. To make the algebra tractable (constant $f(\breve{a})$ and $g(\breve{b})$), versions of these equations were used that assumed $0 < \breve{a}^* < 100$ and $0 < \breve{b}^* < 110$, without loss of generality.

Therefore the first task was to check the validity boundaries of the model. The boundaries $\breve{a}^* = 0$, $\breve{a}^* = 100$, $\breve{b}^* = 0$, and $\breve{b}^* = 110$ are plotted with bold lines in Figure 3.9.

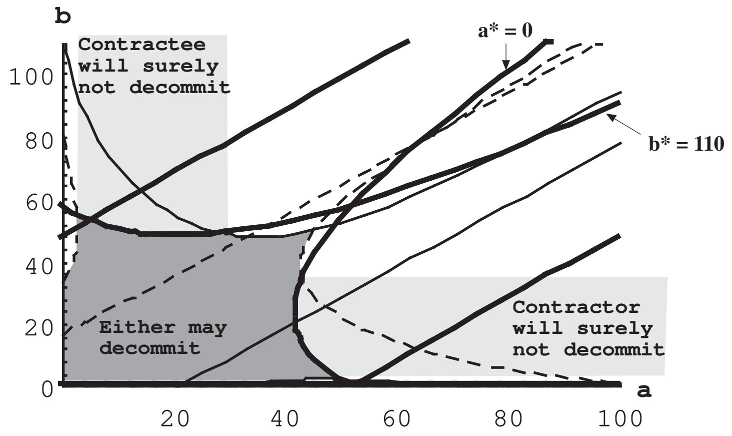

Figure 3.9. Three qualitatively different regions of contracts that are IR for both agents and allow an equilibrium in the SIMUDNP decommitting game. The bold lines are the validity constraints for the programmed model that requires $0 < \breve{a}^* < 100$, and $0 < \breve{b}^* < 110$. One of these validity constraints slices the "either may decommit" region, but the constraint is satisfied on both sides of the line. The solid lines represent the contractor's IR constraint from the programmed model, and the dashed lines represent the contractee's IR constraint. Both agents have one curve from their constraint that is just a root of the constraint but is satisfied on both sides.

Next, the curves at which the IR constraints held with equality were plotted, Figure 3.9. Note that each agent's IR constraint induced three curves, two of which actually bound the IR region. The third one is just a root of the IR constraint, but at both sides of that curve, the IR constraint is satisfied. Now, the dark gray area of Figure 3.9 represents the values of the decommitment penalties a and b for which the validity constraints of the programmed model and the IR constraints are satisfied. In other words, for any such a and b, there exist decommitment thresholds $\breve{a}^*$ and $\breve{b}^*$ such that these form a Nash equilibrium, and there is a nonzero probability for either agent to decommit, and each agent has higher expected payoff with the contract than without it.

As a numeric example, pick a contract where $a = \frac{\rho}{2} = 26.25$, and $b = 30$. Now in Nash equilibrium, the decommitment thresholds are $\breve{a}^* \approx 19.03$, and $\breve{b}^* \approx 88.67$, Figure 3.8. The contractor's expected payoff is approximately $-44.74 > E[-\breve{a}] = -50$, and the contractee's is approximately $59.65 > E[\breve{b}] = 55$. Thus both agents' expected payoffs are higher than without the contract, i.e. the contract is IR for both agents. This suffices to prove the theorem. Nevertheless, we present the other types of equilibria that can occur.

Case 2, Contractor will surely not decommit. If $\breve{a}^* \leq 0$, the contractor will surely not decommit. Now $\breve{b}^*(\rho, a, b, \breve{a}^*) = \rho + b - \frac{a \int_{-\infty}^{\breve{a}^*(\rho,a,b,\breve{b}^*)} f(\breve{a}) d\breve{a}}{\int_{\breve{a}^*(\rho,a,b,\breve{b}^*)}^{\infty} f(\breve{a}) d\breve{a}} = \rho + b$, i.e. the contractee decommits truthfully. Now the contractor's and the contractee's IR constraints become exactly the same as in case 2 of the SEQD example (Eq. 3.10 and 3.11)—and same as in case 2 of the SIMUDBP example. It follows that the open region $2.5 \leq b \leq 34.05$, $\breve{a}^* \leq 0$ is where this type of contracts are IR for both agents and in equilibrium. This region is colored light gray in Figure 3.9.

Case 3, Contractee will surely not decommit. If $\breve{b}^* \geq 110$, the contractee will surely not decommit ($p_b = 0$). Now $\breve{a}^*(\rho, a, b, \breve{b}^*) = \rho - a - \frac{b p_b}{1 - p_b} = \rho - a$, i.e. the contractor decommits truthfully. The contractor's and the contractee's IR constraints become exactly the same as in case 3 of the SEQD example (Eq. 3.13 and 3.14)—and same as in case 3 of the SIMUDBP example. It follows that the open region $2.5 \leq a \leq 30.14$, $\breve{b}^* \geq 110$ is where this type of contracts are IR for both agents and in equilibrium. This region is colored light gray in Figure 3.9.

Case 4, Trivial case. Same as case 4 of the SIMUDBP example. □

In addition to enabling deals that are impossible in SIMUDNP games via full commitment contracts, leveled commitment contracts can increase the efficiency of a deal even if a full commitment contract were possible (the reverse cannot occur):

Theorem 3.13 (Pareto efficiency improvement in SIMUDNP games) *If a SIMUDNP game has at least one IR full commitment contract F and*

1. *$\breve{b}$ is bounded from above, f is bounded, and $\int_{-\infty}^{E[\breve{b}]} f(\breve{a})d\breve{a} > 0$, or*

2. *$\breve{a}$ is bounded from below, g is bounded, and $\int_{E[\breve{a}]}^{\infty} g(\breve{b})d\breve{b} > 0$,*

then that game has a leveled commitment contract that increases the contractor's expected payoff as well as the contractee's expected payoff over any full commitment contract. Therefore, the leveled commitment contract is Pareto superior and IR.

It follows that under the conditions of the theorem, no full commitment contract is Pareto efficient.

In SIMUDNP games—like SIMUDBP and SEQD games but unlike DOP and COBV games—one agent's expected payoff for a given contract may depend on the other agent's—possibly biased—beliefs. For example, the contractee's decision of whether to decommit depends on its belief $f_b(\breve{a})$ of the contractor's upcoming outside offer. For example, if the contractee receives a good outside offer, it would decommit if it acted truthfully. But if the contractee believes—according to $f_b(\breve{a})$—that the contractor is likely to get a good outside offer and decommit, then the contractee can save the decommitment penalty by not decommitting. On the other hand, the contractee's decommitting decision affects the contractor's expected payoff because in case the contractee decommits, the contractor's payoff is either $-\breve{a} + b$ or $-\breve{a}$, and in case the contractee does not decommit, the contractor's payoff is either $-\rho$ or $-\breve{a} - a$. Because of such dependencies, an agent's preference order over potential contracts may depend on the other agent's beliefs. Therefore, in SIMUDNP games with biased asymmetric information, an agent may need to counterspeculate the other agent's beliefs in order to determine a preference order over contracts.

3.2.3 Practical Prescriptions for System Builders

The results from the above canonical games suggest that it is worthwhile from a contract enabling and a contract Pareto improving perspective to incorporate the decommitment mechanism into automated contracting protocols. The decommitment penalties are best chosen by the agents dynamically at contract time as opposed to statically in the protocol. This allows the tuning of the penalties not only to specific negotiation situations and environmental uncertainties, but also to specific belief structures of the agents.

The proposed decommitment mechanism allows an agent to decommit based on local reasoning: no negotiation is necessary at decommitment time. The contracts in this mechanism are simpler than traditional contingency contracts that require—in the worst case—the specification of the contract's alternative obligations for all alternative worlds induced by alternative realizations of combinations of future events. Furthermore, the proposed decommitment method does not require an event verification mechanism like contingency contracts do.

In the presented instance of the simultaneous decommitting game, the Nash equilibrium decommitting strategies were usually closer to truthful ones when a protocol was used where neither pays if both decommit (SIMUDNP, Fig. 3.6) than when a protocol was used where both pay if both decommit (SIMUDBP, Fig. 3.8). Also, as an agent's opponent's decommitment penalty approaches zero, the agent becomes truthful in the former protocol, but starts to increasingly bias its decommitment decisions in the latter. This suggests using the former protocol in practical systems. It also minimizes the number of payment transfers because it does not require any such transfer if both decommit.

In a web of multiple mutual contracts among several agents, classical full commitment contracts induce one negotiation focus consisting of the obligations of the contracts. Under the proposed leveled commitment protocol, there are multiple such foci, and any agent involved in a contract can swap from one such focus to another by decommitting from a contract—by paying the decommitment penalty. It may happen that one such swap makes it beneficial for another agent to decommit from another contract, and so on. To avoid loops of decommitting and recommitting in practice, recommitting can be disabled. This can be implemented by choosing a protocol that specifies that if a contract offer is accepted and later either agent decommits, the original offer becomes void—as opposed to staying valid according to its original deadline that may not have been reached at the time of decommitment.

Even though two agents cannot explicitly recommit to a contract, it is hard to specify and monitor in a protocol that they will not make another contract with an identical content. This gives rise to the possibility of the equivalent of useless decommit-recommit loops. Such loops can be avoided by a mechanism where the decommitment penalties increase with real-time or with the number of domain events or negotiation events. This allows a low commitment negotiation focus to be moved in the joint search space while still making the contracts meaningful by some level of commitment. The increasing level of commitment causes the agents to not backtrack deeply in the negotiations, which can also save computation.

The initially low commitment to contracts can also be used as a mechanism to facilitate linking of deals. Often, there is no contract over a single item that is beneficial, but a combination of contracts among two agents would be [Sandholm, 1993, Sandholm and Lesser, 1995c]. Even if explicit clustering of issues into contracts [Sandholm, 1993, Sandholm and Lesser, 1995c] is not used, an agent can agree to an initially unbeneficial low commitment contract in anticipation of synergic future contracts from the other agent that will make the first contract beneficial [Sandholm and Lesser, 1995c]. If no such contracts appear, the agent can decommit. In a similar way the initially low commitment to contracts can be used as a mechanism to facilitate contracts among more than two agents. Even without explicit multiagent contract protocols [Sandholm and Lesser, 1995c], multiagent contracts can be implemented by one agent agreeing to an initially unbeneficial low commitment contract in anticipation of synergic future contracts from third parties that will make the first contract beneficial [Sandholm and Lesser, 1995c]. Again, if no such contracts appear, the agent can decommit.

In many practical automated contracting settings, agents are bounded rational—e.g. because limited computation resources bound their capability to solve combinatorial problems [Sandholm and Lesser, 1995c, Sandholm and Lesser, 1995a, Sandholm and Lesser, 1997, Sandholm, 1993]. The very fact that an agent's computation is bounded induces uncertainty. For example, the value of a contract may only be probabilistically known to the agent at contract time. The leveled commitment contracting protocol allows the agent to continue deliberation regarding the value of the contract after the contract is made. If the value turns out to be lower than expected, the agent can decommit. On the other hand, a leveled commitment contracting protocol where the decommitment penalties increase quickly in time may be appropriate with computationally limited agents so that the agents do not need to consider the combinatorial number of possible future worlds where alternative combinations of decommitments have occurred [Sandholm and Lesser, 1995c]. The marginal cost of a particular task may be different in each one of these alternative worlds because tasks interact.

3.3 Stages of Commitment

In mutual negotiations, commitment to an offer means that one agent binds itself to a potential contract while waiting for the other agent to either accept or reject its offer. If the other party accepts, both parties are bound to the contract. When accepting, the second party is certain that the contract will be made, but the first party has to commit before it is sure. Commitment has to take place at some stage of the protocol for contracts to take place, but the choice of this stage can be varied. The TRACONET system was designed so that commitment took place in the bidding phase as is usual in the real world: if a task is awarded to him, the bidder has to take care of it at the price mentioned in the bid.

But what stage should commitment take place? Conceptually even the announcement could be committal. Then it could be sent to only one potential bidder at a time and it should include a deadline. If the bidder doesn't bid (thus taking the task set) by the deadline, then the announcer can announce the same task set to another agent. Another protocol would make the bids uncommittal, but the awards committal. In theory, arbitrary long protocols can be devised, but commitment has to take place at some point in the protocol.

The choice of commitment stage can be a static protocol design decision or the agents can decide on it dynamically. For example, the focused addressing scheme of the CNP was implemented so that in low utilization situations, contractors announced tasks, but in high utilization mode, potential contractees signaled availability—i.e. bid without receiving announcements first [Smith, 1980, Van Dyke Parunak, 1987]. So, the choice of a protocol was based on characteristics of the environment. Alternatively, the choice can be made for each negotiation separately before that negotiation begins. We advocate a more refined alternative, where agents dynamically choose the stage of commitment of a certain negotiation during that negotiation. This allows any of the above alternatives, but makes the stage of commitment a negotiation strategy decision, not a protocol design decision. The offered commitments are specified in *contractor messages* and *contractee messages*, Fig. 3.10.

3.4 Richer Negotiation Protocol

In addition to leveled commitment contracts and commitment at different stages, we advocate other new features to contracting protocols. This section discusses an example protocol that incorporates such features. This specific protocol is for negotiation among multiple agents, but the contract that each one of such negotiations leads to is only among two agents.

```
CONTRACTOR MESSAGE:
0. Negotiation identifier
1. Message identifier
2. In-response-to (message id)
3. Sender
4. Receiver
5. Terminate negotiation
6. Alternative 1
  6.1. Time valid through
  6.2. Bind after partner's decommit
  6.3. Offer submission fee
  6.4. Required response submission fee
  6.5. Task set 1
    (a) (Minimum) specification of tasks
    (b) Promised payment fn. to contractee
    (c) Contractor's promised commitment fn.
    (d) Contractee's required commitment fn.
  6.6. Task set 2
       ...
  6.i. Task set i-4
7. Alternative 2
     ...
j. Alternative j-5
```

```
CONTRACTEE MESSAGE:
0. Negotiation identifier
1. Message identifier
2. In-response-to (message id)
3. Sender
4. Receiver
5. Terminate negotiation
6. Alternative 1
  6.1. Time valid through
  6.2. Bind after partner's decommit
  6.3. Offer submission fee
  6.4. Required response submission fee
  6.5. Task set 1
    (a) (Maximum) specification of tasks
    (b) Required payment fn. to contractee
    (c) Contractor's required commitment fn.
    (d) Contractee's promised commitment fn.
  6.6. Task set 2
       ...
  6.m. Task set m-4
7. Alternative 2
     ...
n. Alternative n-5
```

```
PAYMENT/DECOMMIT MESSAGE:
0. Negotiation id
1. Message id
2. Accepted offer id
3. Acceptance message id
4. Sender
5. Receiver
6. Message type
      (payment/decommit)
7. Money transfer
```

Figure 3.10. Contracting messages of a single negotiation.

Figure 3.10 describes the message formats of the proposed new contracting protocol. A negotiation can start with either a contractor or a contractee message, Fig. 3.11. A contractor message specifies exclusive alternative contracts that the contractor is willing to commit to. Within each alternative, the tasks can be split into disjoint

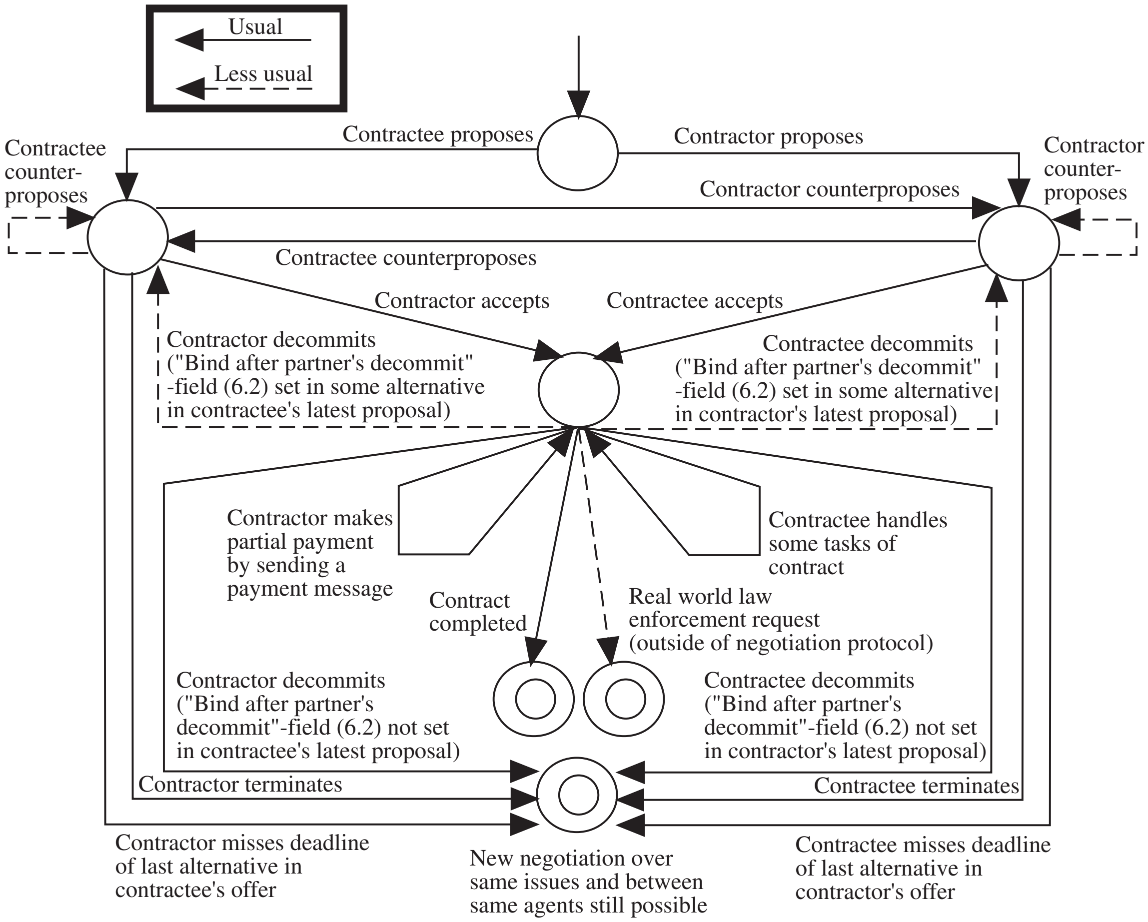

Figure 3.11. State transition diagram of a single negotiation.

task sets by the sender of the message in order for the fields (a) - (d) to be specific for each such task set - not necessarily the whole set of tasks. Each alternative has the following semantics. If the contractee agrees to handle all the task sets in a manner satisfying the minimum required task descriptions (a) (which specify the tasks and constraints on them, e.g. latest and earliest handling time or minimum handling quality), and the contractee agrees to commit to each task set with the level specified in field (d), then the contractor is automatically committed to paying[4] the amounts of fields (b), and can cancel the deal on a task set only by paying the contractee a

[4]Secure money transfer can be implemented cryptographically e.g. by electronic credit cards or electronic cash [Kalakota and Whinston, 1996, Kristol *et al.*, 1994, Low *et al.*, 1994b, Low *et al.*, 1994a].

penalty (c)[5]. Moreover, the contractor is decommitted from all the other alternatives it suggested[6]. If the contractee does not accept any of the alternatives, the contractor is decommitted from all of them. Fields (b), (c) and (d) can be functions of time, of negotiation events, or of domain events, and these times/events have to be observable or verifiable by both the contractor and the contractee. A contractee can accept one of the alternatives of a contractor message by sending a contractee message that has task specifications that meet the minimal requirements (a), and payment functions that meet the required payment functions (b), and commitment functions (c) for the contractee that meet the required commitment functions, and commitment functions (d) for the contractor that do not exceed the contractor's promised commitment. A contractor message can accept one of the alternatives of a contractee message analogously. An agent can entirely terminate a negotiation by sending a message with that negotiation's identifier (field 0), and the terminate-flag (field 5) set.

Alternatively, the contractee can send a contractee message that neither accepts the contractor message (i.e. does not satisfy the requirements) nor terminates the negotiation. Such a message is a *counterproposal*, which the contractor then can accept, terminate the negotiation, or further counterpropose etc. *ad infinitum.* [7] The original CNP did not allow counterproposing: an agent could bid to an announcement or decide not to bid. A contractor had the option to award or not to award the tasks according to the bids. Counterproposing among cooperative agents has been studied for example in [Moehlman *et al.*, 1992, Sen, 1994, Sen, 1996]. Our counterproposing

[5] The "Bind after partner's decommit" (6.2) flag describes whether an offer on an alternative will stay valid according to its original deadline (field 6.1) even in the case where the contract was agreed to, but the partner decommitted by paying the decommitment penalty.

[6] Another protocol would have offers stay valid according to their original specification (deadline) no matter whether the partner accepts, rejects, counterproposes, or does none of these. We do not use such protocols due to the harmfully (see Section 3.6) growing number of pending commitments.

[7] An agent that has just (counter)proposed can counterpropose again (dotted lines in Fig. 3.11). This allows it to add new offers (that share the "In-response-to"-field with the pending ones), but does not allow retraction of old offers. Retraction is problematic in a distributed system, because the negotiation partner's acceptance message may be on the way while the agent sends the retraction.

mechanism is one way of overcoming the problem of lacking truthful abstractions of the global search space (defined by the task sets and resource sets of all the agents) in negotiation systems consisting of self-interested agents.

There are no uncommittal messages in the protocol such as announcements used to declare tasks: all messages have some commitment specification for the sender. In early messages in a negotiation, these commitment specifications can be too low for the partner to accept, and counterproposing occurs. Thus, the level and stage of commitment are dynamically negotiated along with the negotiation of taking care of tasks.

3.5 Contract Types in Task Allocation Negotiation

In terms of search in the global task allocation space, contracts can be viewed as iterative refinement operators that move the single search focus from one task allocation to another. Assuming that each agent is willing to accept any individually rational contract at any time, all contract operators that enhance the social welfare are possible at any given task allocation. In other words, task reallocation contracting of the type that was used for example in TRACONET can be viewed as hill-climbing in the task allocation space, where the height-metric of the hill is social welfare. Social welfare is defined as having the same absolute value as the global cost, but the opposite sign.

This section discusses three new contract types (i.e. contract operators in global task allocation space) for task allocation negotiation: clustering contracts (Section 3.5.1), swap contracts (Section 3.5.2), and multiagent contracts (Section 3.5.3). Finally, Section 3.5.4 shows that these three can be combined into one contract type that is necessary and sufficient for reaching a global task allocation optimum with any hill-climbing algorithm (i.e. any sequence of individually rational contracts) in a finite number of steps without backtracking.

3.5.1 Clustering Contracts

In early CNP implementations, tasks were negotiated one at a time. This is insufficient in general, if the cost or feasibility of carrying out a task depends on the carrying out of other tasks. There may be local optima, where no transfer of a single task between agents enhances the global solution, but transferring a larger set of tasks simultaneously does.

This can be shown even with a very small example of just two tasks: t_1 and t_2, and two agents A and B. Say that the current task allocation is one where agent A's tasks are $T_A = \{t_1, t_2\}$, and B's task are $T_B = \emptyset$. Say that the task handling costs are as follows: $c_A(\emptyset) = 0$, $c_A(\{t_1\}) = c_A(\{t_2\}) = 4$, $c_A(\{t_1, t_2\}) = 5$, $c_B(\emptyset) = 0$, $c_B(\{t_1\}) = c_B(\{t_2\}) = 2$, and $c_B(\{t_1, t_2\}) = 3$. So, the current global cost is 5. Moving t_1 to B would increase the global cost to 6. So would moving t_2 to B. On the other hand, moving both t_1 and t_2 to B would decrease the global cost to 3. So, this clustering contract is the only profitable contract in this game, Fig. 3.12.

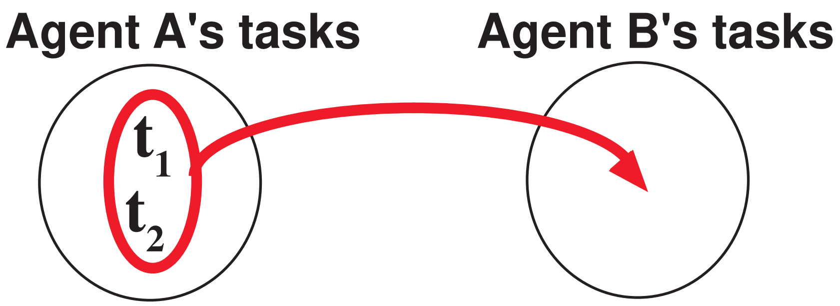

Figure 3.12. A clustering contract.

The need for larger transfers is well known in centralized iterative refinement optimization [Lin and Kernighan, 1971, Waters, 1987], but has been generally ignored in automated negotiation. In some non-automated allocation settings, the need for clusters has been realized. For example, when the Federal Communications Commission auctions airwave bandwidth for restricted geographical areas, the bidders' valuations for the auctioned items depend on what other items they are awarded [McAfee and McMillan, 1996]. For example, some bidders want to receive a cluster of awards that allows them to establish nationwide coverage. In the FCC auctions, explicit clustering was not used, but the agents could construct the clusters from individually auctioned items. A simultaneous ascending auction was used, where each agent sees the other agents' bids. Therefore, the agents could see which clusters they were likely to get, and bid accordingly. The auction was carried out in stages, where each stage had a quantitative activity rule. This disables any bidder from staying out at first, and then bidding on the last moment when it knows the others' bids. If the latter were possible, few agents would want to bid up front. The auction terminated when no agent wanted to raise its bids. The auction designers supported profitable clustering also by allowing agents to withdraw from their bid—e.g. if an agent did not get the cluster that it strived for. Withdrawing was allowed if the withdrawing agent guaranteed the bid price. Specifically, the item was opened for reauctioning, and if the new highest bid was lower than the old one, the withdrawing agent had to pay the difference.

TRACONET extended the CNP to handle task interactions by having the announcer *cluster tasks into sets to be negotiated atomically.* Alternatively, the bidder could have done the clustering by counterproposing. Our protocol generalizes this by allowing either party to do the clustering, Fig. 3.10, at any stage of the protocol.

In general, a cluster can encompass any number of tasks. A solution (task allocation) is called k-optimal if no beneficial clustering contract of any k tasks can be made between any two agents. This is a necessary, but not a sufficient condition for global optimality. Say that $m < n$ without loss of generality. Now, m-optimality does not imply n-optimality. More surprisingly, n-optimality does not imply m-optimality.

TRACONET used clustering of one, two, or three tasks per announcement. An interesting question is how to choose the clusters. TRACONET supports several policies for doing this. One is to use all 1-task clusters until 1-optimality has been reached, then all 2-task clusters until 2-optimality, then all 1-task clusters until 1-optimality then all 2-task clusters until 2-optimality. This continues until both 1- and 2-optimality have been reached. Then all 3-task clusters are used until 3-optimality. Then the process repeats. It stops when 1-, 2-, and 3-optimality have been reached. This process can also be stopped part way. Other schedules can be used as well. A second way to choose clustering is to interleave 1-task, 2-task, and 3-task contracts with the choice being made by domain independent heuristics. For example, TRACONET could cluster two tasks into one announcement whenever the marginal cost of the two was smaller than a constant times the sum of the marginal costs of the two separately. Similar domain independent heuristics were implemented for 3-task contracts. Domain specific clustering heuristics were also implemented, where clustering occurred if the deliveries were geographically close to each other. As expected, these clustering contracts proved useful for small artificially generated problem instances in the sense that a better global solution was reached than with pure 1-task contracts.

3.5.2 Swap Contracts

Sometimes there is no task set size such that transferring such a set from one agent to another enhances the global solution. Yet, there may be a beneficial *swap* of tasks, where the first agent subcontracts a task to the second and the second subcontracts another task to the first agent. The need for swap contracts can be demonstrated even with a very small example of just two tasks: t_1 and t_2, and two agents A and B. Say that the current task allocation is one where agent A's tasks are $T_A = \{t_1\}$, and B's task are $T_B = \{t_2\}$. Say that the task handling costs are as follows: $c_A(\emptyset) = 0$, $c_A(\{t_1\}) = 2$, $c_A(\{t_2\}) = 1$, $c_A(\{t_1, t_2\}) = 5$, $c_B(\emptyset) = 0$, $c_B(\{t_1\}) = 1$, $c_B(\{t_2\}) = 2$, and $c_B(\{t_1, t_2\}) = 5$. So, the current global cost is 4. Moving t_1 to B would increase the global cost to 5. So would moving t_2 to A. On the other hand, moving both t_1

from A to B and simultaneously t_2 from B to A would decrease the global cost to 2. So, this swap contract is the only profitable contract in this game, Fig. 3.13.

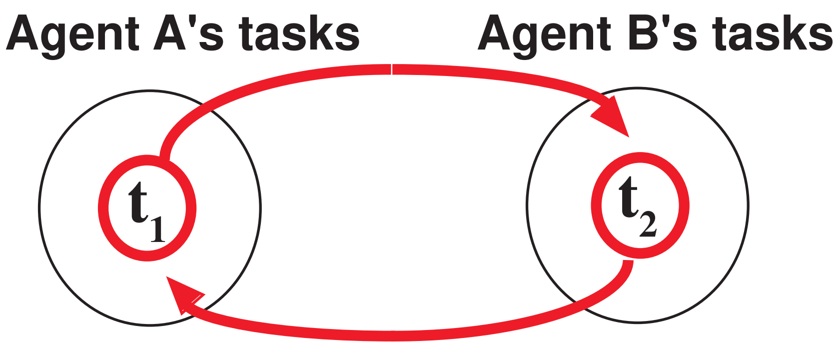

Figure 3.13. A swap contract.

The interaction protocols needed for benevolent (cooperative) agents and those needed for self-interested agents differ significantly. Cooperative agents can be assumed to take care of each others tasks without compensation whenever that is beneficial for the society of agents. Self-interested agents need some compensation to take care of some other agent's task. This compensation can be organized as barter trade: one agent takes care of some of another agent's tasks if the latter agent takes care of some of the first agent's tasks. Barter trades that benefit both agents do not always exist even if it were profitable for the community of agents to move a task from one agent to another. Secondly, the identification of beneficial barter exchanges is nontrivial—especially in a distributed setting. A finer resolution of cooperation among self-motivated agents can be achieved by a monetary compensation mechanism: an agent pays another agent to take care of some of its tasks. The need for swaps shows that payment based exchanges cannot replace all barter exchanges. What is needed is the monetary exchange method (that allows infinitely divisible side-payments) but also a linking mechanism that allows swapping tasks atomically among agents.

Swaps can be explicitly implemented in a negotiation protocol by allowing some task sets in an alternative (Fig. 3.10) to specify tasks to contract in and some to specify tasks to contract out. In the task sets added to implement swaps, "Minimum" in field (a) should be changed to "Maximum" and vice versa. In field (b), "Promised payment fn. to contractee" should be changed to "Required payment fn. from contractee" and "Required payment fn. to contractee" should be changed to "Promised payment fn. from contractee".

3.5.3 Multiagent Contracts

Negotiations may have reached a local optimum with respect to mutual contract operators (cluster contracts and swap contracts) of any size, but solution enhancements would be possible if tasks were transferred among more than two agents. [8] Decentralized multiagent contracts can be implemented for example by circulating the contract message among the parties and agreeing that the contract becomes valid only if every agent signs.

The need for multiagent contracts can be demonstrated even with a very small example of just three tasks: t_1, t_2, and t_3, and three agents A, B, and C. Say that the current task allocation is one where agent A's tasks are $T_A = \{t_1\}$, B's task are $T_B = \{t_2\}$, and C's tasks are $T_C = \{t_3\}$. Say that the task handling costs for agent A are as follows: $c_A(\emptyset) = 0$, $c_A(\{t_1\}) = 2$, $c_A(\{t_2\}) = 5$, $c_A(\{t_3\}) = 1$,

[8] Sathi and Fox have studied the role of grouping buy and sell bids into cascades which may involve multiple agents [Sathi and Fox, 1989]. Their setting is simpler than the one in this dissertation in that the value of a contract to an agent does not depend on which other ones of the agent's bids get accepted. On the other hand, in the settings of this dissertation, an agent's valuation of a contract depends significantly on which other ones of the agent's bids get accepted. The algorithms of Sathi and Fox also differ from those of this dissertation in that they do not allow recontracting once a contract has been made. From a search perspective this means that they do not allow backtracking. They present three heuristic algorithms for choosing the order in which to execute possible contracts. The order is important because one contract can preclude another if the two involve the allocation of the same resource. Because backtracking is not used, the former contract cannot be undone to allow the latter contract, even if the latter is more beneficial than the former. The heuristics focus on handling more constrained requests first. However, they do not guarantee that the globally optimal solution is reached. Some of the algorithms are distributed while others involve centralized processing regarding the global situation.

$c_A(\{t_1, t_2\}) = c_A(\{t_1, t_3\}) = c_A(\{t_2, t_3\}) = 10$, $c_A(\{t_1, t_2, t_3\}) = 15$, and for agent B: $c_B(\emptyset) = 0$, $c_B(\{t_1\}) = 1$, $c_B(\{t_2\}) = 2$, $c_B(\{t_3\}) = 5$, $c_B(\{t_1, t_2\}) = c_B(\{t_1, t_3\}) = c_B(\{t_2, t_3\}) = 10$, $c_B(\{t_1, t_2, t_3\}) = 15$. Say that the costs for agent C are $c_C(\emptyset) = 0$, $c_C(\{t_1\}) = 5$, $c_C(\{t_3\}) = 2$, $c_C(\{t_2\}) = 1$, $c_C(\{t_1, t_2\}) = c_C(\{t_1, t_3\}) = c_C(\{t_2, t_3\}) = 10$, $c_C(\{t_1, t_2, t_3\}) = 15$. So, the current global cost is $2+2+2 = 6$. Cluster contracts of more than one task are impossible here because no agent has more than one task. Any one of the six possible 1-task (cluster) contracts would increase global cost to $0 + 2 + 10 = 12$. Any one of the three possible swaps would increase global cost to $1 + 2 + 5 = 8$. Therefore no cluster contract or swap contract is profitable.

However, moving both t_1 from A to B, and t_2 from B to C, and t_3 from C to A would decrease the global cost to 3. So, this multiagent contract is the only profitable contract in this game, Fig. 3.14.

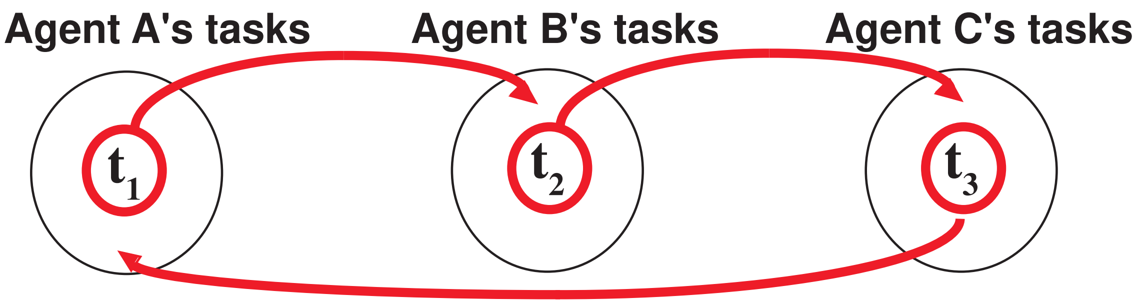

Figure 3.14. A multiagent contract.

3.5.4 Necessity and Sufficiency

When applied alone, no one of the presented three contract types is sufficient for reaching the global optimum. The example in Section 3.5.2 shows that clustering contracts are not sufficient, and the example in Section 3.5.3 shows that swap contracts are not sufficient. Finally, the example in Section 3.5.1 shows that multiagent contracts (at most one task from each agent) are not sufficient.

Similarly, none of the three contract types are sufficient in pairs. The example in Section 3.5.1 shows that swaps and multiagent contracts are not sufficient together. The example in Section 3.5.3 shows that clustering contracts and swaps are not sufficient together. Finally, the example in Section 3.5.2 shows that clustering contracts and multiagent contracts (with no mutual swap between two agents) are not sufficient together.

When applied one operator at a time, even the three contract operators do not form a sufficient set for reaching a global task allocation optimum. This can be shown e.g. via a counterexample where the only profitable contract is one where two agents swap more than two tasks. This contract would therefore need to combine clustering and swaps. Such examples also exist for other combinations of the three contract types.

So, let us define a new contract type, clustering-swap-multiagent contract (CSM-contract), that combines the characteristics of the three previously introduced contract types into one contract type. A CSM-contract lists for each possible pair of agents simultaneously

- the tasks that are to be transferred from the former agent to the latter, and
- the tasks that are to be transferred from the latter agent to the former, and
- the payment that is to be transferred between the two agents.

The following theorem states that CSM-contracts are sufficient for reaching the global task allocation optimum in a finite number of contracts. The result holds for any sequence of individually rational CSM-contracts, i.e. for any hill-climbing algorithm that uses CSM-contracts as iterative refinement search operators in the task allocation space. This means that from the perspectives of social welfare maximization and of individual rationality (not necessarily from the perspective of local payoff maximization), agents can accept individually rational contracts as they are offered. They need not wait for more profitable ones, and the need not worry that a current contract may make a more profitable future contract unprofitable. Neither do they need to accept contracts that are not individually rational in anticipation of future contracts that make the combination beneficial. Furthermore these hill-climbing algorithms do not need to backtrack.

Theorem 3.14 *Let there be a finite number of agents and tasks. If the agents can pick any CSM-contracts, any hill-climbing algorithm finds the global task allocation optimum in a finite number of steps without backtracking.*

This theorem gives a powerful tool for small problem instances where the number of possible task allocations is relatively small. On the other hand, for large problem instances, the number of contracts made before the globally optimal task allocation is reached may be impractically large—albeit finite. In such settings, the anytime character of the contracting scheme is more important. One should not even attempt to reach the global optimum, but instead compute as long as there is time, and then have a solution ready. The presented iterative refinement contracting methods guarantee this, and furthermore, they guarantee that the solution to each agent is no worse than the initial solution where the agent worked individually with its own tasks only. For example on the large-scale real-world distributed vehicle routing problem instance, TRACONET never reached a local optimum even with just basic 1-task contracts—even with multiple hours of negotiation time on five Unix machines.

The equivalent of a complex contract (clustering, swap, or multiagent contract, or any CSM-contract) can be accomplished by a sequence of basic 1-task contracts if the agents are willing to take risks. Even if no 1-task contract is individually beneficial, the agents can sequentially make all the small contracts that sum up to a large beneficial one. Early in this sequence, the global solution degrades until the later contracts enhance it. When making the early commitments, at least one of the agents has to risk taking a permanent loss in case the other agent(s) do not agree to the later contracts that are needed to make the sequence of contracts profitable. Our leveled commitment protocol decreases such risks as much as preferred by allowing agents to break commitments by paying a penalty. If the later contracts in the sequence do not happen, the agent can decommit from the earlier ones. The decommitment penalty function can also be explicitly conditioned on the acceptance of the future contracts, or it may specify low commitment for a short time during which the agent expects to make the remaining contracts of the sequence.

3.6 Contracting Implications of Limited Computation

Interactions of self-interested agents have been widely studied in microeconomics [Kreps, 1990, Varian, 1992, Raiffa, 1982] and DAI [Rosenschein and Zlotkin, 1994, Ephrati and Rosenschein, 1991, Kraus *et al.*, 1992, Durfee *et al.*, 1993], but perfect rationality of the agents has usually been assumed: flawless deduction, optimal reasoning about future contingencies and recursive modeling of other agents. Perfect rationality implies that agents can compute their marginal costs for tasks exactly and immediately, which is untrue in most practical situations. An agent is bounded rational, because its computation resources are costly, or they are bounded and the environment keeps changing—e.g. new tasks arrive and there is a bounded amount of time before each part of the solution is used [Garvey and Lesser, 1994, Sandholm and Lesser, 1994, Zilberstein, 1993, Simon, 1982, Good, 1971]. Contracting agents have the following additional real-time pressures:

- A counteroffer or an acceptance message has to be sent by a deadline (field 6.1)—otherwise the negotiation terminates, Fig. 3.11. If the negotiation terminates, the agent can begin a new negotiation on the same issues, but it will not have the other agent's commitment at first.

- Sending an outgoing offer too late may cause the receiving agent to make a contract on some of the same tasks with some other agent who negotiated earlier—thus disabling this contract even if the offer meets the deadline. In case this deadline abiding offer is an acceptance message—as opposed to a counteroffer—the partner has to pay the decommitment penalty that it had declared.

- The (b)-(d) fields can be functions of response time, Fig. 3.10. An agent may get paid less for handling tasks (or pay more for having tasks handled) or be required to commit more strongly or receive a weaker commitment from the negotiation partner if its response is postponed.

- The agent's cost of breaking commitments (after a contract is made) may increase with time.

3.6.1 Local Deliberation Scheduling

This problem setup leads to a host of local deliberation scheduling issues. An agent has to decide *how much computation* it should allocate to refine its marginal cost estimate of a certain task set. With a bounded CPU, if too much time is allocated, another agent may win the contract before the reply is sent, or not enough time remains for refining marginal costs of other task sets. If too little time is allocated, the agent may make an unbeneficial contract concerning that task set. If multiple negotiations are allowed simultaneously, the agent has to decide on *which sets of tasks* (offered to it or potentially offered by it) its bounded computation should be focused. It may want to ignore some of its contracting possibilities in order to focus more deliberation time to compute marginal costs for task sets of some selected potential contracts. So, there is a tradeoff of getting more exact marginal cost estimates and being able to engage in a larger number of negotiations. Finally, an agent has to decide the *sequence in which it allocates the deliberation to different task sets.* The basic intuition is that the sequence should be chosen so that more urgent task sets are considered first. However, such a greedy heuristic is not adequate: the sequencing problem is a hard combinatorial problem.

3.6.2 Engaging in Multiple Simultaneous Negotiations

The original CNP did not consider an agent's risk attitude toward being committed to activities it may not be able to honor, or the honoring of which may turn out unbeneficial. In our protocol, an agent can take a risk by making offers while the acceptance of earlier committal offers is pending. Contracting during pending commitments speeds up the negotiations because an agent does not have to wait for results on earlier commitments before carrying on with other negotiations. Let us now discuss formalize the risk taking question. We will look at an announce-bid-award contracting protocol where bids and awards have full commitment, but announcements do not have any commitment—as in TRACONET.

Let us first discuss risk taking in bidding. Denote an arbitrary bid by b and the set of tasks of that bid by T^b. Let $B_j^{unsettled}$ be the set of unsettled bids sent by agent j previously. Define $B_j^{pos}(b)$ to be a set of possible bids that can be awarded to agent j when b is also awarded it, i.e., $B_j^{pos}(b) = \{x | x \in B_j^{unsettled}, T^x \cap T^b = \emptyset\}$. This set can be further divided into (possibly overlapping) subsets based on the idea that if some agent has sent announcements that overlap in tasks, it can only award the tasks of one of those announcements. Specifically, let $\mathcal{B}_j^{pos}(b)$ denote the set of (maximally sized) possible sets of bids that can get awarded to agent j. Formally, $\mathcal{B}_j^{pos}(b) = \{B_j^{pos}(b) | [x \in B_j^{pos}(b), x' \in B_j^{pos}(b)] \Rightarrow T^x \cap T^{x'} = \emptyset\}$.

Let T_j be the bidder's current set of tasks. As mentioned in Section 3.1.2, an agent bids its marginal cost which was computed as follows:

$$c_j^{add}(T^b | T_j) = c_j(T_j \cup T^b) - c_j(T_j)$$

Here again $c_j(T)$ is the cost of the optimal routing solution for tasks T with the vehicles of agent j. The marginal cost calculation above assumes that none of the agent's currently pending bids get accepted. On the other hand, to make sure that the current bid is beneficial in all of the alternative worlds induced by different combinations of acceptances of pending bids, the bidder would have to compute the marginal cost as follows:[9]

$$\hat{c}_j^{add}(T^b|T_j) = \max_{B \subseteq Y, Y \in \mathcal{B}_j^{pos}(b)} [c_j(T_j \cup T^b \bigcup_{z \in B} T^z) - c_j(T_j \bigcup_{z \in B} T^z)]$$

This is expensive to compute because the number of alternative worlds induced by different combinations of acceptances of pending bids is exponential in pending bids. To trade off some of this computational complexity against monetary risk, a bidder can approximate this marginal cost by $c_j^{add}(T^b|T_j)$. This incurs risk. For some combination of acceptances of pending bids, the current bid may be unbeneficial, and the bidder may have to accept that contract at a loss. Another way to trade off computational complexity against monetary risk is to approximate the marginal cost by assuming that all of the pending bids get accepted (although sometimes this cannot happen based on the above argument of announcements that overlap in tasks):

$$\tilde{c}_j^{add}(T^b|T_j) = c_j(T_j \cup T^b \bigcup_{z \in B_j^{pos}(b)} T^z) - c_j(T_j \bigcup_{z \in B_j^{pos}(b)} T^z)$$

Note that either one of $c_j^{add}(T^b|T_j)$ or $\tilde{c}_j^{add}(T^b|T_j)$ may be greater than the other, because in the vehicle routing problem, both economies of scale and diseconomies of scale are present. On the other hand, $\hat{c}_j^{add}(T^b|T_j) \geq c_j^{add}(T^b|T_j)$ in any domain by definition. It is faster to compute $c_j^{add}(T^b|T_j)$ than $\tilde{c}_j^{add}(T^b|T_j)$: the former is constant in pending bids, while the latter requires computing the union of the tasks of pending bids (avoiding duplicates). This is $O(N \log N)$ in the total number of tasks mentioned in these bids. The former also gives a better approximation of the marginal cost when bids are seldom awarded to the agent. This is usually the case, if the network has many agents.

[9]A very risk seeking approach would be to assume that the best combination of pending bids gets accepted. This corresponds to changing the max-operator to a min-operator.

In TRACONET, computational complexity was traded off against risk by using the $c_j^{add}(T^b|T_j)$ concept for marginal cost estimation.[10] This choice was static, but more advanced agents should use a risk taking strategy where negotiation risk is dynamically and explicitly traded off against added computation.

In addition to approximating the marginal cost by $c_j^{add}(T^b|T_j)$, a feasibility check was done in the vein of $\tilde{c}_j^{add}(T^b|T_j)$ to make sure that the bidder can have a feasible (not necessarily profitable) solution even if all of the pending bids get accepted.[11] In domains (unlike ours), where the feasibility check often restricts the bidding, the bidder should choose the most profitable combination among the possible combinations of beneficial bids to send.

In the original CNP, an agent could have multiple bids concerning different contracts pending concurrently in order to speed up the operation of the system. We have followed this approach for the same reason, although negotiations over only one contract at a time allow a more precise marginal cost calculation for bidding. If only one bid is allowed to be pending from one agent at a time, $B_j^{pos}(b) = \emptyset$, and $c_j^{add}(T^b|T_j) = \tilde{c}_j^{add}(T^b|T_j) = \hat{c}_j^{add}(T^b|T_j)$. In other words, trading off computational complexity against monetary risk would not be necessary.

Similar risk taking considerations apply to agent i that is considering awarding a task set to a bidder. In the awarding phase the manager has a chance to check that awarding is still beneficial to itself, i.e., it does not have to accept any bid. In deciding whether the awarding is beneficial, the agent has to also consider the unsettled bids that it has sent. Awarding to bid b is beneficial if and only if the bid price is still lower than the current marginal cost for the task set T^b. This marginal cost is the cost of removing the tasks T^b from the agent's local routing solution. Unlike in the bidding phase, $B_i^{pos}(b) = B_i^{unsettled}$ because there is no chance that the agent's pending

[10] Actually, the c_j's are intractable to compute, so $c_j^{add}(T^b|T_j)$ was approximated directly by simulating the addition of the task set into the current (non-optimal) routing solution.

[11] In the vehicle routing problem, the addition of tasks can make a local solution infeasible, but the removal of tasks cannot.

bids include tasks that the agent is currently awarding to others. As in the bidding situation, the set $B_i^{pos}(b)$ can be further divided into (possibly overlapping) subsets based on the idea that if some agent has sent announcements that overlap in tasks, it can only award the tasks of one of those announcements. Specifically, let $\mathcal{B}_i^{pos}(b)$ denote the set of (maximally sized) possible sets of bids that can get awarded to agent i. Formally, $\mathcal{B}_i^{pos}(b) = \{B_i^{pos}(b) | [x \in B_i^{pos}(b), x' \in B_i^{pos}(b)] \Rightarrow T^x \cap T^{x'} = \emptyset\}$.

Let T_i' be the awarder's current set of tasks. As mentioned in Section 3.1.2, an agent bids its marginal cost which was computed as follows:

$$c_i^{remove}(T^b|T_i') = c_i(T_i') - c_i(T_i' \cap T^b) \tag{3.20}$$

Here again $c_i(T)$ is the cost of the optimal routing solution for tasks T with the vehicles of agent i. The marginal cost calculation above assumes that none of the agent's currently pending bids get accepted. On the other hand, to make sure that the current award is beneficial in all of the alternative worlds induced by different combinations of acceptances of pending bids, the awarder would have to compute the marginal cost as follows:[12]

$$\hat{c}_i^{remove}(T^b|T_i') = \min_{B \subseteq Y, Y \in \mathcal{B}_i^{pos}(b)} [c_i(T_i' \bigcup_{z \in B} T^z) - c_i(T_i' \bigcup_{z \in B} T^z \cap T^b)]$$

This is expensive to compute because the number of alternative worlds induced by different combinations of acceptances of pending bids is exponential in pending bids. To trade off some of this computational complexity against monetary risk, an awarder can approximate this marginal cost by $c_i^{remove}(T^b|T_i')$. This incurs risk. For some combination of acceptances of pending bids, the current award may turn out unbeneficial. Another way to trade off computational complexity against monetary risk is to approximate the marginal cost by assuming that all of the pending bids get

[12] A very risk seeking approach would be to assume that the best combination of pending bids gets accepted. This corresponds to changing the min-operator to a max-operator.

accepted (although sometimes this cannot happen based on the above argument of announcements that overlap in tasks):

$$\tilde{c}_i^{remove}(T^b|T_i') = [c_i(T_i' \bigcup_{z \in B_i^{pos}(b)} T^z) - c_i(T_i' \bigcup_{z \in B_i^{pos}(b)} T^z \cap T^b)]$$

Note that either one of $c_i^{remove}(T^b|T_i')$ or $\tilde{c}_i^{remove}(T^b|T_i')$ may be greater than the other, because in the vehicle routing problem, both economies of scale and diseconomies of scale are present. On the other hand, $\hat{c}_i^{remove}(T^b|T_i') \leq c_i^{remove}(T^b|T_i')$ in any domain. It is faster to compute $c_i^{remove}(T^b|T_i')$ than $\tilde{c}_i^{remove}(T^b|T_i')$: the former is constant in pending bids, while the latter requires computing the union of the tasks of pending bids (avoiding duplicates). This is $O(N \log N)$ in the total number of tasks mentioned in these bids. The former also gives a better approximation of the marginal cost when bids are seldom awarded to the agent. This is usually the case, if the network has many agents.

In TRACONET, computational complexity was traded off against risk by using the $c_i^{remove}(T^b|T_i')$ concept for marginal cost estimation.[13] This choice was static, but more advanced agents should use a risk taking strategy where negotiation risk is dynamically and explicitly traded off against added computation. Unlike in sending a bid, in sending an award, an agent does not need to do a feasibility check. The removal of the awarded tasks from the local routing solution cannot make any future task combination infeasible if it would have been feasible without sending the award. However, the basic check of profitability may make the order of awarding important—although this is seldom the case in our domain in practice. The awarding of one task set may disable the beneficial awarding of another. Usually the number of received bids per local announce-bid-award cycle is small, so the awarder could try all combinations of possible awards and carry out the most profitable one.

[13] Actually, the c_j's are intractable to compute, so $c_i^{remove}(T^b|T_i')$ was approximated directly by simulating the removal of the task set from the current (non-optimal) routing solution.

If an agent sends awards only when it has none of its bids pending, $B_i^{pos}(b) = \emptyset$, and $c_i^{remove}(T^b|T_i') = \tilde{c}_i^{remove}(T^b|T_i') = \hat{c}_i^{remove}(T^b|T_i')$. In other words, trading off computational complexity against monetary risk would not be necessary. Figure 3.15 compares results of allowing bidding and awarding while bids pend versus bidding and awarding only when bids do not pend. In the former case (left column), agents can participate in more negotiations per local announce-bid-award loop than in the latter case (right column). On the other hand, in the latter case, each local announce-bid-award loop takes less time due to lower computational complexity, and therefore more local loops fit into the two hours of real-time of this experiment than in the former case. Furthermore, in the latter case the cost decrease is guaranteed to be monotonic while in the former case an agent may have to take occasional unbeneficial awards due to risk taking: $c_j^{add}(T^b|T_j)$ and $c_i^{remove}(T^b|T_i')$ were used to approximate marginal cost.

3.6.3 Enlarging the Bidding and Awarding Context

An agent does not know how many bidders will show up and what their bids will be. Taking the first profitable bid corresponds to the first-swap iterative refinement algorithm in the global task allocation search space. Waiting for all the bids corresponds to the best-swap algorithm in the global search space. Undoing the first swap for a better swap is possible in the leveled commitment protocol.

In general, there is a tradeoff between accepting or (counter) proposing early on and waiting:

- A better offer may be received later.
- Waiting for more simultaneously valid offers enables an agent to identify and accept synergic ones: having more options available at the decision point enables an agent to make more informed decisions.

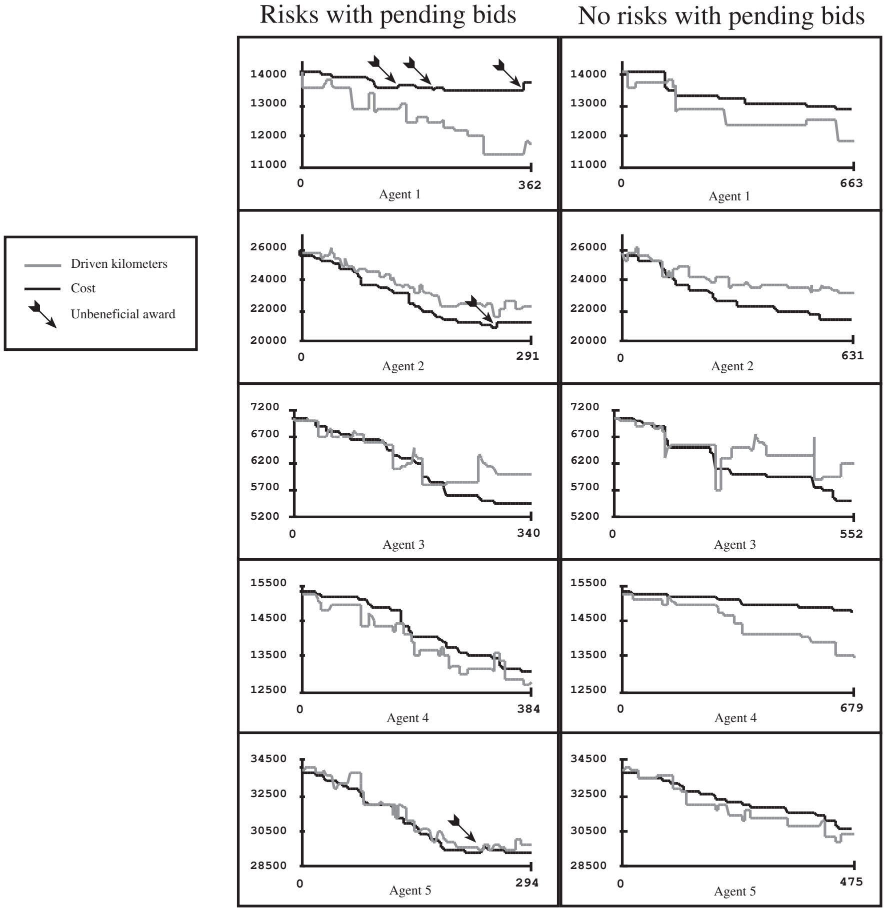

Figure 3.15. Safe vs. risky bidding and awarding: example runs. Two 2-hour runs (left column and right column) on five Unix machines with the real data from five dispatch centers are shown. The category axis shows the number of local announce-bid-award cycles for each agent. The gray curve shows the total length of an agent's truck routes. The black curve shows the local cost for each agent, which takes into account the payments by the contractor to the contractee for carrying out tasks.

- Accepting early on simplifies costly marginal cost computations, because there are fewer options to consider. An option corresponds to an item in the power set of offers that an agent can accept or make.
- By waiting an agent may miss opportunities due to others making related contracts first.

3.7 Distributed Asynchronous Implementation

Much of the automated contracting research done by others has been carried out using centralized simulations of distributed problem solving [Smith, 1980, Smith, 1978, Smith, 1979, Davis and Smith, 1981, Davis and Smith, 1983, Smith and Davis, 1981, Sen, 1994, Sen, 1996]. These approaches have ignored many important problems relating to concurrency and asynchronous message passing that arise in truly distributed implementations of automated contracting systems. The fact that the TRACONET system is truly distributed and uses asynchronous message passing has helped uncover several previously unaddressed implementation questions in automated contracting. The next sections discuss some of these problems and present potential solutions.

3.7.1 Avoiding Message Congestion and Saturation

Message congestion is a major problem in automated contracting [Sandholm, 1993, Smith, 1980, Van Dyke Parunak, 1987]. While an agent takes a long time to process a large number of received messages, even more messages have time to arrive, and there is a high risk that the agent will finally be saturated. This means that it will spend all of its deliberation time on messages that will, by the time that the agent is able to answer, so outdated that they will not lead to contracts. This may happen because other agents have bid and been awarded the contract earlier, or no agent has been awarded the contract, but the deadline of the announcement has passed.

Previous attempts to solve this problem include focused addressing [Smith, 1980] and audience restrictions [Van Dyke Parunak, 1987, Sandholm, 1993]. Focused addressing means that in highly constrained situations, agents with free resources announce availability, while in less constrained situations, agents with tasks announce tasks. This avoids announcing too many tasks in highly constrained situations, where these announcements would seldom lead to results. In less constrained environments, resources are plentiful compared to tasks, so announcing tasks focuses negotiations with fewer messages. Audience restrictions mean that an agent can only announce to a subset of agents which are supposedly most potential.

Focused addressing and audience restrictions are imposed on an agent by a central designer of the agent society. Neither is necessarily viable in open systems with self-interested agents. Such an agent will send a message whenever that is beneficial to itself even though this might saturate other agents. With flat rate media such as the Internet, an agent prefers sending to almost everyone who has non-zero probability of accepting/counterproposing. The society of agents would be better off by less communication via restricted sending, but each agent sends as long as the expected utility from that message exceeds the decrease in utility to that agent caused by the congesting effect of that message. This defines a *tragedy of the commons* (n-player prisoners' dilemma) [Turner, 1992, Hardin, 1968]. The tragedy occurs mainly for low commitment messages (usually early in a negotiation): having multiple high commitment offers out simultaneously increases an agent's negotiation risk (Sec. 3.2) and computation costs (Sec. 3.6), so a self-interested agent will avoid this.

One way to resolve the tragedy is a use-based communication charge. This changes the incentive structure of a self-interested agent so that it will send fewer messages.

Another way is mutual monitoring: an agent can monitor how often a certain other agent sends low commitment messages to it, and over-eager senders can be punished. By mutual monitoring, audience restrictions can also be implemented: if an agent receives an announcement although it is not in the appropriate audience, it can directly identify the sender as a violator. One problem with mutual monitoring is that mechanisms are required to motivate the agents to monitor and punish.

A third way is presented in our proposed contracting protocol (Section 3.4) which allows an agent to determine in its offer (field 6.4) a processing fee that an accepting or counterproposing agent has to submit in its response (field 6.3) for the response to be processed. This implements a self-selecting dynamic audience restriction that is viable among self-interested agents.

In addition to these potential remedies that have to do with changing the contracting protocol, the TRACONET experiments have led to the development of practical methods that can be incorporated in an agent's local strategy in order to reduce message congestion. These methods are in concert with an agent's self-interest because they help an agent avoid saturation locally. For example, the announcer, bidder, and awarder are structured as separate components in an agent's software architecture, and the local program control loops through each one of these in order. When one of these is triggered, it reads in the appropriate messages (the bidder reads announcements, and the awarder reads bids) that have arrived so far. It will not read messages that arrive during its execution. This prevents the agent from getting stuck for an infinite period of time in any particular stage even if large numbers of messages are arriving.

Even with this type of local computation scheduling, some agents got saturated by announcements on every one of our experimental runs, i.e., agents were receiving announcements at a faster pace than they could process. The problem occurred only with announcements, because in our domain the number of them far exceeds the number of other messages. The reason the congested agents could not keep in pace was that the time to handle an announcement increased with the number of previously sent unsettled bids—mainly because of the feasibility check discussed in Section 3.6.2. The more announcements an agent had received, the more bids it was able to make, which slowed it down, and during the bidding process even more announcements were received. The congestion problem was completely solved for our setting by making the bidder consider only announcements newer than a certain time limit. This is sensible also because bids made on older announcements would probably not get to the awarders before the negotiations concerning these announcements would be over.

So, the bidder makes a linear pass through the received announcements and ignores the ones that are older than a certain time limit. Theoretically, there is a chance that even this linear pass takes so long that all responses will be so late as to be in vain. In practice, this never happened in our system.

3.7.2 Terminating Iterative Refinement Negotiations

Knowing when to terminate distributed search is difficult [Durfee and Lesser, 1989, Durfee and Lesser, 1991]—especially with distributed iterative refinement algorithms [Sandholm, 1993] that are not based on a systematic and often slow backtracking scheme. TRACONET used the following termination heuristic for distributed iterative refinement: an agent stops negotiating once it has made no contracts during a certain fixed number of negotiation iterations.

Since then we have developed an exact termination protocol for iterative refinement negotiations. With cooperative, exactly computing agents it guarantees that negotiations end exactly when a local cost minimum in task allocation space has been reached with respect to clustering contracts and each agent's potential local refinement operators. The protocol can be used to reach local optimality with respect to any desired set of cluster contracts, e.g. 1-optimality, 2 -optimality, 1- and 2-optimality, etc. It does not assume that agents have knowledge of each others' tasks.

The idea is that the agents inform each other of situations when they have tried all possible local search operators and announced all possible (with respect to the possible restriction on the number of tasks per announcement) task sets without success. If an agent that has sent this stalemate status information to other agents later gets an award (tasks are allocated to it by contracting) or a domain event (new task from the environment or resource status change, e.g. breakdown), it retracts its stalemate status by sending a message to the other agents. When an agent is in stalemate status, it is still worth while for it to continue announcing and bidding, because other agents may not be in a stalemate and their task allocation may change. This may allow another agent to accept a proposed contract which it would not have accepted earlier. Once all agents are in a stalemate simultaneously, a local optimum has been reached, and the negotiations should be terminated until new domain events occur or new agents log onto the negotiation net. The termination protocol is appropriate in domains where there is relatively little domain volatility, i.e. new tasks and changes in resource status. If there is high domain volatility, a local optimum will never be reached, because the negotiations take a long time in comparison to the domain changes, and in general, each domain change redefines the set of local optima.

This method could be extended to swap contracts and multiagent contracts, but it would be combinatorially explosive in the number of agents. Instead of each agent posting a stalemate status, each subgroup would globally post a stalemate status stating that all possible contracts of the desired type have been tried within the subgroup without success.[14] When all subgroups would be in stalemate status, an optimum of the desired type would have been reached.

When used among computationally limited cooperative agents, even the basic termination method for clustering contracts may terminate the negotiations before a local optimum is reached. For example, an agent may omit reacting to an announcement in order to allocate more time to process other messages. This makes the announcing agent think that the receiver of the announcement could not bid, i.e. that there was no possibility of a beneficial contract. This may lead to termination of the negotiations while the agents erroneously believe that a local optimum has been reached.

The termination protocol also fails to guarantee a local optimum (or even to terminate at all) when used among rational but self-interested agents. First, a self-interested agent may not bid—due to strategic manipulation—even if it could based

[14] In order to know that all such operators have been tried within the subgroup, some level of joint knowledge of the subgroup's tasks would need to be formed.

on marginal cost calculations. Therefore an agent cannot infer the required amount of information from another agent's decision not to bid. Secondly, there is some cost associated with sending each stalemate status message, and it is not always in an agent's self-interest to do so.

3.7.3 Replies vs. Timeouts

The (6.1) field in Figure 3.10 describes how long an offer on an alternative is valid. If the negotiation partner has not answered by that time, the sender of the message gets decommitted from that alternative. An alternative to these strict deadlines is to send messages that have the (b) field be a function of the time of response (similarly for (c) and (d) fields). This allows a contractor to describe a payment that decreases as the acceptance of the contractor message is postponed. Similarly, it allows a contractee to specify required payments that increase as the acceptance of the contractee message is postponed. This motivates the negotiation partner to respond quickly, but does not force a strict deadline, which can inefficiently constrain that agent's local deliberation scheduling. Both the strict deadline mechanism and this time-dependent payment scheme require that the sending or receival time of a message can be verified by both parties.

An alternative to automatic decommitment by the deadline is to have the negotiation partner send a negative reply (negotiation termination message) by the deadline. These forced response messages are not viable among self-interested agents because an agent that has decided not to accept or counterpropose does not necessarily have any reason to send a reply. Sending reply messages also in negative cases allows the offering agent to decommit before the validity time of its offer ends. This frees that agent from considering the effects of the possible acceptance of that offer on the marginal costs of other task sets that the agent is negotiating over. This saved computation can be used to negotiate faster on other contracts. Thus, an agent considering sending a negative reply may want to send it in cases where the offering agent is mostly negotiating with that agent, but not in cases where the offering agent is that agent's competing offerer in most other negotiations. To avoid such speculation, we focus on time-based protocols as discussed above as opposed to ones that require replies.

3.8 Chapter Summary

This chapter on automated contracting presented extensions to the contract net protocol to allow it to work among self-interested computationally limited agents. In our setting, agents reallocate tasks to each other for dynamically constructed charges. Using this scheme, a more profitable global task allocation is reached than the initial one, while not executing a centralized task allocation algorithm. The setting under analysis is more general and realistic than most task allocation settings analyzed before in the multiagent systems literature because both economies and diseconomies of scale are present.

First, the original contract net scheme was enhanced by introducing a formal model for making bidding and awarding decisions. These decisions are made entirely based on each agent's local marginal cost calculations. In combinatorial problems such as vehicle routing, these calculations are intractable, so approximation algorithms are used. Therefore, the beneficiality of a contract is be measured with respect to the actual (non-optimal) routing solution that will be implemented, not the optimal routing solution which would be intractable to compute. A truly distributed implementation of these ideas was presented, where each agent runs in its own Unix process, and they communicate asynchronously over the network. Previous contract net implementations have been centralized simulations. The scheme leads to a distributed anytime task allocation algorithm, which can be terminated at any time, and is still guaranteed to have a feasible solution ready that is no worse for any agent than its initial routing solution. It was shown in the distributed vehicle routing domain—with real vehicle and delivery order data from five dispatch centers—that this technology has the possibility to scale up to large-scale real-world instances of combinatorial problems.

In Section 3.2, a decommitment mechanism was presented for automated contracting protocols that allows the agents to accommodate future events more profitably than traditional full commitment contracts. Each contract specifies a decommitment penalty for both agents involved. To decommit, an agent just pays that penalty to the other agent. This mechanism is better suited for complex computerized contracting settings than contingency contracts because potentially combinatorial and hard to anticipate contingencies need not be considered and conditioned on, no event verification mechanism is necessary, and decommitting can be decided based on local *ex post* deliberation. The method was analyzed using a normative approach: given the protocol, what strategy is each self-interested payoff maximizing agent best off choosing, and then what social outcomes follow. The game-theoretic analysis of the decommitting games handled the possibility that agents decommit manipulatively: an agent tries to avoid the decommitment penalty in case it believes that there is a high probability that it will be freed from the contract's obligations due to the other agent decommitting. This analysis also serves as a normative tool for agents to decide which contracts they should accept based on individual rationality.

In the presented games, leveled commitment contracts are a superset of full commitment ones. The latter can be emulated by setting the decommitment penalties sufficiently high. Therefore, full commitment protocols cannot be better than leveled commitment ones in the sense of Pareto efficiency or social welfare. Neither can they enable a deal that is impossible—based on individual rationality—via a leveled commitment contract. In game types where no opportunities (outstanding outside offers) become void between the contracting and the decommitting time (game types DOP, SEQD, SIMUDBP, and SIMUDNP), there are instances where the new protocol enables contracts that are impossible (not individually rational to the agents) using full commitment contracts. Also, in these game types, leveled commitment contracts improve each agent's expected payoff over any full commitment contract as long as there is some chance that the contractor's outside offer is lower than the expected value of the contractee's, or some chance that the contractee's outside offer is higher than the expected value of the contractor's. Obviously one can also construct game instances where the null deal is so profitable to both agents that no contract—even a leveled commitment one—is individually rational to the agents. In the COBV game where one agent loses an opportunity (outstanding outside offer) by agreeing to a contract, a leveled commitment contract can enable a deal or Pareto improve a deal over a full commitment contract only if that agent's fallback payoff is sufficiently high.

In the DOP and COBV games where only one agent's future outside offer involves uncertainty, the agent with a certain outside offer prefers to not decommit if the contract is originally individually rational to it. Thus only one agent may want to decommit. In these games, an agent's payoff to a contract is unaffected by the other agent's beliefs. Thus also the preference order over contracts is unaffected by the other agent's possibly biased beliefs. It follows that an agent need not counterspeculate its negotiation partner's beliefs, and that an agent cannot incur a loss due to the other agent's erroneous beliefs. On the other hand, in the SEQD, SIMUDBP, and SIMUDNP games where both agents' future outside offers involve uncertainty, an agent's payoff to a contract may depend on the negotiation partner's possibly biased beliefs because they affect the other partner's decommitting decision.

In Section 3.3, different stages of commitment were discussed. It was shown that commitment has to occur at some stage, but this stage can be made a negotiated item. It need not be fixed in the protocol. Section 3.4 presented an example protocol that allows this in addition to allowing other types of flexibility: counterproposing, alternative offers, etc.

Section 3.5 showed that classical contracts of one task at a time are insufficient to reach the globally optimal task allocation because there are local optima—assuming that agents only make individually rational contracts. Three new contract types were presented to overcome this problem: clustering, swaps, and multiagent contracts. These are insufficient when applied alone, in pairs, or even when all three are used separately. On the other hand, these three types can be combined into a new atomic contract type: CSM-contract. Theorem 3.14 showed that if the agents can pick any CSM-contracts, any hill-climbing algorithm finds the global task allocation optimum in a finite number of steps without backtracking. This means that from the perspectives of social welfare maximization and of individual rationality (not necessarily from the perspective of local payoff maximization), agents can accept individually rational contracts as they are offered. They need not wait for more profitable ones, and the need not worry that a current contract may make a more profitable future contract unprofitable. Neither do they need to accept contracts that are not individually rational in anticipation of future contracts that make the combination beneficial. This result is powerful for small problem instances, but for large instances, the number of contracts made before the globally optimal task allocation is reached may be too great to be carried out in practice—albeit finite. Therefore, for the larger problem instances, the guaranteed anytime property is more important.

Section 3.6.1 discussed the implications of agents' computational limitations on contracting. If an agent's computation is bounded and its (real or negotiation) environment is dynamic, or an agents computation is costly, the agent has to explicitly trade off computation quality against the cost that stems from computation. Several real-time pressures were identified that are specific to contracting settings. It was argued that an agent therefore needs to decide which task sets to focus its deliberation on, how much, and in what order. Next, the issue of engaging in multiple negotiations simultaneously was discussed. If an agent bids or awards while some of its older bids are pending, the current bid or award may turn out unbeneficial in the world that materializes by some combination of acceptances of pending bids. Marginal cost calculations were formalized that pessimistically assume that the worst combination gets accepted. With this type of calculation, an agent's local cost is guaranteed to decrease monotonically. An opportunistic approximation was also presented that assumes that none of the pending bids get accepted—but a feasibility check was used to ensure that a feasible solution can be reached locally even if all pending bids get accepted. This approximation is faster to compute because the exponentially many alternative worlds need not be considered, but an agent may occasionally make unbeneficial contracts. So, there is a tradeoff between computational complexity and monetary risk. A marginal cost approximation was also formalized that assumes that all of the pending bids get accepted. The complexity of this is between those of the other two methods.

Alternatively, an agent can refuse to bid or award while previous bids are pending. In this case, the marginal cost can be calculated risk free without considering alternative future worlds. On the other hand, such an agent cannot participate in simultaneous negotiations, and may lose opportunities. On the other hand, this types of agents can have each negotiation take less time because local deliberation is faster. The distributed vehicle routing experiments showed that bidding and awarding while previous bids pend (using the opportunistic pricing) indeed leads to some unbeneficial contracts in practice. On the other hand, the experiments confirm that bidding and awarding only when previous bids do not pend leads to monotonic decrease of the local cost. Which of these methods is best in terms of decreasing local cost most depends on the problem, the problem instances, the set duration of the valid bidding window, and even the interleaving of the asynchronous messages that happens to occur. The experiments were inconclusive regarding this question. Section 3.6.3 concluded the discussion on contracting implications of limited computation by presenting tradeoffs that an agent faces in deciding whether to bid or award early on, or to wait.

Section 3.7 discussed distributed asynchronous implementation of automated contracting. First, message congestion and agent saturation were addressed. The classical solution approaches, focused addressing and audience restrictions, are not viable among self-interested agents: there is an inherent tragedy of the commons because each agent is self-interested and does not care of the saturation of others. Three protocol related solutions were discussed: use-based communication charges, mutual monitoring, and processing fees. However, the problem can be solved without these by incorporating certain policies into each agent's local strategy. These policies are in concert with each agent's self-interest: they need not be externally imposed. The local strategies that completely solved the congestion and saturation problems in our experiments included reading incoming messages in batches and ignoring announcements that were older than a time limit.

Section 3.7.2 presented a negotiation termination method for iterative task reallocation contracting. The termination problem has been considered difficult in previous work. The presented method terminates the negotiations when a local optimum has been reached with respect to a—possibly restricted—set of clustering contracts. It does not assume common knowledge of tasks. The method could be extended to swaps and multiagent contracts. On the other hand, the method is not guaranteed to work among agents that are not cooperative or agents that cannot exactly compute marginal costs. Finally, Section 3.7.3 discussed the role and feasibility of replies and timeouts in implementing automated contracting.

CHAPTER 4

COALITION FORMATION

Self-interested real world parties can often save costs by coordinating their activities with other parties. One can view the situation as agents coordinating their actions (computational and real world) within each coalition but not across coalitions. The motivation for coalition formation among agents, and related research were thoroughly discussed in Section 2.1, and will not be repeated here. This chapter only discusses the new contributions. The treatment assumes familiarity with the concepts presented in Sections 2.1.1 and 2.1.2.1. Coalition formation includes three activities as discussed in Section 2.1.1: coalition structure generation, solving of the optimization problem of each coalition, and dividing the value (or cost) of the coalition among member agents.

In the contracting setting of Chapter 3, these activities are intermixed conceptually and temporally. Agents can contract with any other agent, so in this sense all of the agents work in one large coalition. Second, the solution of the joint task allocation problem is done iteratively while each agent maintains a local solution. Third, payoff distribution occurs at each iteration of the task allocation among agents.

On the other hand, this chapter analyzes these three activities as conceptually separate. The state of the art in coalition formation is extended by incorporating an explicit quantitative and normative model of computational limitations.

4.1 Computation Unit Cost and Algorithm as Rationality Limits

If the problem is hard and the instance is large, it is unrealistic to assume that it can be solved without deliberation costs. This chapter of the dissertation adopts a model of bounded rationality, where each agent has to pay for the computational resources (CPU cycles) that it uses for deliberation. A fixed computation cost $c_{comp} \geq 0$ per CPU time unit is assumed.[1] The domain cost associated with coalition S is denoted by $c_S(r_S) \geq 0$, i.e. it depends on (decreases with) the allocated computation resources r_S, Fig. 4.1. For example in the vehicle routing problem, the domain cost is the sum of the lengths of the vehicles' routes. The functions $c_S(r_S)$ can be viewed as performance profiles of the problem solving algorithm. They are used to decide how much CPU time to allocate to each computation. With this model of bounded rationality, the value of a coalition with bounded rational (BR) agents can be defined. Each coalition minimizes the sum of solution cost (i.e. domain cost) and computation cost:

$$v_S(c_{comp}) = -\min_{r_S}[c_S(r_S) + c_{comp} \cdot r_S].^2 \tag{4.1}$$

The coalition value decreases as the CPU time unit cost c_{comp} increases, Fig. 4.1. Our model also incorporates a second form of bounded rationality: the base algorithm may be incomplete, i.e. it might never find the optimal solution. If it is complete, the BR value of a coalition when $c_{comp} = 0$ equals the rational value ($v_S(0) = v_S^R$). In all, the bounded rational value of a coalition is determined by three factors:

- The *domain problem*: tasks and resources of the agents. Among rational agents this is the only determining factor.
- The *execution architecture* on which the problem solving algorithm is run. Specifically, the architecture determines the unit cost of computation, c_{comp}.

[1] In practice, CPU time can already be bought, e.g. on supercomputers. Similarly, the developing infrastructure for remotely executing agents provides an equivalent setting. For example in Telescript [General Magic, Inc., 1994], the remotely executing agents pay Teleclicks for CPU time to the owner of the host machine. In this dissertation, the market for CPU time is assumed to be so large that the demand of the agents we are studying does not impact the price of a CPU time unit. It is also assumed that this price is common to all agents, which corresponds to an open CPU cycle market.

[2] Throughout this chapter on coalition formation, *min*-operators are used due to their familiarity, although strictly speaking the value of such a *min*-operator may be undefined because $c_S(r_S)$ need not be continuous. Thus, to be precise, *inf*-operators should be used.

- The *problem solving algorithm.* We make no restrictive assumptions as to how effectively the algorithm uses the execution architecture. This is realistic because in practise it is often hard to construct algorithms that optimally (in some sense) use the architecture.

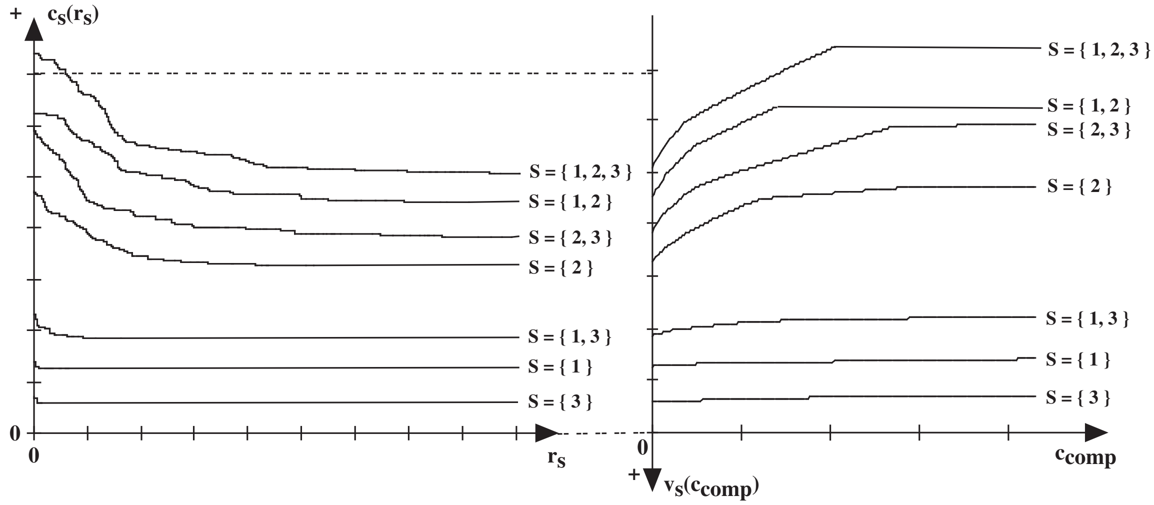

Figure 4.1. Example experiment from the vehicle routing domain with agents 1, 2, and 3. Left: performance profiles, i.e. solution cost as a function of allocated computation resources. The curves become flat when the algorithm has reached a local optimum. Right: BR coalition value as a function of computation unit cost. The value of each coalition is negative because the cost is positive. The curves become flat at a c_{comp} that is so high that it is not worthwhile to take any iterative refinement steps: the initial solutions are used (their computation requirements are assumed negligible).

Conceptually the agents use *design-to-time algorithms* [Garvey and Lesser, 1993, Zilberstein, 1993, Garvey and Lesser, 1994]: once an agent has decided how much CPU time r_S it will allocate to a computation, it can design an algorithm that will find a solution of cost $c_S(r_S)$. The design-to-time framework is used instead of the *anytime* framework [Sandholm and Lesser, 1994, Dean and Boddy, 1988, Boddy and Dean, 1989, Horvitz, 1987, Zilberstein, 1993] because to devise a theory of self-interested agents, the possibility that they design their algorithms to time has to be accounted for. With deterministic performance profiles, for any desired computation time allocation or solution quality, a noninterruptible design-to-time algorithm can be constructed that performs no worse than an interruptible anytime algorithm. We assume that the performance profiles exactly predict the solution cost attained for a given CPU time allocation. So, we have relaxed the assumption that the base level algorithm is optimal (complete and costless), but instead we assume that the meta-level deliberation controller is optimal (exact and costless). Assuming optimality of the meta-level is more realistic than assuming optimality of the base level, but it still does not match reality exactly. In practice there is uncertainty in each performance profile: the meta-level is not exact.[3] Secondly, the performance profile depends on several features of the problem instance, and computing the mapping from the instance to the performance profile [Sandholm and Lesser, 1994] may take considerable time, thus making the meta-level itself costly. In the limit, the base algorithm would be run at the meta-level to determine what it would achieve for a given time setting. Assuming an optimal meta-level enables analyzing bounded rationality at the base level in isolation from uncertainty of the performance profiles. It also allows us to sidestep the problem of having a meta-meta-level controlling the meta-level, a meta-meta-meta-level controlling the meta-meta-level, and so on *ad infinitum*.

In this coalition formation chapter of the dissertation we assume that the problem instances (tasks and resources) of all agents are common knowledge. This is somewhat unrealistic in open environments with a large number of agents. In practice it is often necessary to learn the other agents' characteristics from previous encounters. Alternatively, the agents can be made to explicitly declare their tasks and resources, but they may lie in order to gain monetarily. Rosenschein and Zlotkin [Rosenschein

[3]If the performance profiles are only probabilistically known, anytime algorithms may be desirable due to their flexibility with respect to termination time. In general, for optimal meta-reasoning, the remaining part of a probabilistic performance profile should be conditioned on the algorithm's performance on that problem instance on previous CPU time steps [Sandholm and Lesser, 1994, Zilberstein, 1993].

and Zlotkin, 1994] analyze when rational agents are motivated to declare truthfully. Unfortunately that work assumes only two agents and that they can optimally solve exponentially many $\mathcal{NP}$-complete problems without computation costs. Even under these assumptions, in most cases, truth-telling is not achieved (see Section 2.2.4). The effect of bounded rationality on truthful revelation is unknown.

For now—this is relaxed in Section 4.5—we assume that the agents solve the combinatorial optimization problems equally well and that this is common knowledge. For any coalition's problem and for any setting of CPU time, the cost of the solution potentially generated by each agent is the same. The agents need not generate the same solutions, only the same quality.

With such shared deterministic performance profiles, each agent knows the value $v_S(c_{comp})$ of each potential coalition S up front. Therefore coalition formation will take place before any computation. After collusion, each coalition computes its solution using the optimal amount of CPU time r_S as defined by Equation 4.1. Because in our model, rationality is bounded by CPU time cost, it costs the same for one agent to use nt CPU time units as it costs n agents to use t units. Therefore, it is best if a coalition's optimization problem is solved by a single agent. This is trivially true since an agent could simulate distributed problem solving among n agents for time t by using a local algorithm for nt. Conversely, it is not always possible (due to redundancy etc.) for n agents solving the problem for time t to reach a solution of the same quality as one agent using nt can reach. The computing agent can be arbitrarily chosen from within the coalition, and the coalition pays that agent its true cost for computing. This cost along with the domain solution cost contribute to $v_S(c_{comp})$, which is divided among the agents in the coalition as will be presented later.

4.2 Social Welfare Maximizing Coalition Structure

Any outcome of a game can be analyzed with respect to *social welfare*, which is defined as the sum of the agents' payoffs (Section 1.2.1). The payoff that agent i

gets is called $x_i \in \Re$. The sum of the agents' x_i's has to equal the sum of the values of the coalitions in the coalition structure (CS) that formed: no wealth is generated from nothing and no wealth disappears. With bounded rational (BR) agents, these coalition values incorporate the computation costs.

The concepts of superadditivity and subadditivity were discussed in Section 2.1 along with the implications on social welfare maximizing coalition structure. Now we present a new concept for BR agents that is analogous to superadditivity among rational agents. A game is *bounded rational superadditive* (BRSUP) if the best value that one coalition can reach given the computation cost plus the best value that another coalition can reach given the computation cost is never greater than the best value that these coalition can reach as a composite coalition given the computation cost:

Definition 4.1 *A game is* **bounded rational superadditive (BRSUP)** *for computation unit cost* c_{comp} *if* $(\forall S, T \subseteq A, S \cap T = \emptyset), v_{S \cup T}(c_{comp}) \geq v_S(c_{comp}) + v_T(c_{comp})$.

Every BRSUP game is a *bounded rational grand coalition game*, Fig. 4.2. In such games, BR agents are best off—from a social welfare viewpoint—by all working together, i.e. by forming the grand coalition ($CS^* = \{A\}$). There also exist BR grand coalition games which are not BRSUP: the grand coalition may be the optimal coalition structure although not all local poolings are beneficial.

BR superadditivity does not always coincide with superadditivity. In general, for a given c_{comp}, a game can be superadditive, BRSUP, both, or neither, Fig. 4.2.

Only some non-BRSUP games are BR subadditive, Fig. 4.2:

Definition 4.2 *A game is* **bounded rational subadditive (BRSUB)** *for computation unit cost* c_{comp} *if* $(\forall S, T \subseteq A, S \cap T = \emptyset), v_{S \cup T}(c_{comp}) < v_S(c_{comp}) + v_T(c_{comp})$.

If the game is BR subadditive, agents are best off alone, i.e. by colluding with nobody ($CS^* = \{\{a_1\}, \{a_2\}, ..., \{a_{|A|}\}\}$). In games that are neither BRSUP nor BRSUB, the

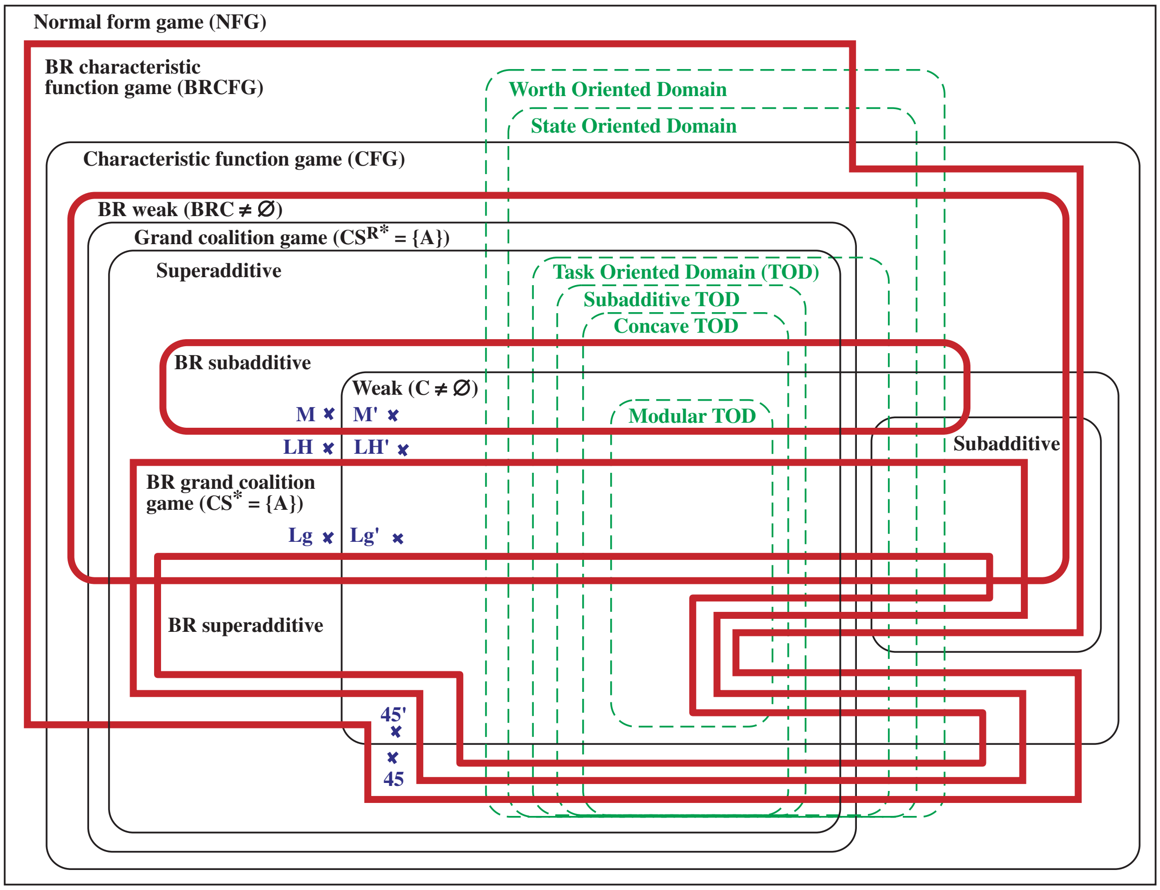

Figure 4.2. Venn diagram of negotiation domains. Normal lines show the classification for rational agents. Bold lines show our new classification for BR agents. Dotted lines show the rational agent classification of Rosenschein and Zlotkin (Section 2.2.4). They use "Subadditive" to mean that an agent's cost for handling tasks is subadditive in tasks. We use subadditive to refer to coalition value functions that are subadditive in agents. The figure does not reflect the fact that they do not allow side payments.

optimal coalition structure varies, and several coalition structures may be equally good with respect to social welfare. We will denote any one of these best coalition structures by CS^*.

The rest of this section analyzes the relationship between the shape of the performance profiles and the class of the game. Specifically, the question is: what types of performance profiles make a game BRSUP (or BRSUB) for all computation unit costs? If the agents have these types of performance profiles, they know the optimal coalition structure irrespective of which execution platform they are running on.

BR superadditivity depends on the performance profiles and the unit cost of computation. The next theorem states a natural condition on the performance profiles. If the condition holds, the game is BRSUP for any c_{comp}.

Theorem 4.1 BRSUP (sufficient condition). $[(\forall S, T \subseteq A, S \cap T = \emptyset, \forall r_S \geq 0, \forall r_T \geq 0), c_{S \cup T}(r_S + r_T) \leq c_S(r_S) + c_T(r_T)] \Rightarrow$ *Game is BRSUP for all* c_{comp}.

The condition states that the domain cost for coalition S after allocating a certain amount r_S of computation plus the domain cost to another coalition T after allocating a certain amount r_T of computation is never less than the domain cost of these coalitions combined after allocating $r_S + r_T$. This is always achievable in theory because in the worst case, the algorithm can allocate r_S on the problem of S and then do the problem of T using r_T separately. Given a large coalition, it is difficult to intelligently guess an efficient decomposition of this type. To be sure of BR superadditivity, the algorithm would need to solve each agent's problem separately—thus ensuring superadditivity trivially by additivity.

Usually, the algorithm that is used on the composite problem does not apply this type of problem decomposition. The real desideratum is not necessarily to generate algorithms that guarantee BR superadditivity (and thus the superiority of the grand coalition over other coalition structures), but algorithms that provide the highest social welfare (for the best coalition structure, which need not be the grand coalition). Sometimes these goals are conflicting. Whether the algorithm's performance profiles actually satisfy the conditions for BR superadditivity without using a decomposition method depends on the problem, the specific instances under study, and the algorithm itself.

In general, the game can be BRSUP for all c_{comp} even if the above condition does not hold on the performance profiles:

Theorem 4.2 $[(\forall S, T \subseteq A, S \cap T = \emptyset, \forall r_S \geq 0, \forall r_T \geq 0), c_{S \cup T}(r_S + r_T) \leq c_S(r_S) + c_T(r_T)] \not\Leftarrow$ *Game is BRSUP for all* c_{comp}.

It is reasonable to assume that the performance profile $c_S(r)$ is decreasing in r if the agent can inexpensively store the best solution it has arrived at so far. Furthermore, $c_S(r)$ is often convex in r: greater savings are achieved in the early stages of computation and the savings per time unit decrease as problem solving proceeds. We conjecture that performance profiles of design-to-time algorithms are almost always convex. On the other hand, performance profiles of anytime algorithms are typically not convex at points where the base algorithm switches from one approach to another. One example is completing an iterative refinement algorithm by running an exhaustive complete algorithm after the refinement phase. Another example is switching from using one refinement operator (e.g. 2-swap in TSP [Lin and Kernighan, 1971, Sandholm, 1993]) to using another refinement operator (e.g. 3-swap in TSP). Furthermore, refinements often decrease solution cost in a step-wise, noncontinuous manner rendering the performance profiles locally nonconvex—as in our experiments (Fig. 4.1 left). If the algorithm is stochastic, these step-related nonconvexities are reduced as the performance profile is averaged over multiple runs. The performance profiles in our experiments exhibited an overall convex nature, but also had true local nonconvexities (because the design-to-time algorithms were constructed from anytime algorithms, and were not tailored for each time setting separately, Sec. 4.4). Convexity is significant because with convex performance profiles, a domain is BRSUP for all computation unit costs if *and only if* the condition of Theorem 4.1 on the performance profiles holds:

Theorem 4.3 BRSUP (necessary and sufficient condition). *Let us restrict ourselves to such performance profiles that* $\forall U \subseteq A, c_U(r)$ *is* **decreasing** *and* **convex** *in* r. *Now,* $[(\forall S, T \subseteq A, S \cap T = \emptyset, \forall r_S \geq 0, \forall r_T \geq 0), c_{S \cup T}(r_S + r_T) \leq c_S(r_S) + c_T(r_T)] \Leftrightarrow$ *Game is BRSUP for all* c_{comp}.

Analogous to Theorem 4.1, there is an easy sufficient condition on the performance profiles that guarantees that the game is BR subadditive for all computation unit costs:

Theorem 4.4 BRSUB (sufficient condition). $[(\forall S, T \subseteq A, S \cap T = \emptyset, \forall r_S \geq 0, \forall r_T \geq 0), c_{S \cup T}(r_S + r_T) > c_S(r_S) + c_T(r_T)] \Rightarrow$ *Game is BRSUB for all* c_{comp}.

4.3 Stability of the Coalition Structure

In the previous section we presented conditions on the performance profiles that describe what coalition structure the agents are best off forming from the social welfare viewpoint. In this section we analyze the *stability* of that coalition structure. Can the social good be distributed among the agents so that each agent is motivated to stay with the social welfare maximizing coalition structure (individual rationality)? Furthermore, can it be distributed so that every subgroup of agents is motivated to stay (coalition rationality)? As discussed in Section 2.1.2.1, the *core* is the solution concept that satisfies both of these conditions.

Now we introduce the analog of the core for BR agents.

Definition 4.3 *The* **bounded rational core (BRC)** *for computation unit cost* c_{comp} *is* $BRC(c_{comp}) = \{(\vec{x}, CS^*) | \forall S \subset A, \sum_{i \in S} x_i \geq v_S(c_{comp})$ *and* $\sum_{i \in A} x_i = \sum_{j \in CS^*} v_{S_j}(c_{comp})\}$.

If the BRC is not empty, BR agents can divide the social good among themselves in a way that no subgroup is motivated to break away from CS^*. Sometimes the BRC is empty, but this does not always coincide with the core being empty. There are games, where the BRC and the core exist, games where either one of them exists separately, and games where both are empty, Fig. 4.2.

If the agents are best off working separately, the coalition structure with separate agents is stable, Fig. 4.2:

Theorem 4.5 BRC in BRSUB games (necessary and sufficient condition). *Game is BRSUB for some* $c_{comp} \Rightarrow BRC(c_{comp}) \neq \emptyset$.

In domains that are not BR subadditive, the BRC is sometimes empty. The condition $C \neq \emptyset$ can be converted into necessary and sufficient conditions on the v_S^R's in games where the grand coalition maximizes social welfare [Shapley, 1967, Charnes and Kortanek, 1966]. We convert the condition $BRC(c_{comp}) \neq \emptyset$ into conditions on the $v_S(c_{comp})$'s analogously. Let $B_1, ..., B_p$ be distinct, nonempty, proper subsets of A. The set $\mathcal{B} = \{B_1, ..., B_p\}$ is called *balanced* if there are positive coefficients $\lambda_1, ..., \lambda_p$ such that $\forall i \in A, \sum_{\{j|i \in B_j\}} \lambda_j = 1$. A *minimal balanced set* includes no other balanced sets.

Theorem 4.6 BRC in grand coalition games (necessary and sufficient condition). *In games where $CS^* = \{A\}$ for some c_{comp}, $BRC(c_{comp}) \neq \emptyset$ iff for every minimal balanced set $\mathcal{B} = \{B_1, ..., B_p\}, \sum_{j=1}^{p} \lambda_j v_{B_j}(c_{comp}) \leq v_A(c_{comp})$.*

Example. In any 3-agent game where $CS^* = \{A\}$ for some c_{comp}, $BRC(c_{comp}) \neq \emptyset$ iff $v_{\{1\}}(c_{comp}) + v_{\{2,3\}}(c_{comp}) \leq v_{\{1,2,3\}}(c_{comp})$ and $v_{\{2\}}(c_{comp}) + v_{\{1,3\}}(c_{comp}) \leq v_{\{1,2,3\}}(c_{comp})$ and $v_{\{3\}}(c_{comp}) + v_{\{1,2\}}(c_{comp}) \leq v_{\{1,2,3\}}(c_{comp})$ and $v_{\{1\}}(c_{comp}) + v_{\{2\}}(c_{comp}) + v_{\{3\}}(c_{comp}) \leq v_{\{1,2,3\}}(c_{comp})$ and $\frac{1}{2}v_{\{1,2\}}(c_{comp}) + \frac{1}{2}v_{\{1,3\}}(c_{comp}) + \frac{1}{2}v_{\{2,3\}}(c_{comp}) \leq v_{\{1,2,3\}}(c_{comp})$. All but the last inequality are implied by the fact that $CS^* = \{A\}$.

Example. In any 4-agent game where $CS^* = \{A\}$ for some c_{comp}, $BRC(c_{comp}) \neq \emptyset$ iff the 41 inequalities of Table 4.1 hold. Constraints 1, 2, 3 and 5 correspond to partitions of A (all λ's are 1). They are thus implied by the fact that $CS^* = \{A\}$.

In BRSUP games, a subset of the above inequalities suffices. Let us call a minimal balanced set *proper* if no two of its elements are disjoint.

Theorem 4.7 BRC in BRSUP games (necessary and sufficient condition). *In a game that is BRSUP for some c_{comp}, $BRC(c_{comp}) \neq \emptyset$ iff for every proper minimal balanced set $\mathcal{B} = \{B_1, ..., B_p\}$, $\sum_{j=1}^{p} \lambda_j v_{B_j}(c_{comp}) \leq v_A(c_{comp})$. Furthermore, this set of inequalities is minimal: no smaller set is sufficient.*

Example. In a 3-agent game that is BRSUP for some c_{comp}, $BRC(c_{comp}) \neq \emptyset$ iff $\frac{1}{2}v_{S_{\{1,2\}}}(c_{comp}) + \frac{1}{2}v_{S_{\{1,3\}}}(c_{comp}) + \frac{1}{2}v_{S_{\{2,3\}}}(c_{comp}) \leq v_{S_{\{1,2,3\}}}(c_{comp})$.

Table 4.1. Conditions for existence of the BRC in a 4-agent grand coalition game. Last column shows the number of constraints generated from that constraint by permuting the agents (including the presented permutation).

Id	Constraint	#
1	$v_{\{1,2\}}(c_{comp}) + v_{\{3,4\}}(c_{comp}) \leq v_{\{1,2,3,4\}}(c_{comp})$	3
2	$v_{\{1,2,3\}}(c_{comp}) + v_{\{4\}}(c_{comp}) \leq v_{\{1,2,3,4\}}(c_{comp})$	4
3	$v_{\{1,2\}}(c_{comp}) + v_{\{3\}}(c_{comp}) + v_{\{4\}}(c_{comp}) \leq v_{\{1,2,3,4\}}(c_{comp})$	6
4	$\frac{1}{2}v_{\{1,2,3\}}(c_{comp}) + \frac{1}{2}v_{\{1,2,4\}}(c_{comp}) + \frac{1}{2}v_{\{3,4\}}(c_{comp}) \leq v_{\{1,2,3,4\}}(c_{comp})$	6
5	$v_{\{1\}}(c_{comp}) + v_{\{2\}}(c_{comp}) + v_{\{3\}}(c_{comp}) + v_{\{4\}}(c_{comp}) \leq v_{\{1,2,3,4\}}(c_{comp})$	1
6	$\frac{1}{2}v_{\{1,2\}}(c_{comp}) + \frac{1}{2}v_{\{1,3\}}(c_{comp}) + \frac{1}{2}v_{\{2,3\}}(c_{comp})$ $+v_{\{4\}}(c_{comp}) \leq v_{\{1,2,3,4\}}(c_{comp})$	4
7	$\frac{1}{2}v_{\{1,2,3\}}(c_{comp}) + \frac{1}{2}v_{\{1,4\}}(c_{comp}) + \frac{1}{2}v_{\{2,4\}}(c_{comp})$ $+\frac{1}{2}v_{\{3\}}(c_{comp}) \leq v_{\{1,2,3,4\}}(c_{comp})$	12
8	$\frac{2}{3}v_{\{1,2,3\}}(c_{comp}) + \frac{1}{3}v_{\{1,4\}}(c_{comp}) + \frac{1}{3}v_{\{2,4\}}(c_{comp})$ $+\frac{1}{3}v_{\{3,4\}}(c_{comp}) \leq v_{\{1,2,3,4\}}(c_{comp})$	4
9	$\frac{1}{3}v_{\{1,2,3\}}(c_{comp}) + \frac{1}{3}v_{\{1,2,4\}}(c_{comp}) + \frac{1}{3}v_{\{1,3,4\}}(c_{comp})$ $+\frac{1}{3}v_{\{2,3,4\}}(c_{comp}) \leq v_{\{1,2,3,4\}}(c_{comp})$	1

Example. In a 4-agent game that is BRSUP for some c_{comp}, $BRC(c_{comp}) \neq \emptyset$ iff the 11 conditions acquired from Table 4.1's constraints 4, 8 and 9 are satisfied.

Next we present conditions on the performance profiles that are *sufficient* to guarantee that the BRC exists. According to Theorem 4.5, the conditions on the performance profiles that guarantee BR subadditivity (Theorem 4.4) form one such set of conditions. The following set suffices for games where $CS^* = \{A\}$:

Theorem 4.8 BRC in grand coalition games (sufficient condition). *In games where $CS^* = \{A\}$ for some c_{comp}, [for every minimal balanced set $\mathcal{B} = \{B_1, ..., B_p\}$, $(\forall B \in \mathcal{B}, \forall r_B \geq 0)$ $\sum_{j=1}^{p} \lambda_j c_{B_j}(r_{B_j}) \geq c_A(\sum_{j=1}^{p} \lambda_j r_{B_j})] \Rightarrow BRC(c_{comp}) \neq \emptyset$.*

If $CS^* = \{A\}$ for *all* $c_{comp}(\geq 0)$, the above conditions guarantee existence of the $BRC(c_{comp})$ for *all* $c_{comp}(\geq 0)$. This would mean coalition stability for any execution platform. In BRSUP games, fewer conditions suffice:

Theorem 4.9 BRC in BRSUP games (sufficient condition). *In a game that is BRSUP for some* $c_{comp} \geq 0$, *[for every proper minimal balanced set* $\mathcal{B} = \{B_1, ..., B_p\}$, $(\forall B \in \mathcal{B}, \forall r_B \geq 0)\ \sum_{j=1}^{p} \lambda_j c_{B_j}(r_{B_j}) \geq c_A(\sum_{j=1}^{p} \lambda_j r_{B_j})] \Rightarrow BRC(c_{comp}) \neq \emptyset$.

Again, if the game is BRSUP for *all* $c_{comp}(\geq 0)$, the above conditions guarantee existence of the $BRC(c_{comp})$ for *all* $c_{comp}(\geq 0)$.

Example. In a 3-agent game that is BRSUP $\forall c_{comp}$, $[(\forall r_{\{1,2\}} \geq 0, \forall r_{\{1,3\}} \geq 0, \forall r_{\{2,3\}} \geq 0), \frac{1}{2}c_{\{1,2\}}(r_{\{1,2\}}) + \frac{1}{2}c_{\{1,3\}}(r_{\{1,3\}}) + \frac{1}{2}c_{\{2,3\}}(r_{\{2,3\}}) \geq c_{\{1,2,3\}}(\frac{1}{2}r_{\{1,2\}} + \frac{1}{2}r_{\{1,3\}} + \frac{1}{2}r_{\{2,3\}})] \Rightarrow \forall c_{comp}, BRC(c_{comp}) \neq \emptyset$.

4.4 Experimental Results: Distributed Vehicle Routing

From the theoretical arguments it is known that problems and problem instances can be generated to populate any region of the Venn diagram of coalition games (Figure 4.2). However, this does not mean that real-world problems and problem instances uniformly populate this space. The role of the experiments of this section is to analyze where a particular real world problem and its instances fall in the space of coalition games. Some quite surprising results appeared.

Coalition formation among bounded rational agents was tested on the vehicle routing problem using one week real-world vehicle and order data from five geographically distributed dispatch centers. The problem and the large instances were described in Section 1.1.1, where it was also shown that the problem is $\mathcal{NP}$-complete.

The domain cost $c_S(r_S)$ for a coalition S was the sum of the route lengths of the vehicles of that coalition (while handling all of its orders) in the solution that had been reached after computation r_S. So, the rational value (v_S^R) of each coalition is defined by the tasks (delivery orders) and the resources (vehicles, depots) of the agents in the coalition. The problem instances in our example are so large that even the smallest ones are too hard to solve optimally. Therefore, rational coalition formation algorithms for vehicle routing problems [Lundgren *et al.*, 1992] are unusable in this case.

Our problem is outside the domain classification of Rosenschein and Zlotkin [Rosenschein and Zlotkin, 1994], Fig. 4.2, because agents do not have symmetric capabilities due to heterogeneous fleets. If their definition were extended to allow asymmetric capabilities, our domain would be in SOD $\setminus$ TOD. Our domain would not be a TOD because any one agent is not necessarily able to individually handle all tasks of all agents. If we further dropped the maximum route length constraint (this experiment will also be presented), and restricted ourselves to domains where each center has at least one sufficient vehicle to satisfy the weight/volume constraints of any order of any center (not true in our data), then the domain would be a TOD. The following simple example shows that it would not be a "Subadditive TOD" because the depots are geographically distributed. Let us look at a game with just two agents (A1 and A2), two delivery tasks (T1 and T2), and two identical vehicles—one for each agent. Say that the pickup site and the drop-off site of T1 are close to A1's depot, and T2's pickup and drop-off are close to A2's depot. Now say that the depots are far from each other. Thus the sum of the route lengths when A1 manages T1 and A2 manages T2 is lower than when either agent individually manages both tasks.

To analyze a game we ran the same algorithm on the vehicle routing problem of each subgroup of agents separately and thus acquired a performance profile for each potential coalition. The algorithm first generates an initial solution by giving each vehicle one long delivery and then, in order, giving each vehicle the delivery that can be added to its route with the least cost without violating the constraints. The second phase of the algorithm is based on iterative refinement. At each step, a delivery (chosen from a randomly ordered circular list) is removed from the routing solution and inserted back to the solution, but into the least expensive place while not violating the constraints. The drop-off location of the delivery has to be inserted after the pickup location into the same vehicle's route, but not necessarily into the same leg. We ran the refinement algorithm until no remove-insert operation enhanced the solution: a local optimum was reached. In the performance profiles we ignored the time to construct the initial solution, and only viewed how the solution cost decreased with more CPU seconds of iterative refinement, Fig. 4.1 left. The refinement algorithm is an anytime algorithm, but because the performance profiles are exact (as explained, they are precomputed for experimental purposes by running the base algorithm itself), the agents do not gain information from execution on that instance so far. Therefore the algorithm is equivalent to a design-to-time algorithm for our purposes.

We analyzed all of the $\binom{5}{3} = 10$ 3-agent games that can be acquired by choosing 3 of the 5 dispatch centers. There are 7 subgroups of the 3 agents: $\{1\}$, $\{2\}$, $\{3\}$, $\{1,2\}$, $\{2,3\}$, $\{3,1\}$, $\{1,2,3\}$ and 5 coalition structures: $\{\{1\}, \{2\}, \{3\}\}$, $\{\{1\}, \{2,3\}\}$, $\{\{2\}, \{1,3\}\}$, $\{\{3\}, \{1,2\}\}$, $\{\{1,2,3\}\}$. Figure 4.1 shows the performance profiles with agents 1, 2 and 3.

Each of our games is superadditive for rational agents because at worst, a composite coalition can use the solutions of separate coalitions. In general, a game can be non-superadditive only if the collusion process itself involves some cost, e.g. anti-trust penalties. Thus rational agents would be best off by forming the grand coalition. Surprisingly, none of the games were BRSUP for any c_{comp}, Fig. 4.3. For c_{comp} in the mid-range, the 3-agent games were often BRSUB (point M in Fig. 4.2), while in the low and high ranges (point LH in Fig. 4.2), they were often neither BRSUP nor BRSUB. In some of these mixed games, for low c_{comp}, the grand coalition was the best coalition structure (point Lg in Fig. 4.2). Existence of the core for rational agents is unknown for our games: the points M, LH, and Lg might really be M', LH', and Lg'. The BRC was non-empty in all 3-agent games for all values of c_{comp}. To summarize, rational agents would be best off forming the possibly unstable grand coalition, while

BR agents should form varying coalition structures (the grand coalition for some low values of c_{comp}), which are always stable. We also reran the experiments without the maximum route length restriction, and these results prevailed, Fig. 4.3.

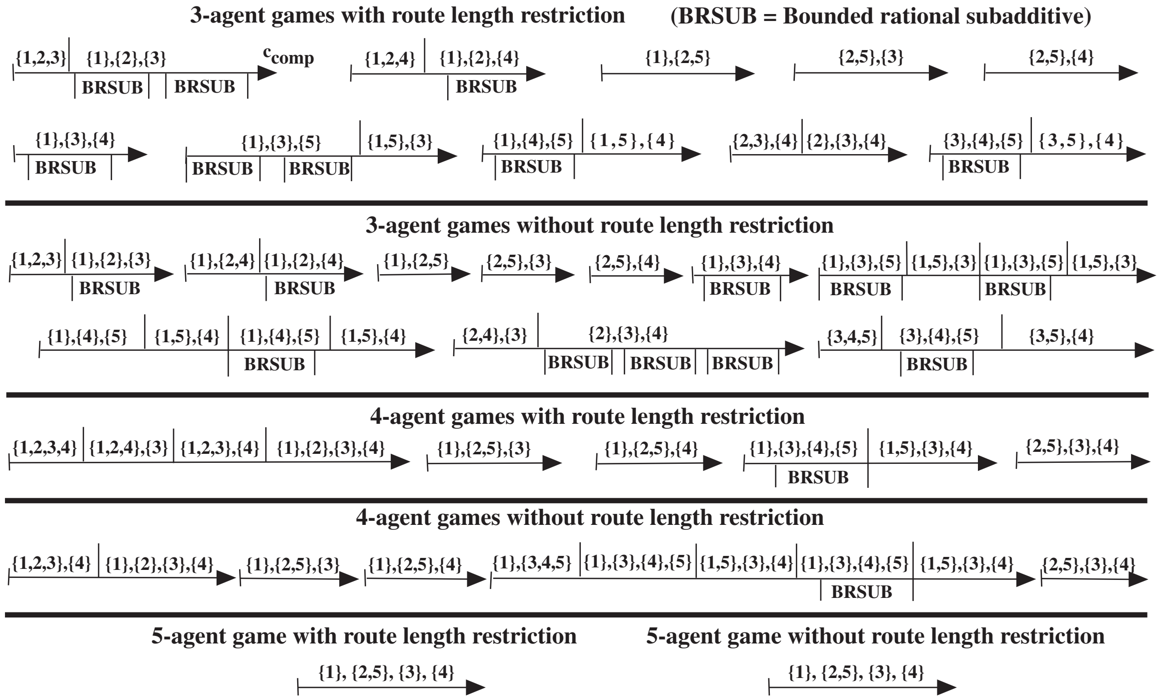

Figure 4.3. Optimal coalition structure (CS^*) and bounded rational subadditivity as a function of c_{comp}. Tested by evaluating all possible coalition structures and super/subadditivity at varying points of c_{comp} chosen from a grid where c_{comp} is always incremented by 1%.

Centers 2, 3 and 5 were located near each other, while 1 and 4 were far from each other and the other centers. Centers 1, 3, 4 and 5 transported heavy low volume items, while 2 transported light voluminous items. Centers 1..5 had 65, 200, 82, 124, and 300 deliveries, and 10, 13, 21, 18, and 15 vehicles respectively. Both with and without the route length restriction, 2 and 5 were best off by only mutually colluding for any c_{comp}. Their deliveries have considerable areal overlap due to adjacency, and the light voluminous items and heavy low volume items can be profitably joined into the weight and volume constrained vehicles. Centers 2 and 3 did not collude as much as 2 and 5 because 3's vehicles had tighter volume constraints than 5's—hindering the transport of 2's goods. No other two centers besides 2 and 5 were always best off in a 2-agent coalition independent of the third agent of the game. Relaxing the route length constraint increased collusion between the distant 2 and 4 while demoting collusion of the adjacent 2 and 3.

Next we analyzed the $\binom{5}{4} = 5$ 4-agent games and the 5-agent game with and without the route length restriction. In every game, the existence of $BRC(c_{comp})$ varied many times as a function of c_{comp}, but it existed for the largest values of c_{comp}. No game was BRSUP for any c_{comp}, but some games were BRSUB for values in the medium range, Fig. 4.3. The grand coalition was the best coalition structure in only one of the twelve games with four and five agents. This happened for low c_{comp} in the game with agents 1, 2, 3, and 4, and the route length restriction. When this occurred, $BRC(c_{comp})$ happened to be non-empty (point Lg (or Lg') in Fig. 4.2). In all of the three, four, and five agent games, $BRC(c_{comp})$ was always nonempty when the best coalition structure was the grand coalition. Thus, depending on c_{comp}, the games were at the points M, LH, Lg, or 45 (or M', LH', Lg', or 45') in Figure 4.2. The best coalition structure varied despite the fact that rational agents would be best off forming the grand coalition due to superadditivity. Again, whenever both agents 2 and 5 participated, they were best off by mutually colluding for all computation unit costs. In those games no other agents colluded.

Put together, the main surprising result is that although rational agents should always form the grand coalition, this is all but obvious among bounded rational agents. None of the vehicle routing games of our experiments—using real data and a reasonable iterative refinement algorithm—exhibited BR superadditivity. The observed BR subadditivity of some of the games implies a non-empty BRC: the best coalition structure in those games is stable. Even when BR subadditivity did not hold, the BRC was often non-empty—especially for large c_{comp}.

Another interesting observation is that the presented normative theory prescribes the bounded rational agents to choose coalition structures that agree closely with what human agents would select. The best BR coalition structures mostly agreed with our intuitions of what coalitions should form based on strategic domain specific considerations such as adjacency of the dispatch centers and the combinability of their loads. On the other hand, these coalition structures differ significantly from those which rational agents would choose.

Finally, although not uniformly true, higher computation unit costs seem to promote smaller coalitions than lower computation unit costs. This has a possible intuitive explanation. Each step of the refinement algorithm takes $\Theta(vd^2)$ time, where v is the number of vehicles and d is the number of deliveries. Because this is superlinear in deliveries, a larger coalition can make fewer refinement steps in a given time than the agents in partitions of that coalition can. To compensate, a refinement step of the larger coalition would need to reduce solution cost more than a refinement step of a smaller coalition. The size of the saving has to be averaged over all refinement steps in the optimal time allocation. If c_{comp} is low, more time is allocated, and small coalitions will often run out of profitable refinements. If c_{comp} is high, less time is allocated, and all coalitions will have profitable refinements, though the larger coalition will have time to make fewer of them. Thus it was not surprising that in games where the grand coalition was optimal, it was optimal for very small computation unit costs only.

Surprisingly, two agents colluding was often better than all agents working separately even for large computation unit costs. The result that higher computation unit costs promote smaller coalitions is somewhat deemphasized by our choice of not including the initial solution construction phase in the performance profiles. Shifting the performance profiles right to begin at the time when the initial solution was finished (instead of at zero) would shift the performance profiles of small coalitions less than the performance profiles of large coalitions because the initial solution construction is superlinear both in tasks and vehicles. Thus small coalitions would gain an advantage—that is most significant for large c_{comp}. If the time of initial solution generation is discarded, the best coalition structure for the highest computation unit costs depends only on the quality of the initial solutions of the different coalitions because no refinement steps are beneficial. For example, coalitions $\{1,5\}$, $\{2,5\}$, and $\{3,5\}$ achieved a better initial solution cost than the sum of the initial solution costs of the two agents separately, Fig. 4.3.

4.5 Domain Solution Interactions among Coalitions and Different Performance Profiles among Agents

As is common practice, so far in this chapter we studied coalition formation in characteristic function games (CFGs), Figure 4.2. In such games, the rational value of each coalition S is given by the characteristic function v_S^R, and is thus not a function of the actions of nonmembers. However, in general, the value of a coalition may depend on the actions of nonmembers due to positive and negative interactions of the agents' solutions. Such settings can be modeled as normal form games (NFGs), Fig. 4.2. CFGs are a strict subset of NFGs.[4]

Negative interactions between a coalition and nonmembers are often caused by shared resources of finite capacity. Once nonmembers are using the resource to a certain extent, not enough of that resource is available to agents in the coalition to carry out the planned solution at the minimum cost. Negative interactions can also be caused by conflicting goals. In satisfying their goals, nonmembers may actually move the world further from the coalition's goal state(s) [Rosenschein and Zlotkin, 1994]. Positive interactions are often caused by partially overlapping goals. In satisfying their goals, nonmembers may actually move the world closer to the coalition's goal

[4]The two are equivalent in constant-sum games with unrestricted side-payments and perfect communication. In such games, the characteristic function value of a coalition is its minimax value from the normal form game [van der Linden and Verbeek, 1985].

state(s), from where the coalition can reach its goals less expensively than it could have without the actions of nonmembers.

Let us now introduce a new domain class: bounded rational characteristic function game (BRCFG), Fig. 4.2. In BRCFGs, the value of each coalition S is defined by the bounded rational value $v_S(c_{comp})$. Thus, so far in this chapter we have studied BRCFGs.

Interactions between domain solutions of different coalitions may exclude some problems from the class BRCFG also. In general, the rational value of a coalition may depend on the actions of nonmember agents due to positive and negative interactions of the agents' domain solutions as discussed above. Such games are NFGs, but not CFGs. For the same reason, the value of some BR coalition's domain solution—computed by a BR agent—may depend on the actions of nonmembers, and is therefore not characterized by any $v_S(c_{comp})$. Such games are not in the class BRCFG.

There is also another reason why a game may not be a BRCFG. So far games where each agent has the same performance profile for a given coalition were presented. In general, domains where the agents have different performance profiles—due to different algorithms—are not BRCFGs, because the value of a coalition sometimes depends on the computational actions of nonmembers. The value of a coalition can depend on whether an outside agent is willing to compute the solution for the coalition (for a payment) if its algorithm is better than any of the algorithms of the agents in the coalition.

Games where the agents have different unit costs (c_{comp}) for computation—e.g. due to different execution architectures—are also in general not BRCFGs. Actually such games are analogous to games with a global c_{comp} but agents with different performance profiles. Namely, a game where agents have different computation unit costs can be modeled as a game with a uniform computation unit cost after the c_{comp}-axis of each $v_S(c_{comp})$ function is appropriately rescaled based on the real c_{comp} of the corresponding coalition S.

There exist BRCFGs that are not CFGs. This is due to the fact that one can construct games where the domain cost of the *actual solution* (for any coalition) attained by the algorithm of a BR agent may be independent of the actions of nonmembers even though the domain cost of the *best solution* attained by a rational agent depends on the actions of nonmembers. For example, in some domains it is possible to restrict oneself to using algorithms that only consider solutions whose value is not affected by nonmembers. There also exist CFGs that are not BRCFGs. For example, the agents may have different performance profiles and therefore the bounded rational value of a coalition may depend on whether nonmembers are willing to do the computation for the coalition. There is also another reason why some CFGs are not BRCFGs. The algorithms that the agents use may produce solutions whose values depend on the actions of nonmembers although the value of the optimal solution would not.

In the distributed vehicle routing problem of this dissertation, there were no positive or negative domain solution interactions between coalitions. There were no shared resources because all of the resources—vehicles and depots—were exclusively and exhaustively distributed among the agents (and thus among coalitions). Secondly, each agent (and thus each coalition) had its own goal: delivering all of the parcels at the lowest possible cost. A coalition's handling of its deliveries was unaffected by how nonmember agents handle their deliveries. Therefore this vehicle routing problem is a CFG. For the same reason, domain solution interactions do not preclude the problem from belonging to the class BRCFG. So, if all agents have the same performance profiles—as was assumed earlier—the distributed vehicle routing games are BRCFGs.[5] Yet if the agents had different performance profiles or computation unit costs, the games would not necessarily be within BRCFG.

[5]Also in the distributed scheduling domain, domain interactions do not occur unless agents share resources. On the other hand, even in toy problems such as the blocks world, positive and negative interactions often occur.

In non-CFGs, superadditivity, subadditivity, and the core are undefined (see Section 2.1.2.9). Thus, other solution concepts are necessary. Reasonable alternatives include the Nash equilibrium (Section 2.1.2.9), the Strong Nash equilibrium (Section 2.1.2.10) and the Coalition-Proof Nash equilibrium (Section 2.1.2.11). Analogous problems arise with BR agents. In non-BRCFGs, BR superadditivity, BR subadditivity, and the BRC are undefined. Again, other solution concepts are necessary, e.g. the Nash equilibrium or some of its mentioned refinements.

4.6 Chapter Summary

This chapter studied settings where agents coordinate their computational actions and real-world actions within each coalition but not across coalitions. A normative, domain-independent theory of coalitions in combinatorial domains was presented, where the rationality of self-interested agents is bounded by computational complexity. This work is an extension of game theory, which classically assumes perfect rationality: algorithms that find the optimal solution, and zero computation unit cost. On the other hand, in this chapter, computational limitations were quantitatively modeled by a unit cost of computation and the performance profiles of the agents' problem solving algorithms.

A domain classification was presented for bounded rational agents (Figure 4.2). It was also related to two existing domain classifications for fully rational agents: the traditional classification of game theory, and the classification of Rosenschein and Zlotkin.

The algorithms used by the agents significantly impact the coalition structure that should form as well as its stability. From the model of bounded rationality of this chapter, the optimal coalition structure can always be computed, and its stability determined. In addition, general theorems were proven that analytically state which types of performance profiles cause the best coalition structure to be the grand coalition irrespective of the computation unit cost (execution platform). Similar theorems were proven for the coalition structure where all agents work separately. General theorems were also presented that relate the performance profiles to the non-emptiness of the bounded rational core, which determines the stability of the coalition structure.

Although almost all domains are superadditive, bounded rational superadditivity is surprisingly all but obvious in practice. None of the vehicle routing games of our experiments—using real data and a reasonable iterative refinement algorithm—exhibited bounded rational superadditivity. The optimal coalition structure for bounded rational agents varied although rational agents should always form the grand coalition. Section 4.2 developed conditions on the performance profiles that guarantee bounded rational superadditivity. It also discussed a separate solving approach—based on a problem decomposition step—that guarantees that the base algorithm fulfills those conditions. With our reasonable deterministic iterative refinement algorithm, these conditions were—somewhat surprisingly—never met. The real desideratum is not necessarily to generate algorithms that guarantee bounded rational superadditivity (and thus the superiority of the grand coalition over other coalition structures), but algorithms that provide the highest social welfare (for the best coalition structure, which need not be the grand coalition). Sometimes these goals are conflicting.

The observed bounded rational subadditivity of some of the games implies a non-empty bounded rational core: the best coalition structure in those games is stable. Even when bounded rational subadditivity did not hold, the bounded rational core was often non-empty—especially for large computation unit costs. The theoretical model shows that there are games where a coalition structure is stable for rational agents but not for bounded rational ones, games where the reverse holds, games where a coalition structure is stable for both, and games where it is stable for neither.

The experiments suggest that often with superlinear iterative refinement steps, low computation unit costs promote large coalitions while high computation unit costs suggests smaller ones. A plausible explanation for this phenomenon was presented.

Another interesting observation is that the presented normative theory prescribes the bounded rational agents to choose coalition structures that agree closely with what human agents would select. The best coalition structures among bounded rational agents mostly agreed with our intuitions of what coalitions should form based on strategic domain specific considerations such as adjacency of the dispatch centers and the combinability of their loads. On the other hand, these coalition structures differ significantly from those which rational agents would choose.

CHAPTER 5

CONTRACT EXECUTION

In multiagent systems [Sandholm and Lesser, 1995c, Sandholm and Lesser, 1996, Sandholm and Lesser, 1995a, Sandholm and Lesser, 1997, Rosenschein and Zlotkin, 1994, Durfee *et al.*, 1993, Kraus *et al.*, 1992], the agents are provided with an interaction protocol, but each agent may choose its own strategy. This allows the agents to be constructed by separate designers and/or represent different real world parties. Agents in such systems often act based on self-interest, and the protocols have to be constructed accordingly. An example interaction protocol is the auction, where some agents bid to take responsibility for a task, which is awarded to the lowest price bidder. The bids are binding: if an agent makes a bid and the task is awarded to it, it must take responsibility for the task at that price. Among real world agents, this protocol is enforced by law.

Such enforced protocols are problematic when used among computational agents. First, there may be a lack of laws for interactions of computational agents, or the agents may be governed by different laws—e.g. sited in different countries. It may also be the case that the laws are not strictly enforced or that enforcing them (e.g. by litigation) is impractically expensive. We would like the agents' interactions to work properly independent of fluctuations in enforcement. Secondly, a computational agent may vanish at any point in time, e.g. by killing its own process. Thus, the laws cannot be enforced unless the terminated agent represented some real world party and the connection between the agent and the real world party can be traced. For example, the Telescript technology [General Magic, Inc., 1994] follows the approach of strictly tying each agent to its real world party. On the contrary, we analyze exchanges among more autonomous agents and study the possibility of exchange without enforcement (e.g. with unknown real world parties or no litigation possibility). In cases where this type of exchange is possible, it is clearly preferable to the strictly enforced mode of exchange due to savings in enforcement costs and lack of enforcement uncertainty—assuming that the cost of breaking deliveries into smaller units is practical and not expensive.

The fulfillment of a mutual contract can be viewed as one agent delivering and the other agent paying. We propose a method for carrying out such an exchange without enforcement. The exchange is managed so that for both agents—supplier and demander—at any point in the exchange, the future gains from carrying out the rest of the exchange (*cooperating* according to the contract) are larger than the gains from *defecting*. Defection means terminating the exchange prematurely by vanishing. For example, defection may be beneficial to a demander agent if the supplier agent has delivered much more than what the demander has yet paid for. By intelligently splitting the exchange into smaller portions, the agents can avoid situations where at least one of them is motivated to defect. We will call a sequence of deliveries and payments *safe* if neither agent is motivated to defect at any point in the exchange.

The basic idea of enhancing cooperation by making the present less important compared to the future has been suggested for example in [Axelrod, 1984]. Also, Telser has analyzed safe exchanges of goods and payments in a setting that is similar to the one of this chapter [Telser, 1980]. However, his work differs from the work of this chapter in many respects. It analyzes a sequence of exchanges as a repeated game where the end of the sequence is not known—except probabilistically. His analysis has a microeconomic character, but no game theoretic solution concepts are explicitly used. On the other hand, this chapter invokes specific game theoretic solution concepts. His agents can only defect between complete exchanges in the sequence, while ours evaluate the option of breaching within an exchange. Also, we explicitly relate the safe transactions to the agents' value functions while he uses two models of how the defection cost relates to the continuation cost without justifying those models. He also does not explicitly address the role of time while this chapter does. Finally, he analyzes a sequence where the order and sizes of the individual exchanges are fixed (i.e. given) while we present computationally plausible algorithms for splitting an exchange into parts, and a sound and complete algorithm for sequencing those parts.

We propose an *exchange manager module* to be added to each agent's architecture, Fig. 5.1. This module is potentially different for each agent. Its role is to schedule

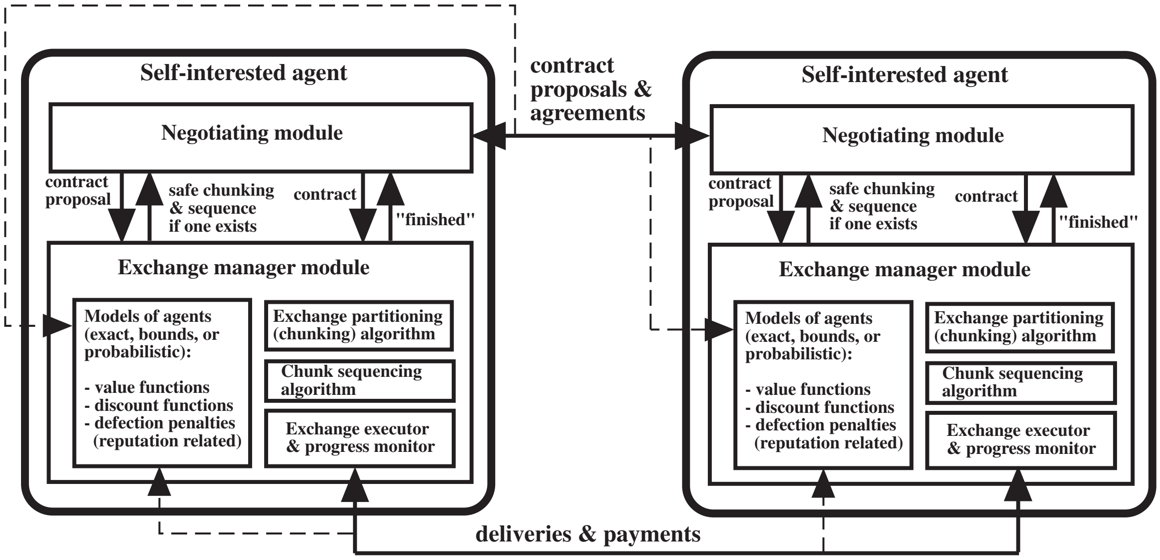

Figure 5.1. The architecture of two self-interested agents with special emphasis on the exchange manager modules. In the negotiation system, there may be more than two agents. The exchange manager module gives the negotiation module feedback of whether a proposed contract can be carried out safely, i.e. in equilibrium. The exchange manager also executes the exchange in each actual contract. The dashed lines show what information the exchange manager uses to update its models of other agents.

the agent's deliveries (or payments) in such a way that the opponent is not motivated to defect at any point in the exchange. This is in the agent's self-interest. The exchange manager also provides the agent's negotiator module with information on whether a certain proposed contract can be carried out safely. Unless protocol enforcement is guaranteed, the negotiator should only agree to contracts that can be executed so that the opponent is not motivated to defect at any point of the exchange. Automated negotiation has been mostly studied with respect to *ex ante* rationality: what contracts seem desirable to the agents and are stable (in equilibrium in the sense that agents do not want to deviate from abiding by the contract given that other agents do not deviate) *before* the contract is carried out [Sandholm and Lesser, 1995a, Sandholm and Lesser, 1997, Sandholm and Lesser, 1995c, Sandholm and Lesser, 1996, Sandholm, 1993, Rosenschein and Zlotkin, 1994, Kraus *et al.*, 1992, Wellman, 1992, Durfee *et al.*, 1993, Tennenholtz and Moses, 1989]. We suggest that contracts should also fulfill the condition of *ex post* rationality: abiding by the contract should be desirable to the agents at each step of the carrying out of the contract. *Ex post* conditions were studied in multiagent planning without side payments in [Brainov, 1994].

5.1 Unenforced Exchange of Goods and Payments

Our model analyzes exchanging goods—information, computation services, or other types—for payments. The exchange proceeds on two axes: the portion of goods of the contract delivered by exchange step n is called $x_n \in [0,1]$, and the cumulative payment so far is $p_n \in [0, p^{contr}]$. The quantity p^{contr} is the total payment specified in the contract. The agents can make simultaneous moves and they observe the other agent's moves so far. They value the goods x according to nondecreasing functions that are in equivalent units of payment p. The supplier's value function $v_s(x)$ describes how much cost the supplier incurs by generating and delivering x. The demander's value function $v_d(x)$ describes what the goods x are worth to the demander. Trivially, $x_0 = 0$, $p_0 = 0$, and $v_s(0) = v_d(0) = 0$.

At any point (x_n, p_n) of the exchange, the supplier can choose $x_{n+1} \geq x_n$. The mapping from any sequence $(x_0, p_0), (x_1, p_1), ..., (x_n, p_n)$ to x_{n+1} is called the supplier's exchange *strategy*, call it $S_{supplier}$. Similarly, the demander's exchange strategy $S_{demander}$ is a mapping from any $(x_0,p_0),(x_1,p_1),...,(x_n,p_n)$ to $p_{n+1} \geq p_n$.[1] Together $S_{supplier}$ and $S_{demander}$ induce a *path of play* in the game (i.e. starting from $(x_0,p_0)=(0,0)$): $(x_0,p_0),(x_1,p_1),...$. This model incorporates the possibility that for some step q, $x_q = x_{q+1} = ... = \lim_{M\to\infty} x_M$ or $p_q = p_{q+1} = ... = \lim_{M\to\infty} p_M$, i.e. cumulative delivery or payment is not increased after a certain step of the exchange. This may be due to the completion of the exchange or due to premature breach. The supplier's payoff starting from subgame (x_n,p_n) where $n \geq 0$ is[2]

$$\begin{aligned}\pi_s^{(x_n,p_n)}(S_{supplier},S_{demander}) &= \sum_{q=n}^{\infty} p_{q+1}-p_q-[v_s(x_{q+1})-v_s(x_q)] \\ &= \lim_{M\to\infty} p_M - p_n - [v_s(x_M)-v_s(x_n)]\end{aligned}$$

and the demander's payoff starting from subgame (x_n,p_n) is

$$\begin{aligned}\pi_d^{(x_n,p_n)}(S_{supplier},S_{demander}) &= \sum_{q=n}^{\infty} v_d(x_{q+1})-v_d(x_q)-[p_{q+1}-p_q] \\ &= \lim_{M\to\infty} v_d(x_M)-v_d(x_n)-[p_M-p_n]\end{aligned}$$

Defecting, i.e. exiting, in the exchange corresponds to choosing $\lim_{M\to\infty} x_M < 1$ by the supplier or $\lim_{M\to\infty} p_M < p^{contr}$ by the demander.

Roughly speaking, a payoff maximizing supplier agent will cooperate throughout the rest of the exchange from an arbitrary point (x,p) if its future compensation is at least as great as its future cost[3], i.e. $p^{contr} - p \geq v_s(1) - v_s(x)$. This assumes that the demander will actually finally increase cumulative payment to p^{contr}. An equilibrium analysis with respect to this issue will be presented shortly. To facilitate that analysis,

[1]Although we allow strategies of this general form, we will show in this chapter that all of the desirable properties can be obtained by Markov strategies where an agent's each move is a function of current state (x_n,p_n) only, not of the entire history.

[2]The physical transaction cost of delivering and paying is assumed negligible compared to the cost of producing the goods and the value of the payments. It is therefore not incorporated in the model. It follows that the agents' payoffs do not depend on the number of parts that the whole exchange is broken into.

[3]If equality holds, the agent is indifferent between cooperating and defecting. Throughout this chapter we assume that indifferent agents will cooperate. Note also that throughout the chapter we analyze only remaining payoffs, not total payoffs, because the already incurred payoffs are constant with respect to the remaining game.

$p^{max}(x)$ is defined based on the above intuition as the maximum cumulative payment that the demander can pay for a given cumulative delivery x without inducing the supplier to defect:

$$p^{max}(x) \stackrel{\text{def}}{=} p^{contr} - v_s(1) + v_s(x). \tag{5.1}$$

See Figure 5.2 left.

Roughly speaking, a rational demander agent will cooperate throughout the rest of the exchange from an arbitrary point (x, p) if the future compensation it has to pay is smaller than or equal to its added value, i.e. $p^{contr} - p \leq v_d(1) - v_d(x)$. This assumes that the supplier will finally increase total delivery to 1, which will be shown to be an equilibrium shortly. Now, $p^{min}(x)$ is defined as the minimum cumulative payment that has to be made for a given cumulative delivery x so that the demander is not induced to defect:

$$p^{min}(x) \stackrel{\text{def}}{=} p^{contr} - v_d(1) + v_d(x). \tag{5.2}$$

Clearly, $p^{max}(x)$ and $p^{min}(x)$ are nondecreasing in x. For the supplier to have agreed to the contract, $p^{contr} \geq v_s(1)$ has to hold. It follows that $p^{max}(0) \geq 0$. Similarly, for the demander to have agreed to the contract, $p^{contr} \leq v_d(1)$. It follows that $p^{min}(0) \leq 0$.

If the agents do not know each other's value functions, they can use bounds they know. The supplier is safe using an upper bound for $p^{min}(x)$, i.e. a lower bound for $v_d(1)$ and an upper bound for $v_d(x)$. The demander is safe using a lower bound for $p^{max}(x)$. Although the agents are safe using these bounds, even possible exchanges are disabled if the bounds are too far off.

The next sections present an equilibrium study of when safe exchange can actually occur. The analysis is slightly different for countable and uncountable goods.

5.2 Countable Goods

Countable goods are goods that are inherently split into atomic chunks. Conceptually there can be a finite or an infinite number of such chunks. The chunks cannot

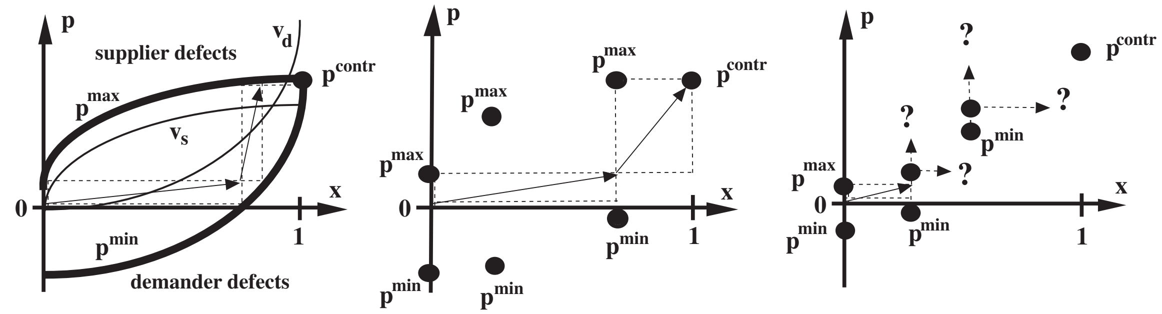

Figure 5.2. Examples of unenforced exchange. Left: safe exchange with uncountable goods. Middle: safe exchange of countable goods possible. Right: safe exchange of countable goods not possible. In each of the three cases, $p^{min}(1) = p^{max}(1) = p^{contr}$. The agents make simultaneous moves and try to stay inside the safe region, i.e. the supplier does not want to deliver too much at once and the demander does not want to pay too much at any one step.

be further split, i.e. they are not arbitrarily divisible. In this section we assume that the delivery order of the chunks is fixed. For example in the TRACONET (TRAnsportation COoperation NET) multiagent system [Sandholm, 1993], agents representing dispatch centers negotiated over whose vehicles should transport which parcels. Taking care of one parcel is an atomic chunk because the task cannot be split. Sometimes a contract between two agents involved multiple tasks (in order to avoid local optima in distributed task allocation [Sandholm, 1993]) so the total exchange could have been split into smaller parts. For example, the demander could first pay part of the contract price. Then the supplier would deliver one parcel. After that the demander would pay some more, the supplier would deliver another parcel etc.

Now two exchange strategies are formally presented; one for the supplier and one for the demander. Intuitively, the supplier's strategy is to deliver as much as it can while still making sure that the demander will benefit more from cooperating throughout the rest of the exchange than from vanishing at any point. The demander's strategy is to pay as much as it can while still making sure that the supplier will benefit more from cooperating throughout the rest of the exchange than from exiting. These strategies have the desired properties—under certain conditions—that they lead to completion of the exchange in a minimal number of steps and that they are in equilibrium: neither agent can do better by switching to another strategy given that the other agent does not switch. The required conditions are minimal in the sense that if they do not hold, *no* strategies lead to completion of the exchange in equilibrium. These properties are formally stated in the two theorems that follow the definitions of the strategies.

Definition 5.1 (Supplier's strategy S_s for countable goods) *At any point (x_n, p_n) of the exchange, if $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$, deliver an amount such that cumulative delivery $x_{n+1} = max\{x \in X | p^{min}(x) \leq p_n\}$. Else exit.*

Definition 5.2 (Demander's strategy S_d) *At any point (x_n, p_n) of the exchange, if $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$, pay an amount such that cumulative payment $p_{n+1} = p^{max}(x_n)$. Else exit.*

The following theorems regarding these strategies use two concepts from game theory. *Nash equilibrium* [Nash, 1950b, Kreps, 1990, Fudenberg and Tirole, 1991] means that each agent is motivated to adhere to its specified strategy given that the other agent adheres to its specified strategy. *Subgame perfection* [Selten, 1965, Kreps, 1990, Fudenberg and Tirole, 1991] means that the equilibrium is a Nash equilibrium at any point (x_n, p_n) of the exchange, not only the beginning of it. This means that the equilibrium remains an equilibrium after the agents have partially carried out the exchange. Furthermore, it is an equilibrium at points (x, p) of the exchange that will actually not be reached by the agents in the exchange process. For these reasons, subgame perfection precludes incredible threats/promises and provides some robustness against external perturbances.

Theorem 5.1 *With a finite number of countable goods (discrete $X \subset [0, 1]$), for nondecreasing $p^{min}(x)$ and $p^{max}(x)$, the strategies S_s and S_d form a subgame perfect*

Nash equilibrium[4] *if for every two consecutive amounts of cumulative delivery* $x, x' \in X$, $p^{max}(x) \geq p^{min}(x')$.[5] *The exchange is completed in a finite number of steps. Furthermore, there is no subgame perfect Nash equilibrium that leads to completion in fewer steps. See Fig. 5.2 middle.*

Theorem 5.2 *With a finite number of countable goods (discrete* $X \subset [0,1]$*), for nondecreasing* $p^{min}(x)$ *and* $p^{max}(x)$*, there is no subgame perfect Nash equilibrium leading to completion of the exchange if for some two consecutive amounts of cumulative delivery* $x, x' \in X$, $p^{max}(x) < p^{min}(x')$. *See Fig. 5.2 right.*

From the condition $p^{max}(x) \geq p^{min}(x')$ and the fact that $p^{min}(x)$ is nondecreasing we see that the following has to hold for safe exchange: $p^{max}(x) \geq p^{min}(x)$. In terms of the agents' value functions this can be written as $v_d(x) - v_s(x) \leq v_d(1) - v_s(1)$. This means that the agents' joint profit must be higher (or equal) at $x = 1$ than at any other $x \in [0, 1)$. If the agents would have been better off by making the contract for a smaller amount of goods, an isolated safe exchange is impossible. Furthermore, at $x = 0$ this gives $v_s(1) \leq v_d(1)$, which is an intuitive condition for the contract to have been made in the first place. Specifically, $v_s(1) \leq p^{contr} \leq v_d(1)$, Fig. 5.2 left, i.e., neither agent incurs a loss in the contract.

The fulfillment of the condition $p^{max}(x) \geq p^{min}(x')$ is facilitated by $p^{max}(x)$ being high and $p^{min}(x')$ being low. This means that safe exchange is enhanced if the supplier incurs most of its cost from the early portion of the exchange, while the possibility that the demander acquires most of its value already from the early parts hinders safe exchange. Luckily, in practise, the supplier usually does have setup costs and the demander often acquires most of the value from the completion of the exchange.

[4]The equilibria in Theorems 5.1, 5.3, 5.5 and 5.7 are not unique. For example, the strategies that specify that the demander never pays anything and the supplier never delivers anything at any point of the exchange also form a subgame perfect Nash equilibrium.

[5]Consecutive amounts are amounts that differ by just one chunk. This consideration is independent of the agents' strategies. At any step of the exchange, the supplier can deliver the next chunk, a number of next chunks, or nothing. Thus $x_n = x \not\Rightarrow x_{n+1} = x'$.

Theorems 5.1 and 5.2 state that rather stringent conditions have to be met to enable unenforced isolated exchange of a finite number of countable goods. Substituting $x = 1$ in the definitions of $p^{max}(x)$ and $p^{min}(x)$ gives $p^{max}(1) = p^{min}(1) = p^{contr}$. According to the theorems, safe exchange is possible only if $p^{max}(x) \geq p^{min}(x')$ for every consecutive x and x'. Let us call the size of the last delivery Δx. So for safe exchange the following has to hold: $p^{max}(1 - \Delta x) \geq p^{min}(1) = p^{max}(1)$. Thus the increasing function $p^{max}(x)$ has to be constant during the last step (Fig. 5.2 middle). This means that the supplier's value function $v_s(x)$ is constant. So an isolated safe exchange is possible only if the supplier does not incur any cost from generating and delivering the last chunk. This occurs for example when the supplier has had to acquire a number of the last deliverables atomically. Its cost of acquiring the deliverables can be entirely attributed to the first one, while it can deliver these items separately with only the first one increasing $v_s(x)$ (assuming negligible costs of physically delivering). This may not occur very often in practise (Fig. 5.2 right). Intuitively, when there is no future benefit to be gained from exchanging, agents are better off defecting on the current move.

If this problem occurs in an isolated exchange of a finite number of countable goods, it spoils the entire exchange. On the last move the supplier does not want to increase delivery to $x = 1$, because the demander would defect. Similarly, the demander does not want to increase cumulative payment above $p^{max}(1-\Delta x)$, because the supplier would defect. Both agents know that the last part of the exchange will not take place due to this. So they can analyze the exchange as if it did not have the last part. Now the second to last part has the same problem (unless the supplier can deliver that part without cost): neither agent wants to initiate that part. Again, both agents know this and so on. This backward induction can be carried out up to the first exchange. So, neither agent will make any move, and the exchange will not take place. Theoretically, there can be an infinite number of countable goods. In such cases this exchange spoiling backward induction argument does not apply because at no point can an agent say that the next move is the last. Backward induction is inapplicable also with uncountable goods. This facilitates safe unenforced exchange of uncountable goods, as discussed in the next section. Both with countable and with uncountable goods, the problem of requiring that the supplier can deliver the last part without costs can be overcome by considering related future interactions of the agent, Sec. 5.4.

5.3 Uncountable Goods

This section analyzes the exchange of uncountable goods, i.e. goods that are not split into given fixed chunks but can be split arbitrarily—e.g. gasoline, coffee, or monetary currency without rounding. First, a simple condition for safe exchange is presented. Intuitively it says that if the demander has paid up to $p^{max}(x)$, then the supplier can safely supply at least ϵ more without the total payment falling below $p^{min}(x+\epsilon)$.

Condition 5.1 (Safe unenforced exchange of uncountable goods) $\exists \epsilon > 0$ *s.t.* $\forall x^* \in [0, 1-\epsilon], p^{max}(x^*) \geq p^{min}(x^* + \epsilon)$.

This condition is fully equivalent to two more detailed conditions which are presented here for the interested reader. Intuitively, the first one (ϵ-reachability) states the conditions under which a safe exchange can proceed to some amount of cumulative delivery. The second one (ϵ-departability) states the conditions under which safe exchange can proceed from some amount of cumulative delivery.

Condition 5.2 (ϵ-reachability) $\exists \epsilon > 0$ *s.t.* $\forall x^* \in (0, 1]$,

1. $p^{max}(x^*) = p^{min}(x^*)$, *and* $p^{max}(x)$ *is constant in the closed left* ϵ*-neighborhood of* x^*, *or*

2. $p^{max}(x^*) > p^{min}(x^*)$, $\lim_{x \to x^*-} p^{max}(x) > p^{min}(x^*)$, *and* $p^{max}(x^* - \epsilon) \geq p^{min}(x^*)$, *or*

3. $p^{max}(x^*) > p^{min}(x^*)$, $\lim_{x \to x^*-} p^{max}(x) = p^{min}(x^*)$, *and* $p^{max}(x)$ *is constant in the closed left* ϵ*-neighborhood of* x^*.

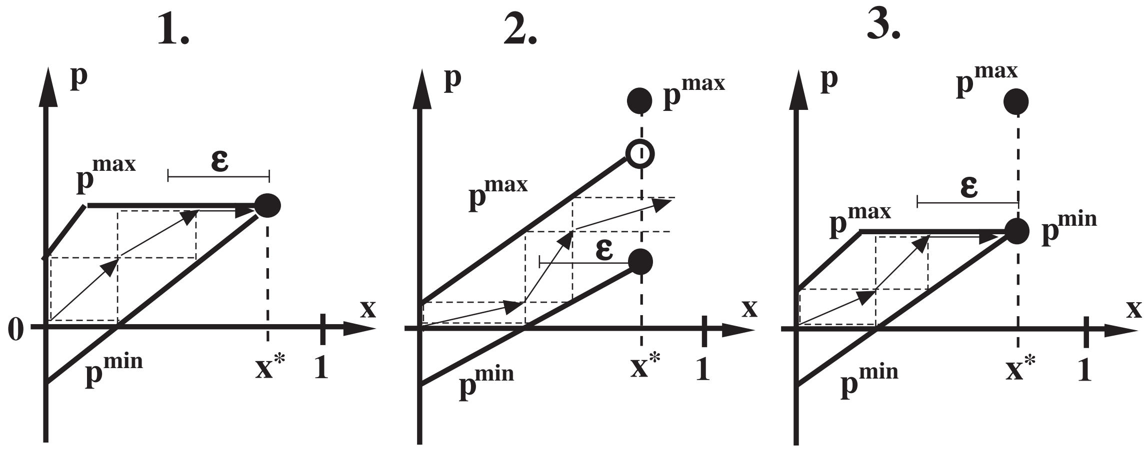

Figure 5.3. Exchanging uncountable goods: reaching a point (ϵ-reachability).

Condition 5.3 (ϵ-departability) $\exists \epsilon > 0$ *s.t.* $\forall x^* \in [0, 1)$,

1. $p^{max}(x^*) = p^{min}(x^*)$, *and* $p^{min}(x)$ *is constant in the closed right* ϵ*-neighborhood of* x^*, *or*

2. $p^{max}(x^*) > p^{min}(x^*)$, $\lim_{x \to x^*+} p^{min}(x) < p^{max}(x^*)$, *and* $p^{max}(x^*) \geq p^{min}(x^* + \epsilon)$, *or*

3. $p^{max}(x^*) > p^{min}(x^*)$, $\lim_{x \to x^*+} p^{min}(x) = p^{max}(x^*)$, *and* $p^{min}(x)$ *is constant in the closed right* ϵ*-neighborhood of* x^*.

In what follows, theorems regarding these conditions are stated. They are stated in terms of Condition 5.1. Before that, however, the supplier's strategy needs to be defined for uncountable goods, because S_s is not necessarily well defined in such settings:

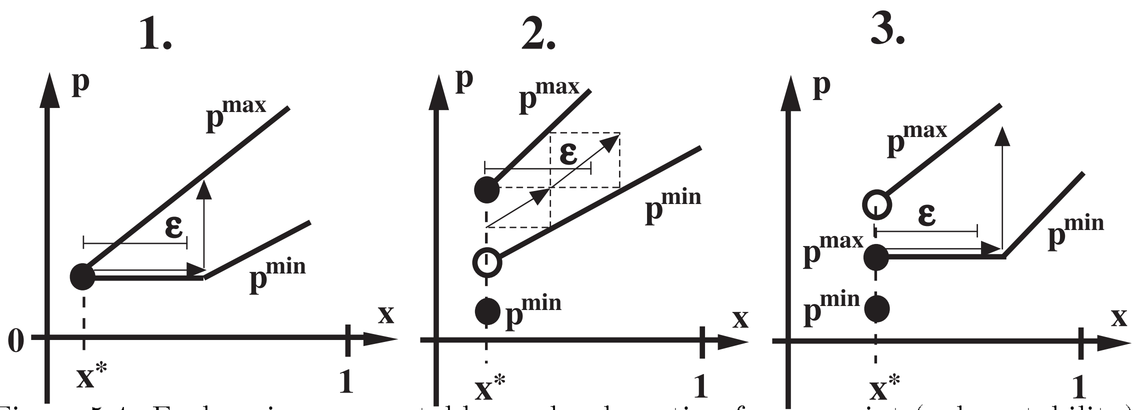

Figure 5.4. Exchanging uncountable goods: departing from a point (ϵ-departability).

Example. Result of max-operator undefined. Let again $x_0 = p_0 = 0$, and say

$$p^{min}(x) = \begin{cases} 0 & \text{if } x < 1 \\ 1 & \text{if } x \geq 1 \end{cases}$$

Now S_s specifies $x_1 = max\{x \in X | p^{min}(x) \leq p_0\}$. But this quantity is undefined. It is not 1 because then $0 = p_0 < p^{min}(x) = p^{min}(1) = 1$. For the same reason it cannot be any number above 1 either. It cannot be any number $x' < 1$ because another number $x'' = \frac{x'+1}{2} > x'$ fulfills the condition $0 = p_0 \geq p^{min}(x'') = 0$.

Thus we have to define a new strategy for the supplier:

Definition 5.3 (Supplier's strategy $S_{s_{un}}$ for uncountable goods) *At any point (x_n, p_n) of the exchange, if $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$, deliver an amount such that cumulative delivery $x_{n+1} = max\{x \in X | p^{min}(x) \leq p_n\}$ if this is defined and $x_{n+1} = max(x_n, sup\{x \in X | p^{min}(x) \leq p_n\} - \underline{\epsilon})$ otherwise. Here $\underline{\epsilon}$ is fixed, $0 < \underline{\epsilon} < \epsilon$, and ϵ is from Condition 5.1. Exit if $p_n > p^{max}(x_n)$, (or $p_n < p^{min}(x_n)$).*

The subtraction of $\underline{\epsilon}$ from the supremum is done in order to maintain the invariant $p^{min}(x_{n+1}) < p_n$. Note that whenever well-defined, $max\{x \in X | p^{min}(x) \leq p_n\}$ is equal to $sup\{x \in X | p^{min}(x) \leq p_n\}$. Now, in the above example, $S_{s_{un}}$ prescribes $x_1 = sup\{x \in X | p^{min}(x) \leq p_0\} - \underline{\epsilon} = sup\{x \in X | p^{min}(x) \leq 0\} - \underline{\epsilon} = 1 - \underline{\epsilon}$.

The demander's strategy S_d does not need to be redefined because it is well defined also with uncountable goods.

Theorem 5.3 *With uncountable goods ($X = [0,1]$), for nondecreasing $p^{min}(x)$ and $p^{max}(x)$, the strategies $S_{s_{un}}$ and S_d form a subgame perfect Nash equilibrium if $\exists \epsilon > 0$ s.t. $\forall x^* \in [0, 1-\epsilon], p^{max}(x^*) \geq p^{min}(x^* + \epsilon)$, i.e. if Condition 5.1 holds. The exchange is completed in a finite number of steps. Furthermore, $\exists \underline{\epsilon}^* > 0$ s.t. $\forall \underline{\epsilon}$ where $0 < \underline{\epsilon} \leq \underline{\epsilon}^*$, the exchange has a minimal number of steps out of all subgame perfect Nash equilibrium exchanges leading to completion.*

Theorem 5.4 *With uncountable goods ($X = [0,1]$), for nondecreasing $p^{min}(x)$ and $p^{max}(x)$, there is no subgame perfect Nash equilibrium leading to completion of the exchange in a finite number of steps if $\forall \epsilon > 0, \exists x^* \in [0, 1-\epsilon]$ s.t. $p^{max}(x^*) < p^{min}(x^* + \epsilon)$, i.e. if Condition 5.1 does not hold.*

Theorems 5.3 and 5.4 state that isolated unenforced exchange of uncountable goods is safe if (and only if) some initial delivery can be made, every intermediate amount of delivery can be reached and departed, and the final amount of delivery can be reached without exceeding $p^{max}(x)$ or moving below $p^{min}(x)$. These theorems do not assume continuity of $p^{max}(x)$ (equivalently $v_s(x)$) or $p^{min}(x)$ (equivalently $v_d(x)$). Neither do they assume that $p^{max}(x)$ or $p^{min}(x)$ is strictly increasing. It follows from Theorems 5.3 and 5.4 that with continuous $p^{max}(x)$ and $p^{min}(x)$, the exchange can be carried out in subgame perfect Nash equilibrium in a finite number of steps if and only if $\exists \epsilon > 0$ s.t. $\forall x \in [0,1]$, either $p^{max}(x) > p^{min}(x)$ or $p^{max}(x) = p^{min}(x)$ and $p^{max}(x)$ is constant in a left neighborhood of x (except at $x = 0$), and $p^{min}(x)$ is constant in a right neighborhood of x (except at $x = 1$). If in addition to continuity, $p^{max}(x)$ and $p^{min}(x)$ are strictly increasing, the exchange can be made safely if and only if $\forall x \in [0,1], p^{max}(x) > p^{min}(x)$.

Isolated safe exchange can be problematic also in the case of uncountable goods. Substituting $x = 1$ into the definitions of $p^{max}(x)$ and $p^{min}(x)$, we see that $p^{max}(1) =$

$p^{contr} = p^{min}(1)$. From case 1 of Condition 5.2 we see that full delivery ($x = 1$) can be reached only if $p^{max}(x)$ is constant in a left neighborhood of $x = 1$. If the value function of the supplier $v_s(x)$ is not constant in the end of the exchange, the exchange cannot be completed. So again—as was the case with finitely many countable goods—an isolated safe exchange is possible only if the supplier does not incur any cost from generating and delivering the last portion of the goods. This problem is less severe than in the case of finitely many countable goods. For example in the case where $p^{min}(x)$ and $p^{max}(x)$ are continuous and $p^{min}(x) < p^{max}(x)$ for all x in a left neighborhood of $x = 1$, and $\lim_{x \to 1} p^{min}(x) = \lim_{x \to 1} p^{max}(x) = p^{contr}$, the exchange can be made to *approach* completion—e.g. by our original strategies S_s and S_d. On the resulting equilibrium path, the size of the deliveries decreases and approaches zero—thus making the number of deliveries infinite (Fig. 5.2 left). This allows the agents to reach a cumulative delivery that is arbitrarily close to 1 because the backward induction argument that disabled the entire exchange in the case of a finite number of countable goods does not hold. There is no particular exchange step at which the agents could reason that neither will make a move.

The next section describes how the entire exchange can be carried out by accounting for the fact that defecting during this exchange will adversely affect the agent's future.

5.4 Non-Isolated Exchange

Often an agent interacts with other agents more than once. One interaction may affect the agent's future interactions. For example, if an agent defects in the current exchange, its counterpart may not want to take on future contracts with that agent. Moreover, the counterpart can notify other agents that the agent defected. Thus, the agent's interactions with third parties may also be hindered by defecting in the current exchange. The hindering future impact of a defection can be thought of as an extra cost. This future cost may motivate agents to cooperate in the current exchange even if it would be rational to defect in it when considered in isolation. The methods for calculating defection impacts on the future are beyond the scope of this dissertation. We assume that both agents know their own and their opponent's defection costs. We denote the supplier's defection cost by c_s^{def} and the demander's by c_d^{def}. The defection costs can be incorporated into the model by redefining $p^{max}(x)$ and $p^{min}(x)$, Fig. 5.5:

$$p^{max'}(x) \stackrel{\text{def}}{=} p^{contr} - v_s(1) + v_s(x) + c_s^{def}. \tag{5.3}$$

$$p^{min'}(x) \stackrel{\text{def}}{=} p^{contr} - v_d(1) + v_d(x) - c_d^{def}. \tag{5.4}$$

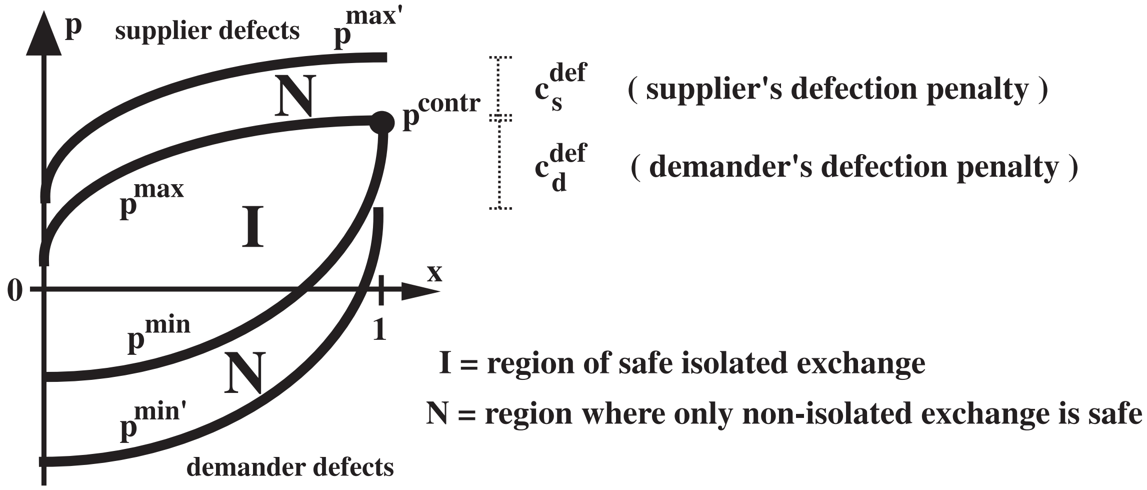

Figure 5.5. Defection penalties of non-isolated exchange give leeway to safe moves.

In isolated exchange, substituting $x = 1$ in the definitions of $p^{max}(x)$ and $p^{min}(x)$ gave $p^{min}(1) = p^{contr} = p^{max}(1)$. This led to the problem that the exchange could not be carried out to completion—unless $p^{max}(x)$ was constant in the end of the exchange. In non-isolated exchange, substituting $x = 1$ gives $p^{min'}(1) = p^{contr} - c_d^{def}$ and $p^{max'}(x) = p^{contr} + c_s^{def}$. The defection penalties give leeway to the exchange, thus possibly enabling safe exchange to be completed even if $p^{max}(x)$ is not constant in the end.

The contract price p^{contr} could be exceeded due to this leeway. To avoid this, the demander's strategy can be modified so that at any point in the exchange, the demander increases cumulative payment to $min(p^{contr}, p^{max'}(x))$ instead of $p^{max'}(x)$. This will not hinder exchange, because the condition takes effect after the full contract price has been paid.

Non-isolated exchange is more fruitful than isolated exchange, because it facilitates safe completion. The theorems on the possibilities of subgame perfect Nash equilibrium exchange (5.1, 5.2, 5.3, and 5.4) apply directly to the case of non-isolated exchange with the new definitions $p^{max'}(x)$ and $p^{min'}(x)$ substituted in place of $p^{max}(x)$ and $p^{min}(x)$, and the minor modification in the demander's strategy.

With uncountable goods, even the smallest reputation related defection penalties suffice to allow safe completion of the exchange—even if the supplier cannot deliver the last portion without cost. On the other hand, with countable goods, the defection penalties need to be larger if the smallest possible atomic chunks are large so that the supplier can safely deliver the last chunk, and the demander can safely pay for the last portion.

If the agent does not know the defection cost of the opponent, it can use a lower bound for that cost. This way the agent is safe, but if the bound is too far off, even possible exchanges are disabled.

5.5 Role of Time

This section addresses real time in the exchange: will an exchange take place immediately, or will it be infinitely postponed, or something in between? Nonincreasing discount functions $f(t_n)$ are assumed where $0 \leq f(t_n) \leq 1$, and $f(0) = 1$. Subscripts s and d distinguish between the supplier and the demander, and superscripts p and v characterize whether the discount applies to payment or the value of goods. For example a discount function value $f_d^v(3) = \frac{1}{2}$ means that at time three, a good is only worth half of what it was worth in the beginning of the exchange to the demander. As another example, using constant interest rate (r) compounded interest, the discount function is $f(t_n) = e^{-rt_n}$ [Rasmusen, 1989].

Real time is incorporated into the protocol by allowing the agents to postpone their moves. Again, the exchange conceptually proceeds in steps, but now the agent that makes a move during a step determines the *time* of the step. Note that the protocol does not specify which agent is to move at each step: simultaneous moves are also legal. For example, say that the exchange has reached delivery x_n and payment p_n and both agents are currently postponing. At this point, the supplier can bind the variable t_{n+1} representing the time of the next action by increasing cumulative delivery to $x_{n+1} > x_n$. Alternatively, the demander can bind t_{n+1} by increasing cumulative payment to $p_{n+1} > p_n$. Although the first agent to make a new move determines t_{n+1}, it can happen—due to simultaneous moves—that both agents determine the (same) value for t_{n+1}.

5.5.1 Role of Time in Isolated Exchange

This subsection analyzes the role of time in exchanges where an agent's defection does not impact its future contracting possibilities. The next subsection will discuss time in non-isolated exchanges.

In any subgame (x_n, p_n, t_n), the supplier's remaining payoff is

$$\pi_s^{(x_n,p_n,t_n)}(S_s^{timed}, S_d^{timed}) = \sum_{q=n}^{\infty} f_s^p(t_{q+1})[p_{q+1} - p_q] - f_s^v(t_{q+1})[v_s(x_{q+1}) - v_s(x_q)]$$

and the demander's payoff starting from subgame (x_n, p_n, t_n) is

$$\pi_d^{(x_n,p_n,t_n)}(S_s^{timed}, S_d^{timed}) = \sum_{q=n}^{\infty} f_d^v(t_{q+1})[v_d(x_{q+1}) - v_d(x_q)] - f_d^p(t_{q+1})[p_{q+1} - p_q]$$

Now, specific real-time incorporating strategies are introduced for both players. They are analogous to the original strategies S_s and S_d except that the agents make their moves *immediately*.

Definition 5.4 (Supplier's strategy S_s^{timed} for countable goods) *At any point* (x_n, p_n, t_n) *of the exchange where* $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$, immediately *deliver an amount such that cumulative delivery* $x_{n+1} = max\{x \in X | p^{min}(x) \leq p_n\}$, *and* $t_{n+1} = t_n$. *Exit if* $p_n > p^{max}(x_n)$ *(or* $p_n < p^{min}(x_n)$*).*

Definition 5.5 (Demander's strategy S_d^{timed}) *At any point* (x_n, p_n, t_n) *of the exchange where* $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$, immediately *pay an amount such that cumulative payment* $p_{n+1} = p^{max}(x_n)$, *and* $t_{n+1} = t_n$. *Exit if* $p_n < p^{min}(x_n)$ *(or* $p_n > p^{max}(x_n)$*).*

The following theorem states that neither agent is motivated to unilaterally deviate from immediate exchange if certain conditions hold on the discount factors. This result is not obvious because conceivably the agent who is behind in deliveries or payments could be better off by postponing so that the remainder of the exchange becomes less important due to the discounts. The proof of Theorem 5.5 is quite tedious.

Theorem 5.5 *With a finite number of countable goods (discrete* $X \subset [0, 1]$*), for nondecreasing* $p^{min}(x)$ *and* $p^{max}(x)$*, the strategies* S_s^{timed} *and* S_d^{timed} *form a Nash equilibrium if for every two consecutive amounts of cumulative delivery* x *and* x', $p^{max}(x) \geq p^{min}(x')$, *and* $\forall t \geq 0, f_s^p(t) \leq f_s^v(t), f_d^p(t) \geq f_d^v(t)$. *Moreover, the equilibrium is a Nash equilibrium in every subgame that is reached. The exchange is completed immediately in a finite number of steps. Furthermore, out of all Nash equilibria that remain Nash equilibria in every subgame that is reached and lead to immediate completion, this equilibrium guarantees a minimal number of steps to completion.*

So, isolated unenforced exchange is feasible if the supplier discounts payments more sharply (or equally) than production costs and the demander discounts the value of goods more sharply (or equally) than payment. The condition on the supplier's discount functions is rather natural. For example in a stable environment, the supplier's current value of producing an item should remain constant, but obviously payment is discounted. The condition on the demander's discount functions is more stringent. It is realistic in the case where the demander needs the goods urgently. An agent need not know the opponent's exact discount functions. It is sufficient to know whether they fulfill the conditions. The equilibrium concept of the theorem is slightly weaker than subgame perfection because it only guarantees that the equilibrium is a Nash equilibrium in subgames that are reached—not all subgames. In practise this means that if, for some unknown reason, the exchange has been delayed, it is not guaranteed that the agents are motivated to proceed immediately or at all:

Theorem 5.6 *The Nash equilibrium of Theorem 5.5 is not subgame perfect.*

Clearly, by Theorem 5.2, immediate exchange of countable goods is impossible in equilibrium if the condition on the consecutive x's does not hold. Similarly, by Theorem 5.4, immediate exchange is not possible with uncountable goods if Condition 5.1 does not hold. If they do hold, immediate exchange is possible also in the uncountable case. To show this, we again define a strategy for the supplier for uncountable goods where there is a possibility that the result of the max-operator in S_s^{timed} is undefined.

Definition 5.6 (Supplier's strategy $S_{s_{un}}^{timed}$ for uncountable goods) *At any point (x_n, p_n, t_n) of the exchange, if $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$,* immediately *deliver an amount such that cumulative delivery $x_{n+1} = max\{x \in X | p^{min}(x) \leq p_n\}$ if this is defined and $x_{n+1} = max(x_n, sup\{x \in X | p^{min}(x) \leq p_n\} - \underline{\epsilon})$ otherwise. Here $\underline{\epsilon}$ is fixed, $0 < \underline{\epsilon} < \epsilon$, and ϵ is from Condition 5.1. Exit if $p_n > p^{max}(x_n)$, (or $p_n < p^{min}(x_n)$).*

Theorem 5.7 *With uncountable goods ($X = [0, 1]$), for nondecreasing $p^{min}(x)$ and $p^{max}(x)$, the strategies $S_{s_{un}}^{timed}$ and S_d^{timed} form a Nash equilibrium if $\exists \epsilon > 0$ s.t. $\forall x^* \in [0, 1-\epsilon], p^{max}(x^*) \geq p^{min}(x^* + \epsilon)$ (i.e. if Condition 5.1 holds), and $\forall t \geq 0, f_s^p(t) \leq$*

$f_s^v(t), f_d^p(t) \geq f_d^v(t)$. Furthermore, the equilibrium is a Nash equilibrium in every subgame that is reached. The exchange is completed immediately in a finite number of steps. Furthermore, $\exists \underline{\epsilon}^* > 0$ s.t. $\forall \underline{\epsilon}$ where $0 < \underline{\epsilon} \leq \underline{\epsilon}^*$, the exchange has a minimal number of steps out of all Nash equilibrium exchanges that remain equilibria in subgames that are reached and lead to immediate completion.

Theorem 5.8 *The Nash equilibrium of Theorem 5.7 is not subgame perfect.*

Both with countable and uncountable goods, there is a trivial method to make the strategies that were presented earlier subgame perfect. This can be done by defining the strategies as before, but specifying that if the current $t_n > 0$, the agents will exit. Obviously the supplier is best off by delivering no further if the demander pays no further, and vice versa. Thus these strategies are best responses to each other in subgames where $t_n > 0$. But earlier it was shown that the original strategies formed an equilibrium in each subgame along the path of play starting at $t_0 = 0$. Thus the new strategies form a subgame perfect equilibrium. The problem with this new subgame perfect equilibrium is that it does not lead to completion of the exchange from any point where $t_n > 0$.

If the conditions on the discount functions (in Theorem 5.5 or 5.7) do not hold, the outcomes vary. A supplier wants to carry out the exchange at a time t when its $f_s^p(t)$ is high and $f_s^v(t)$ is low. The demander prefers to exchange when $f_d^p(t)$ is low and $f_d^v(t)$ is high. These times may not coincide. The exact forms of the discount functions define whether the exchange can be carried out in equilibrium immediately, by slightly postponing (different moves in the exchange may be postponed different amounts), or only by postponing infinitely.

5.5.2 Role of Time in Non-Isolated Exchange

Next, it is shown that time discounts reduce the advantages of non-isolated exchange in some cases. We assume that the *current value* of each agent's defection cost does not change—which seems realistic. If an agent discounts payments, this means that its *absolute value* of the defection cost increases with time. The following theorem states that with certain types of discount functions, the exchange cannot proceed outside of the region of isolated exchange (I in Fig. 5.5) without being delayed. The result that the discount factors on payments need to reach zero usually means that the delay is infinitely long. Thus, in such settings, taking advantage of the defection penalties of non-isolated exchange—by moving into region N in Fig. 5.5—to facilitate safe exchange is usually infeasible. Intuitively, an agent wants to postpone a negative net benefit into the future where it is heavily discounted. For the following theorem, recall from Section 5.4 that the definitions of $p^{max'}(x)$ and $p^{min'}(x)$ take the defection penalties into account.

Theorem 5.9 *If* $\lim_{t\to\infty} f_s^v(t) = 0$ *in any subgame where* $p^{max}(x_n) < p_n \leq p^{max'}(x_n)$*, and* $\forall t \geq t_n, f_s^p(t) \leq f_s^v(t)$*, there is no Nash equilibrium that remains an equilibrium in every subgame that is reached and results in completing the exchange before (supplier's) delays have caused* $f_s^v(t) = f_s^p(t) = 0$*. Similarly, if* $\lim_{t\to\infty} f_d^p(t) = 0$ *in any subgame where* $p^{min'}(x_n) \leq p_n < p^{min}(x_n)$*, and* $\forall t \geq t_n, f_d^p(t) \geq f_d^v(t)$*, there is no Nash equilibrium that remains an equilibrium in every subgame that is reached and results in completing the exchange before (demander's) delays have caused* $f_d^p(t) = f_d^v(t) = 0$*.*

The conditions $\lim_{t\to\infty} f_d^p(t) = 0$ and $f_s^p(t) \leq f_s^v(t)$ are almost always true in practise. The condition $f_d^p(t) \geq f_d^v(t)$ is true if the demander needs the goods urgently. The supplier's discount function for its goods need not approach 0 however. Its cost of producing goods (discounted to present) may not even decrease with the production date. This may sometimes allow the demander to facilitate exchange by safely over-paying and moving into the upper region N in Figure 5.5 because the conditions of the negative result (Theorem 5.9) are not fulfilled.

5.5.3 Deadlines and Lateness Penalty Schedules

The negative result (Theorem 5.9) stems from not considering infinite postponing a violation of the contract. This can be changed by specifying *deadlines* or *lateness penalty schedules* for the agents in the contract. Note that these do not have to be externally enforced. If the contract is not abided by (e.g. deadlines not honored or lateness penalties not paid), the defecting agent will suffer the defection penalty (c_s^{def} or c_d^{def}) due to how its defection will affect its future contracts—e.g. through degraded reputation. So, strictly speaking a contract matters only in non-isolated exchange, and therefore forcing timely exchange by deadlines or lateness penalties is possible only in such cases. This highlights the value of Theorems 5.5 and 5.7 for isolated exchange that guarantee that immediate exchange is an equilibrium and does not need to be forced. Even in non-isolated exchange, deadlines and lateness penalties are meaningful only as long as abiding by the deadline or paying the lateness penalty is less expensive than suffering the defection penalty. Lateness penalty schedules are preferable to strict deadlines because they are less risky for the agent who is potentially subject to them, but the other agent can still tailor the lateness penalty schedule to motivate the former to move immediately.

Example. Enabling exchange by an unenforced deadline. Let $v_s(x) = \frac{7}{10}x$, $v_d(x) = \frac{13}{10}x$, $c_s^{def} = c_d^{def} = \frac{1}{4}$, $f_s^v(t) = f_d^v(t) = 1$, and $f_s^p(t) = f_d^p(t) = \frac{100}{100+t}$. Now let us analyze a contract that specifies $p^{contr} = 1$ and an unenforced deadline at $t = 1$. The supplier prefers to execute the exchange as early as possible because $f_s^v(t) - f_s^p(t)$ strictly increases with time. The demander exchanges as late as possible because $f_d^p(t) - f_d^v(t)$ strictly decreases with time. Knowing this, the supplier loses nothing by also postponing. Thus the exchange takes place right before the deadline at $t = 1$ (or not at all). It turns out that the strategies S_s and S_d (at time $t = 1$) form a Nash equilibrium that leads to completion of the exchange and remains an equilibrium in every subgame that is reached—by just substituting the following equations for

$p^{max'}_{deadlined}$ and $p^{min'}_{deadlined}$ in place of p^{max} and p^{min} correspondingly in the strategies. As earlier, the intuitive condition for the supplier to cooperate is

$$f_s^p(1)[p^{contr} - p(x)] \quad \geq \quad f_s^v(1)[v_s(1) - v_s(x)] - c_s^{def}$$

from which we get

$$\begin{aligned} p^{max'}_{deadlined}(x) &= \frac{f_s^p(1)p^{contr} - f_s^v(1)v_s(1) + f_s^v(1)v_s(x) + c_s^{def}}{f_s^p(1)} \\ &= p^{contr} - \frac{101}{100}[v_s(1) - v_s(x) - c_s^{def}] \\ &= 0.707x + 0.5455 \end{aligned}$$

Similarly, the intuitive condition for the demander to cooperate is

$$f_d^v(1)[v_d(1) - v_d(x)] \quad \geq \quad f_d^p(1)[p^{contr} - p(x)] - c_d^{def}$$

from which we get

$$\begin{aligned} p^{min'}_{deadlined}(x) &= \frac{f_d^p(1)p^{contr} - f_d^v(1)v_d(1) + f_d^v(1)v_d(x) - c_d^{def}}{f_d^p(1)} \\ &= p^{contr} - \frac{101}{100}[v_d(1) - v_d(x) + c_d^{def}] \\ &= 1.313x - 0.5655 \end{aligned}$$

The equilibrium path is $(0,0), (0.4307, 0.5455), (0.8462, 0.8500), (1,1)$. The supplier's payoff from the exchange is $f_s^p(1)p^{contr} - f_s^v(1)v_s(1) = \frac{100}{101} \cdot 1 - 1 \cdot \frac{7}{10} \approx 0.29$, and the demander's payoff is $f_d^v(1)v_d(1) - f_d^p(1)p^{contr} = 1 \cdot \frac{13}{10} - \frac{100}{101} \cdot 1 \approx 0.31$. Note that the exchange is possible in equilibrium although $p^{max}(x)$ is not constant at the end of the exchange.

5.6 Delivery sequencing

So far we have discussed exchanges in which the delivering order of the goods is fixed beforehand.[6] In this section we analyze an exchange where partial deliveries

[6] An analogous situation occurs when the order of delivery does not matter because all goods are equivalent.

(individual countable goods, or atomic chunks of countable or uncountable goods) can be delivered in any order, as long as all of them get delivered. The problem differs significantly based on whether the chunks to sequence are independent or not. These cases are analyzed in Sections 5.6.1 and 5.6.2 respectively.

5.6.1 Sequencing Independent Deliveries

A *chunk sequencing algorithm* is presented which is an integral part of the exchange manager module of a self-interested agent, Fig. 5.1. It is assumed (this is relaxed in Section 5.6.2) that the demander's added value from one chunk does not depend on the other chunks delivered so far, and that the supplier's cost for delivering a chunk does not depend on other chunks delivered earlier. This enables us to associate each chunk c with two values, Δp_c^{max} and Δp_c^{min}, that fully characterize how much the maximum and the minimum cumulative payments change as c is delivered.

For example, an agent could make a contract to carry out a number of matrix multiplications. Multiplying two matrices neither facilitates nor hinders multiplying some other two, so the chunks are independent with respect to the supplier. The chunks are truly independent if they are independent with respect to the demander also—based on the uses of the multiplication results.

Call a delivery sequence safe if $min(p^{max}(x), p^{contr}) \geq p^{min}(x')$ for all consecutive x and x'. We provide a fast greedy algorithm that finds a safe ordering if one exists. The algorithm takes six inputs: a set of chunks C, a vector of Δp_c^{max} values, a vector of Δp_c^{min} values, the contract price p^{contr}, and the defection penalties ($c_s^{def} = c_d^{def} = 0$ in the case of isolated exchange).

Algorithm 5.1 SEQ-CHUNKS$(C,\ \boldsymbol{\Delta p^{max}},\ \boldsymbol{\Delta p^{min}},\ p^{contr},\ c_s^{def},\ c_d^{def})$

1. $p_{init}^{max} = p^{contr} + c_s^{def}, p_{init}^{min} = p^{contr} - c_d^{def}$.

2. *For every* c *in* C *do /* Sets bounds for* p *at* $x = 0$ **/*

   $$p_{init}^{max} = p_{init}^{max} - \Delta p_c^{max}, p_{init}^{min} = p_{init}^{min} - \Delta p_c^{min}.$$

3. *If* $p_{init}^{max} < 0$ *or* $p_{init}^{min} > 0$ *return "NO SOLUTION".*

4. *Divide* C *into two sets* POS *and* NEG *s.t.*

   $POS = \{c \in C | \Delta p_c^{max} - \Delta p_c^{min} \geq 0\}$ *and*

   $NEG = \{c \in C | \Delta p_c^{max} - \Delta p_c^{min} < 0\}$.

5. $p^{max} = p_{init}^{max}, p^{min} = p_{init}^{min}, n_p = |POS|, n_n = |NEG|$.

6. *For* $i = 1$ *to* n_p

   $FEASIBLES = \{c \in POS | p^{min} + \Delta p_c^{min} \leq p^{max}\}$.

   If $FEASIBLES = \emptyset$ *return "NO SOLUTION".*

   $c^* = \arg\max_{c \in FEASIBLES} \Delta p_c^{max} - \Delta p_c^{min}$.

   $chunk[i] = c^*$.

   $p^{max} = p^{max} + \Delta p_{c^*}^{max},\ p^{min} = p^{min} + \Delta p_{c^*}^{min}$.

   $POS = POS - \{c^*\}$.

7. $p^{max} = p^{contr} + c_s^{def}, p^{min} = p^{contr} - c_d^{def}$.

8. *For* $i = n_n + n_p$ *down to* $n_p + 1$

   $FEASIBLES = \{c \in NEG | p^{min} \leq p^{max} - \Delta p_c^{max}\}$.

   If $FEASIBLES = \emptyset$ *return "NO SOLUTION".*

   $c^* = \arg\max_{c \in FEASIBLES} \Delta p_c^{min} - \Delta p_c^{max}$.

   $chunk[i] = c^*$.

   $p^{max} = p^{max} - \Delta p_{c^*}^{max},\ p^{min} = p^{min} - \Delta p_{c^*}^{min}$.

   $NEG = NEG - \{c^*\}$.

9. *Return the ordered vector "chunk". First chunk to be delivered is in "chunk*$[1]$*".*

Step 6 of the algorithm sequences the chunks with positive $\Delta p_c^{max} - \Delta p_c^{min}$ in a forward passing greedy manner to try to increase p^{max} as much as possible while increasing p^{min} as little as possible. Intuitively, the algorithm tries to maximize the range of possible safe prices at each x. Step 7 just computes p^{max} and p^{min} at the end of the whole sequence of chunks. Step 8 makes a greedy backward pass. It tries to allocate the chunks with negative $\Delta p_c^{max} - \Delta p_c^{min}$ so as to use as little as possible of the beneficial difference $\Delta p_c^{max} - \Delta p_c^{min}$ in the end of the sequence. Intuitively, this difference is saved for the middle of the sequence, from where it has time to affect more chunks (lying later in the sequence).

To solve our sequencing problem, we tried several greedy algorithms starting with the intuitive ones. Most of them do not guarantee that a safe sequence is found even if one exists. For example, the algorithms that greedily pass only forward and maximize $\Delta p_c^{max} - \Delta p_c^{min}$ or minimize Δp_c^{min} can be defeated by counterexamples with just two chunks, i.e. there exist such simple problem instances where the algorithms do not find a safe sequence even though one exists. On the other hand, our algorithm is sound and complete:

Theorem 5.10 *Algorithm 5.1 finds a safe ordering if one exists, and returns "NO SOLUTION" otherwise. It always terminates in $O(|C|^2)$ time.*

Sometimes the division of the exchange into chunks is not externally fixed but can be decided by the agents, e.g. at contract time. For this reason, an *exchange partitioning (chunking) algorithm* should be added to the architecture of the exchange manager module of a self-interested agent, Fig. 5.1. Exchange chunking can be done *top down* by generating a chunking and then testing its safety by running Algorithm 5.1. If it is not safe, the chunking can be refined by splitting chunks further. Splitting is monotonic in the sense that no split can make a safe exchange unsafe. Therefore this splitting algorithm does not need to backtrack. The top down method can be used for uncountable goods also. The minus side of the approach is the need to guess the splits. If they are guessed badly, possibly many more chunks are generated than are necessary to enable safe exchange. A *bottom up* approach for chunking is to sequence the smallest possible atomic chunks using Algorithm 5.1. Next, the agents may be able to deliver multiple atomic chunks at any one step of the exchange—via strategies S_s and S_d, or S_s^{timed} and S_d^{timed}—without changing the order of the chunks. Bottom up chunking requires no guessing of splits but it can be computationally complex if the number of smallest possible chunks is large. It cannot be applied to uncountable goods because the number of smallest possible chunks is infinite.

5.6.2 Sequencing Interdependent Deliveries

Sometimes partial deliveries are interdependent. The value of a chunk may depend on which chunks have been delivered before it. For example in manufacturing, the first products can be thought of as more costly than subsequent ones because the fixed costs (e.g. rent, acquired equipment) can be associated with the earlier products. On a more operational level, a grinding machine can be thought of as producing its output items less and less expensively, because the setup cost for that type of items can be spread among more items. Similarly, a data retrieval agent may incur large costs in searching for certain information. Once the information is found, subsequent searches of related information are less expensive. The demander may also value a chunk differently depending on the other chunks delivered so far. In the TRACONET system [Sandholm, 1993], the chunks (transport tasks) were interdependent for both the supplier and the demander. Transporting a parcel often affects the marginal cost of transporting others. For example, a vehicle may be able to carry two parcels to adjacent locations, thus reducing the marginal cost of both tasks. Conversely, one parcel may fill up the vehicle so that another task must be handled by a more costly vehicle. Some contracts involved multiple tasks. So if the safe exchange mechanisms of this dissertation had been used, sequencing of interdependent chunks would have been required.

In general, interdependent goods cannot be sequenced in polynomial time in the number of chunks if it is required that a safe solution is found if one exists. Just representing the problem requires $\Theta(2^{|C|})$ space because for each set of chunks in the power set of all chunks, p^{max} and p^{min} have to be represented—and in the worst case this information cannot be compressed. Nevertheless, if the number of chunks per contract is small—as in TRACONET—exponential search among sequences of chunks is viable. In such cases the advantages of safe unenforced exchange often outweigh the extra computational load. Furthermore, special cases of the problem may be solvable in polynomial time, e.g. the case of independent chunks discussed earlier.

5.7 Renegotiation Avoidance

After an irrevocable delivery or payment, the agent that gained from it may want to renegotiate the contract. For example after the first partial delivery, the demander may want to renegotiate the contract for a lower price. The demander knows that the original contract price was safe for the supplier, so now that the supplier has already "lost" the first delivery, the supplier should be willing to carry out the rest of the exchange at a lower price. On the other hand, if the contract execution method of this chapter is used, the supplier knows that any point in the exchange is safe for the demander. Therefore, if the supplier can commit to not renegotiating, the demander is motivated to follow the original contract and to not vanish.

Renegotiation is more likely in unsafe exchange [Lax and Sebenius, 1981, Raiffa, 1982]. For example, when an international company initiates a mining venture in a developing country, it has to invest most of the capital up-front. This unsafe move motivates the developing country to renegotiate the conditions of the mining venture (profit division etc.). Due to expropriation risk the company cannot avoid renegotiation.

5.8 Chapter Summary

This chapter presented a method for carrying out mutual exchanges among self-motivated agents without third party enforcement. Larger exchanges were split into smaller parts so that at no point was either agent motivated to defect (in equilibrium). The maximum size delivery that the supplier can safely make at any point in the exchange was shown as well as the maximum amount that the demander can safely pay. The possibility of safe exchange depends on the demander agent's and the supplier agent's value functions for the goods of the contract. Safe exchange is enhanced if the supplier incurs most of its cost from the early portion of the exchange, while the possibility that the demander acquires most of its value already from the early parts hinders safe exchange. Luckily, in practise, the supplier usually does have setup costs and the demander often acquires most of the value from the completion of the exchange.

Isolated safe exchange can be carried out entirely only if the supplier can deliver the last part without cost. With uncountable goods it can often be carried out arbitrarily close to completion even if this is not the case. Considering defection's adverse effect on future negotiations often enables completing the exchange.

Under the presented conditions on their discount functions, agents are motivated to carry out isolated exchanges immediately. Time discounts reduce the viability of taking advantage of non-isolated exchange. In such cases, immediate moves can be stimulated by deadlines or lateness penalties which themselves need not be enforced.

Some domains allow goods to be delivered in different chunks (batches) and those chunks to be delivered in different sequences. A top-down and a bottom-up chunking algorithm was presented and their relative merits discussed. A non-trivial quadratic algorithm was presented for sequencing independent chunks. It was proven to find a safe sequence if one exists. The general problem cannot be solved in polynomial time in the number of chunks when chunks are interdependent. Future research should include designing polynomial algorithms for other special cases of this problem besides the case of independent chunks. Finally, we showed that safe exchange helps prevent unfair renegotiation.

CHAPTER 6

CONCLUSIONS

This dissertation studied interactions of self-interested agents whose rationality is bounded by limited computation. The developed methods are mainly targeted to inherently distributed combinatorial problems—e.g. resource and task allocation and multiagent planning and scheduling—in situations where agents may have different goals, and each agent is trying to maximize its own good without concern of the global good. Because agents cannot solve the underlying combinatorial problems optimally, they cannot enumerate or evaluate all alternatives in advance as is commonly assumed in game theory. The agents have to trade solution quality off with the amount of computation. The developed methods enable open negotiation systems that operate with separately designed self-interested agents in combinatorial real world domains such as distributed vehicle routing, multi-enterprise manufacturing planning and scheduling, and meeting scheduling.

The approach is quantitative. The issues were theoretically analyzed and experiments were carried out on a five agent $\mathcal{NP}$-complete distributed vehicle routing problem when appropriate. The experiments were run on large-scale problem instances using real one week vehicle and delivery order data that was collected from five dispatch centers.

Our normative approach borrows solution concepts from game theory. It allows agents to choose their own strategies that are best in terms of self-interest: the agents are not unrealistically assumed to abide to centrally imposed strategies. This work contributes to three broad areas within multiagent systems research: automated contracting, coalition formation, and contract execution.

In our automated contracting setting, agents iteratively reallocate tasks among themselves while improving the task allocation with each contract. Previous contract net implementations assumed that agents handle tasks for each other for free, while in Section 3.1 we extended the model to self-interested agents that handle other agents' tasks only if paid. Previous contract net work has used domain specific heuristic methods for making bidding and awarding decisions. We introduced a domain independent formal model for making these decisions: bidding and awarding are entirely based on local marginal cost calculations. This allows contracts that are guaranteed to be individually rational for both parties, and which therefore provably enhance social welfare. In the vehicle routing domain—and many other combinatorial domains—the marginal cost calculation requires solving two $\mathcal{NP}$-complete problems, which is intractable. Therefore, approximation algorithms were used for calculating marginal costs. The consequences of over and underestimating were discussed, and it was shown that profitability is really defined with respect to the actually computed solutions, not the optimal solutions which are unobtainable. The resulting negotiation scheme is an anytime task reallocation algorithm that can be terminated at any time, and is guaranteed to have a feasible solution for each agent that is no worse than the agent's initial solution: participation is individually rational. It was experimentally shown via a truly distributed asynchronous implementation that the technology has a possibility to scale to large scale problem instances. In this part of the contracting work we used a market assumption: agents accept any individual rational contracts. This differs from game theoretic assumptions in that agents do not lie about tasks or marginal costs, and they do not look ahead into the contracting process. On the other hand, in other parts of the contracting chapter and the other chapters, agents were not assumed to act sincerely. They could act strategically based on self-interest in the vein of game theory.

In Section 3.2, a decommitment mechanism was presented for automated contracting protocols that allows the agents to accommodate future events more profitably than traditional full commitment contracts. Each contract specifies a decommitment penalty for both agents involved. These penalties can be negotiated on a per contract basis, and they can differ between the contract parties. To decommit, an agent just pays the penalty to the other agent. Several reasons were presented why this protocol is better suited for computerized contracting settings than contingency contracts. The game-theoretic analysis of the decommitting games handled the possibility that agents decommit manipulatively: an agent tries to avoid the decommitment penalty in case it believes that there is a high probability that it will be freed from the contract's obligations due to the other agent decommitting. This analysis also serves as a normative tool for agents to decide which contracts they should accept based on individual rationality. The bargaining associated with choosing *ex ante* among these individually rational contracts was not addressed.

In the presented games, leveled commitment contracts are a superset of full commitment ones because the latter can be emulated by setting the decommitment penalties sufficiently high. Therefore, full commitment protocols cannot be better than leveled commitment ones in the sense of Pareto efficiency or social welfare. Neither can they enable a deal that is impossible—based on individual rationality—via a leveled commitment contract. In games where no outside offers become void between the contracting and the decommitting time (game types DOP, SEQD, SIMUDBP, and SIMUDNP), there are instances where the new protocol enables contracts that are impossible (not individually rational to the agents) using full commitment contracts (Theorems 3.1, 3.8, 3.10, and 3.12). Also, in these game types, leveled commitment contracts improve each agent's expected payoff over any full commitment contract as long as there is some chance that the contractor's outside offer is lower than the expected value of the contractee's, or some chance that the contractee's outside offer is higher than the expected value of the contractor's (Theorems 3.2, 3.9, 3.11, and 3.13). Obviously one can also construct game instances where the agents' outside offers are so profitable that no contract—even a leveled commitment one—is individually rational to the agents. In the game (COBV) where one agent loses an outstanding outside offer by agreeing to a contract, a leveled commitment contract can enable a deal or Pareto improve a deal over a full commitment contract only if that agent's fallback payoff is sufficiently high (Theorems 3.4 and 3.5).

In the games (DOP and COBV) where only one agent's future outside offer involves uncertainty, the agent with a certain outside offer prefers to not decommit. In these games, an agent's payoff to a contract is unaffected by the other agent's beliefs (Theorems 3.3 and 3.7). It follows that an agent need not counterspeculate its negotiation partner's beliefs, and that an agent cannot incur a loss due to the other agent's erroneous beliefs.[1] On the other hand, in the games (SEQD, SIMUDBP, and SIMUDNP) where both agents' future outside offers involve uncertainty, an agent's payoff to a contract may depend on the negotiation partner's possibly biased beliefs because they affect the other partner's decommitting decision.

In Section 3.3, different stages of commitment were discussed. It was shown that commitment has to occur at some stage, but this stage can be made a negotiated item: it need not be fixed in the protocol. Section 3.4 presented an example protocol that allows this in addition to allowing other types of flexibility: counterproposing, alternative offers, etc.

Section 3.5 showed that classical contracts of one task at a time are insufficient for reaching the globally optimal task allocation because there are local optima—assuming that agents only make individually rational contracts. Three new contract types were presented to overcome this problem: clustering, swaps, and multiagent contracts. It was shown via counterexamples that these are insufficient when applied

[1] However, the agents may perceive a contract to be individually rational to both although it really is not (Theorem 3.6). This error cannot hurt the agent that has unbiased beliefs (Corollary 3.1).

alone, in pairs, or even when all three are used separately. However, these three can be combined into a new atomic contract type: CSM-contract. Theorem 3.14 showed that if the agents can pick any CSM-contracts, any hill-climbing algorithm finds the global task allocation optimum in a finite number of steps without backtracking. This means that from the perspectives of social welfare maximization and of individual rationality (not necessarily from the perspective of local payoff maximization), agents can accept individually rational contracts as they are offered. They need not wait for more profitable ones, and the need not worry that a current contract may make a more profitable future contract unprofitable. Neither do they need to accept contracts that are not individually rational in anticipation of future contracts that make the combination beneficial. This result is powerful for small problem instances, but for large instances, the number of contracts made before the globally optimal task allocation is reached may be too great to be carried out in practice—albeit finite. Therefore, for the larger problem instances, the guaranteed anytime property is more important: the negotiations can be terminated at any time, and each agent will have a feasible solution that is no worse than its initial solution.

Section 3.6.1 discussed the contracting implications of agents' computational limitations. First, several real-time pressures were identified that are specific to contracting settings. It was argued that an agent therefore needs to decide which task sets to focus its deliberation on, how much, and in what order. Next, the issue of engaging in multiple simultaneous negotiations was discussed. If an agent bids or awards while some of its older bids are pending, the current bid or award may turn out unbeneficial in the world that materializes by some combination of acceptances of pending bids. Marginal cost calculations were formalized that pessimistically assume that the worst combination gets accepted. With this type of calculation, an agent's local cost is guaranteed to decrease monotonically. An opportunistic approximation was also presented that assumes that none of the pending bids get accepted—but a feasibility check was used to ensure that a feasible solution can be reached locally even if all pending bids get accepted. This approximation is faster to compute because the exponentially many alternative worlds need not be considered, but an agent may occasionally make unbeneficial contracts. So, in general, there is a tradeoff between computational complexity and monetary risk.

Alternatively, an agent can refuse to bid or award while previous bids are pending. In this case, the marginal cost can be calculated risk free without added complexity because alternative future worlds need not be considered. On the other hand, such an agent cannot participate in simultaneous negotiations, and may thus lose opportunities. On the other hand, this types of agents can have each negotiation take less time because local deliberation is faster. The distributed vehicle routing experiments showed that bidding and awarding while previous bids pend (using the opportunistic pricing) indeed leads to some unbeneficial contracts in practice. The experiments also confirm the result that bidding and awarding only when previous bids do not pend leads to monotonic decrease of the local cost. Which of these methods is best in terms of decreasing local cost most depends on the problem, the problem instances, the chosen duration of the valid bidding window, and even the interleaving of the asynchronous messages that happens to occur. The experiments were inconclusive regarding this question. Section 3.6.3 concluded the discussion on contracting implications of limited computation by presenting tradeoffs that an agent faces in deciding whether to bid or award early on, or to wait.

Section 3.7 discussed distributed asynchronous implementation of automated contracting. First, message congestion and agent saturation were addressed. The classical solution approaches, focused addressing and audience restrictions, are not viable among self-interested agents: there is an inherent tragedy of the commons because agents do not care about the saturation of others. Three protocol related solutions were discussed: use-based communication charges, mutual monitoring, and processing fees. However, the problem can be solved without these by incorporating certain policies into each agent's local strategy. These policies are in concert with each agent's self-interest: they need not be externally imposed. The local strategies that completely solved the congestion and saturation problems in our experiments included reading incoming messages in batches and ignoring announcements that were older than a time limit. As the second general issue of distributed implementation, Section 3.7.2 presented a negotiation termination method for iterative task reallocation contracting. The presented method terminates the negotiations when a local optimum has been reached with respect to a—possibly restricted—set of clustering contracts. It does not assume common knowledge of tasks. The method could be extended to swaps and multiagent contracts. On the other hand, the method is not guaranteed to work among agents that are not cooperative or agents that cannot exactly compute marginal costs. Finally, as the third general issue of distributed implementation, Section 3.7.3 compared reply-based and timeout-based contracting protocols. The former are easier to implement in an asynchronous manner, but only the latter are viable among self-interested agents.

Chapter 4 analyzed coalition formation among bounded rational agents. The agents coordinate their computational and real world actions within each coalition, but not across coalitions. Therefore, computation is centralized within each coalition, but distributed across coalitions. We analyzed collusion of bounded rational agents using a quantitative model of the agents' algorithms performance profiles and the computation unit cost. From this formal model, the social welfare maximizing coalition structure can always be determined when the performance profiles and computation unit cost are known. However, for example when agents are sent to execute at a remote host, the computation unit cost is not necessarily known in advance. To attack this problem, a sufficient condition (Theorem 4.1) on the performance profiles was presented that guarantees that any two coalitions are best off merging *for any computation unit cost*, i.e. for any execution platform. It follows that the best coalition structure would be the grand coalition. Next it was shown that the presented condition is not a necessary condition in general (Theorem 4.2) but is one if the performance profiles exhibit diminishing returns to added computations (Theorem 4.3). This is almost always the case with design-to-time algorithms, and often anytime algorithms exhibit this general character also. Finally, a sufficient condition on the performance profiles was presented that guarantees that all agents are best off operating individually for any computation unit cost (Theorem 4.4).

The stability of the coalition structure was analyzed in terms of the *core* solution concept. The structure is considered stable if no agent, subgroup, or the whole group of agents can increase their payoff by breaking off from the coalition structure and forming a new coalition. From the formal model of computational limitations, the stability of the coalition structure can always be determined. There are games that have stable coalition structures for both rational and BR agents, for one but not the other, and for neither. Theorems relating the shapes of the performance profiles and the computation unit cost to stability were also presented. First, if these computation limitations are such that the agents are best off operating individually, then that coalition structure is stable (Theorem 4.5). Second, necessary and sufficient conditions on the coalition's bounded rational values were presented that guarantee stability if the mentioned beneficial merging property (bounded rational superadditivity) holds (Theorem 4.7), or more generally, if the best coalition structure is the grand coalition (Theorem 4.6). Finally, sufficient conditions were presented on the performance profiles that guarantee stability when the beneficial merging property holds (Theorem 4.9), or more generally, if the best coalition structure is the grand coalition (Theorem 4.8).

A domain classification was also presented for bounded rational agents (Figure 4.2). It was related to two existing domain classifications for fully rational agents:

the traditional one from game theory, and the one by Rosenschein and Zlotkin. The domain classification carries with it information about the optimal coalition structure and its stability. It also incorporates domain classes where the value of a coalition is affected by the actions of nonmembers. Such games occur if agents have different optimization algorithms or if there are domain solution interactions—unlike in the vehicle routing problem and many other real world problems. Such games require different solution concept as was discussed in Section 4.5.

Coalitions were experimentally analyzed using the real-world data from the distributed vehicle routing problem. A local routing algorithm that was based on iterative refinement was used. The experiments show that the computational limitations of the agents significantly impact the coalition structure that should form as well as its stability. Although the beneficial merging property holds for rational agents in almost all domains, it was surprisingly all but obvious in practice among bounded rational agents. None of the vehicle routing games of our experiments exhibited this property for bounded rational agents. The optimal coalition structure for bounded rational agents varied although rational agents should always form the grand coalition. Section 4.2 developed conditions on the performance profiles that guarantee that the beneficial merging property holds for bounded rational agents. It also discussed a separate solving approach—based on a problem decomposition step—that guarantees that the base algorithm fulfills those conditions. With our reasonable deterministic iterative refinement algorithm, these conditions were—somewhat surprisingly—never met. The real desideratum is not necessarily to generate algorithms that guarantee beneficial merging (and thus the superiority of the grand coalition over other coalition structures), but algorithms that provide the highest social welfare (for the best coalition structure, which need not be the grand coalition). Sometimes these goals conflict.

In the experimental games where the agents were best off separately, the coalition structures were stable as our theory predicts. Even in games were subgroups were not necessarily best off by splitting up, the coalition structures were often stable—especially for large computation unit costs.

The experiments suggest that often with superlinear iterative refinement steps, low computation unit costs promote large coalitions while high computation unit costs suggests smaller ones. A plausible explanation for this phenomenon was presented.

Another interesting observation is that the presented normative theory prescribes the bounded rational agents to choose coalition structures that agree closely with what human agents would select. The best coalition structures among bounded rational agents mostly agreed with our intuitions of what coalitions should form based on strategic domain specific considerations such as adjacency of the dispatch centers and the combinability of their loads. On the other hand, these coalition structures differ significantly from those which rational agents would choose.

Chapter 5 of the dissertation presented a method for carrying out exchanges without external enforcement. Exchanges were split into parts so that at no point was either agent motivated to defect. The maximum size delivery that the supplier can safely make at any point in the exchange was shown as well as the maximum amount that the demander can safely pay. The possibility of safe exchange depends on the demander agent's and the supplier agent's value functions for the goods of the contract. Safe exchange is enhanced if the supplier incurs most of its cost from the early portion of the exchange, while the possibility that the demander acquires most of its value already from the early parts hinders safe exchange. Luckily, in practise, the supplier usually does have setup costs and the demander often acquires most of the value from the completion of the exchange.

Isolated safe exchange can be carried out entirely only if the supplier can deliver the last part without cost. Whenever isolated safe exchange is possible, our strategies handle it in subgame perfect Nash equilibrium in a minimal number of steps

(Theorems 5.1 and 5.2). With uncountable goods, safe exchange can often be carried out arbitrarily close to completion even if the last part cannot be delivered without cost. Whenever such exchange is possible, our strategies handle it in subgame perfect Nash equilibrium in a minimal number of steps (Theorems 5.3 and 5.4). Considering defection's adverse effect on future negotiations often enables completing the exchange (Section 5.4).

Under the presented conditions on their discount functions, agents are provably motivated to carry out isolated exchanges immediately. Using the presented strategies, such exchanges take place in Nash equilibrium in the minimal number of steps (Theorems 5.5 and 5.7), and can be made subgame perfect. Time discounts reduce the viability of taking advantage of non-isolated exchange (Theorem 5.9). In such cases, immediate moves can be stimulated by deadlines or lateness penalties which themselves need not be enforced (Section 5.5.3).

Some domains allow goods to be delivered in different chunks (batches) and those chunks to be delivered in different sequences. A top-down and a bottom-up chunking algorithm was presented and their relative merits discussed. A non-trivial quadratic algorithm (Algorithm 5.1) was presented for sequencing independent chunks. It was proven to find a safe sequence if one exists (Theorem 5.10). The general problem cannot be solved in polynomial time in the number of chunks when chunks are interdependent. Finally, we showed that safe exchange helps prevent unfair renegotiation.

6.1 Future Research

Future research in automated contracting includes extending the marginal cost based implementation to a game theoretic setting where agents may lie about their marginal costs and look ahead into the contracting process. This should include normative methods for local deliberation scheduling. Explicit strategies for striking the optimal tradeoffs between marginal cost deliberation and monetary risk taking should be devised for the case where the agent engages in multiple negotiations simultaneously and for the case where it does not.

A contracting implementation using the whole set of CSM-contracts, and scheduling policies for applying those contracts are desirable. It would also be useful to implement the presented negotiation termination protocol for clustering contracts, and extend it to swaps and multiagent contracts. Termination protocols that are guaranteed to work among self-interested and computationally limited agents should also be devised. It is also desirable to develop an implementation of contracting that is purely based on deadlines, not on reply messages. Finally, it would be interesting to apply the technologies developed in this dissertation to other negotiation domains.

There are a host of interesting open research issues in leveled commitment contracts. One should study more closely the best pace to increase the decommitment penalties with time or with occurring events. A normative theory relating the performance profiles of the algorithms of bounded rational agents to the issue of leveled commitment is also desirable. It would also be interesting to study the beneficiality of leveled commitment contracts from the perspective of the society of agents—not only the two agents making the contract. Finding the optimal decommitment penalties for these two perspectives would also be desirable, and the bargaining process of choosing among individually rational contracts should be addressed. Finally, the relationship between leveled commitment contracting and explicit CSM-contracts should be studied in more detail.

Our model of bounded rationality in coalition formation was based on costly computation resources. Future work includes analyzing another model, where each agent has a fixed free CPU and no more CPU time can be bought. If the domain cost increases with real time due to a dynamic environment, such agents with *bounded computational capabilities* are often best off by distributing the computation. In the *costly computation model* of the coalition formation chapter of this dissertation, it was best to allocate each coalition's computation to a single agent. The models are equivalent if the domain cost increases linearly with real time and distribution does not speed up computation.

Extensions to coalition formation include generalizing the methods to agents with different performance profiles, probabilistic performance profiles, and anytime algorithms where the performance profiles are conditioned on execution so far [Sandholm and Lesser, 1994, Zilberstein, 1993]. Agents with probabilistic performance profiles may want to reselect a coalition if the value of their original coalition is lower than expected—but sunk computation cost has already been incurred. Future research also includes studying agents that can refine solutions generated by others, and the development of interaction protocols that efficiently guide self-interested agents towards the optimal and stable (whenever possible) coalition structures—as determined by the theory developed in this dissertation.

In the area of contract execution, we looked at totally safe exchanges, where each agent knew its opponent's and its own value function, discount functions, and defection penalty (i.e. cost of making reputation worse). We explained how agents could use bounds for these if they are not exactly known. If the bounds were too far off, even possible exchanges were disabled. Often it is the case that agents can estimate a distribution for each of these, although strict bounds are not available or the bounds are too far off to allow unenforced exchange. Using these distributions the agents can take a calculated risk of making moves that are unsafe with a certain probability. This approach of using distributions is also useful to the agent in trying to model the possibility of changes in the opponent's value function, discount functions or defection penalty that happen during the exchange due to the opponent interacting in its environment—e.g. getting other offers and contracts, or receiving domain events that alter the value function.

Another approach is to try to bound one's losses by making the partial exchanges small enough so that even if the opponent defects, the loss will be within a bound. In both the probabilistic risk method and the loss bounding method there is a tradeoff between making the exchange safer by using small partial exchanges and minimizing partitioning costs (e.g. physical per part delivery costs) by using large ones. Future research should include studying unenforced exchange where such transaction costs are explicitly modeled. Finally, either a probabilistic approach or a loss bounding approach can be used to address the risk of the opponent accidentally defecting—e.g. losing contact due to a technical fault.

Future research should also include designing polynomial algorithms for other special cases of the chunk sequencing problem besides the case of independent chunks.

APPENDIX

PROOFS OF LESS INTUITION

Proof. (Theorem 3.1). Example. Let $\breve{b} = 55$ and let $f(\breve{a}) = \begin{cases} 0.01 & \text{if } 0 \le \breve{a} \le 100 \\ 0 & \text{otherwise} \end{cases}$. Now there exists no full commitment contract F that satisfies both IR constraints because that would require $\breve{b} \le \rho_F \le E[\breve{a}]$ which is impossible because $55 = \breve{b} \ge E[\breve{a}] = 50$. Let us analyze a leveled commitment contract where $\rho = 60$, $a = 10$, (and $b \ge 0$). The contractor's IR constraint is satisfied:

$$\int_{-\infty}^{\infty} f(\breve{a})[-\breve{a}]d\breve{a} \le \int_{-\infty}^{\rho-a} f(\breve{a})[-\breve{a}-a]d\breve{a} + \int_{\rho-a}^{\infty} f(\breve{a})[-\rho]d\breve{a}$$

$$\Leftrightarrow \int_{0}^{100} f(\breve{a})[-\breve{a}]d\breve{a} \le \int_{0}^{60-10} f(\breve{a})[-\breve{a}-10]d\breve{a} + \int_{60-10}^{100} f(\breve{a})[-60]d\breve{a}$$

$$\Leftrightarrow -50 \le -17.5 - 30$$

Now, $p_a = \int_{-\infty}^{\rho-a} f(\breve{a})d\breve{a} = \int_{0}^{60-10} f(\breve{a})d\breve{a} = 0.5$. Thus the contractee's IR constraint is also satisfied:

$$\breve{b} \le [1-p_a]\rho + p_a[\breve{b}+a]$$

$$\Leftrightarrow 55 \le [1-0.5]60 + 0.5[55+10]$$

$$\Leftrightarrow 55 \le 62.5$$

This completes the proof. $\square$

Proof. (Theorem 3.2). Under F, the contractor's payoff is $-\rho_F$, and the contractee's ρ_F. Now we construct a leveled commitment contract defined by ρ, a, and b. Let $a = \rho_F - \breve{b}$, (and $b \geq 0$). Choose $\rho = \rho_F$, and $a = \rho_F - \breve{b} + \epsilon$. The contractor decommits if $-\breve{a} - a > -\rho \Leftrightarrow \breve{a} < \rho - a = \breve{b} - \epsilon$. This has nonzero probability because bounded f and $\int_{-\infty}^{\breve{b}} f(\breve{a})d\breve{a} > 0$ imply $\exists \epsilon > 0$ s.t. $\int_{-\infty}^{\breve{b}-\epsilon} f(\breve{a})d\breve{a} > 0$. The contractee's expected payoff increased: it is ρ_F if the contractor does not decommit, and $\breve{b} + a = \rho_F + \epsilon > \rho_F$ if the contractor does (the latter has nonzero probability). The contractor's expected payoff also increased:

$$-\rho_F < \int_{-\infty}^{\rho-a} f(\breve{a})[-\breve{a} - a]d\breve{a} + \int_{\rho-a}^{\infty} f(\breve{a})[-\rho]d\breve{a}$$

$$\Leftrightarrow 0 < \int_{-\infty}^{\rho-a} f(\breve{a})[-\breve{a} - a + \rho_F]d\breve{a}$$

which is implied by $\int_{-\infty}^{E[\breve{b}]-\epsilon} f(\breve{a})d\breve{a} > 0$. (Because f is bounded, all of this probability mass cannot be on a single point $\breve{a} = \rho - a$ $(= E[\breve{b}] - \epsilon)$). □

Proof. (Theorem 3.3). Say the contractor's information is unbiased, i.e. $f_a(\breve{a}) = f(\breve{a})$. The contractor's expected payoff for not accepting either contract is $\int_{-\infty}^{\infty} f_a(\breve{a})[-\breve{a}]d\breve{a}$. The contractor's payoff for the full commitment contract is $-\rho_F$, and its expected payoff for the leveled commitment contract is $\int_{-\infty}^{\rho-a} f_a(\breve{a})[-\breve{a}-a]d\breve{a} + \int_{\rho-a}^{\infty} f_a(\breve{a})[-\rho]d\breve{a}$. None of these depend on the contractee's information.

Now say the contractee's information is unbiased, i.e. $f_b(\breve{a}) = f(\breve{a})$. The contractee's payoff for not accepting either contract is $\breve{b}$. Its payoff for the full commitment contract is ρ_F, and its expected payoff for the leveled commitment contract is $[1 - p_a]\rho + p_a[\breve{b} + a] = [1 - (\int_{-\infty}^{\rho-a} f_b(\breve{a})d\breve{a})]\rho + (\int_{-\infty}^{\rho-a} f_b(\breve{a})d\breve{a})[\breve{b} + a]$. None of these depend on the contractor's information. □

Proof. (Theorem 3.4). For a full commitment contract F, $\breve{b} \leq \rho_F \leq E[\breve{a}]$. No such F exists iff $\breve{b} > E[\breve{a}]$. Now say that $\breve{b} > E[\breve{a}]$, and assume—for contradiction—

that some leveled commitment contract L defined by ρ, a, and b satisfies both IR constraints. Thus,

$$\begin{aligned}
& [1-(\int_{-\infty}^{\rho-a} f(\breve{a})d\breve{a})]\rho + (\int_{-\infty}^{\rho-a} f(\breve{a})d\breve{a})[\bar{b}+a] \geq \breve{b} \\
& > E[\breve{a}] \geq \int_{-\infty}^{\rho-a} f(\breve{a})[\breve{a}+a]d\breve{a} + \int_{\rho-a}^{\infty} f(\breve{a})\rho d\breve{a} \\
\Rightarrow \quad & (\int_{-\infty}^{\rho-a} f(\breve{a})d\breve{a})[\bar{b}+a] > \int_{-\infty}^{\rho-a} f(\breve{a})[\breve{a}+a]d\breve{a} \\
\Leftrightarrow \quad & (\int_{-\infty}^{\rho-a} f(\breve{a})d\breve{a})\bar{b} > \int_{-\infty}^{\rho-a} f(\breve{a})\breve{a}d\breve{a} \\
\Leftrightarrow \quad & \bar{b} > \frac{\int_{-\infty}^{\rho-a} f(\breve{a})\breve{a}d\breve{a}}{\int_{-\infty}^{\rho-a} f(\breve{a})d\breve{a}} \geq \bar{b}
\end{aligned}$$

Contradiction. Thus no such contract L satisfies both IR constraints. □

Proof. (Theorem 3.5). Under F, the contractor's payoff is $-\rho_F$ and the contractee's ρ_F. Assume—for contradiction—that there exists a leveled commitment contract L (defined by ρ, a, and b) that increases at least one of these payoffs while not decreasing the other, i.e.

$$\begin{aligned}
& \int_{-\infty}^{\rho-a} f(\breve{a})[-\breve{a}-a]d\breve{a} + \int_{\rho-a}^{\infty} f(\breve{a})[-\rho]d\breve{a} \geq -\rho_F \\
\text{and} \quad & [1-(\int_{-\infty}^{\rho-a} f(\breve{a})d\breve{a})]\rho + (\int_{-\infty}^{\rho-a} f(\breve{a})d\breve{a})[\bar{b}+a] \geq \rho_F \\
& \text{and at least one of the above inequalities is strict.} \\
\Rightarrow \quad & \int_{-\infty}^{\rho-a} f(\breve{a})[\breve{a}+a]d\breve{a} + \int_{\rho-a}^{\infty} f(\breve{a})\rho d\breve{a} \\
& < [1-(\int_{-\infty}^{\rho-a} f(\breve{a})d\breve{a})]\rho + (\int_{-\infty}^{\rho-a} f(\breve{a})d\breve{a})[\bar{b}+a] \\
\Leftrightarrow \quad & \int_{-\infty}^{\rho-a} f(\breve{a})\breve{a}d\breve{a} < (\int_{-\infty}^{\rho-a} f(\breve{a})d\breve{a})\bar{b} \\
\Leftrightarrow \quad & \frac{\int_{-\infty}^{\rho-a} f(\breve{a})\breve{a}d\breve{a}}{\int_{-\infty}^{\rho-a} f(\breve{a})d\breve{a}} < \bar{b} \\
\Rightarrow \quad & \bar{b} < \bar{b}
\end{aligned}$$

Contradiction. Thus no such L exists. □

Proof. (Theorem 3.6). A full commitment contract can satisfy the IR constraints iff $\int_{-\infty}^{\infty} f_a(\breve{a})\breve{a}d\breve{a} \geq \breve{b}$. Now say that $f(\breve{a}) = f_a(\breve{a}) = \begin{cases} 0.01 & \text{if } 0 \leq \breve{a} \leq 100 \\ 0 & \text{otherwise} \end{cases}$ and

$\breve{b} = 55$, i.e. no full commitment contract satisfies both IR constraints. Now let $\bar{b} = 0$. It follows by Theorem 3.4 that no leveled commitment contract satisfies both IR constraints. No full commitment contract satisfies both PIR constraint because it would require $55 = \breve{b} \leq \rho_F \leq E_a[\breve{a}] = 50$. Now we show a leveled commitment contract that satisfies both PIR constraints. Let $f_b(\breve{a}) = \begin{cases} 0.01 & \text{if } 50 \leq \breve{a} \leq 150 \\ 0 & \text{otherwise} \end{cases}$. Now the contractee's PIR constraint is

$$55 \leq [1 - (\int_{-\infty}^{\rho-a} f_b(\breve{a})d\breve{a})]\rho + (\int_{-\infty}^{\rho-a} f_b(\breve{a})d\breve{a})[0 + a]$$

Substituting $\rho = 60$, $a = 10$ gives

$$55 \leq [1 - (\int_{-\infty}^{60-10} f_b(\breve{a})d\breve{a})]60 + (\int_{-\infty}^{60-10} f_b(\breve{a})d\breve{a})10 \Leftrightarrow 55 \leq 60 + 0$$

The contractor's PIR constraint is

$$-50 \leq \int_{-\infty}^{\rho-a} f_a(\breve{a})[-\breve{a} - a]d\breve{a} + \int_{\rho-a}^{\infty} f_a(\breve{a})[-\rho]d\breve{a}$$

and substituting $\rho = 60$, $a = 10$ gives

$$-50 \leq \int_{-\infty}^{60-10} f_a(\breve{a})[-\breve{a} - 10]d\breve{a} + \int_{60-10}^{\infty} f_a(\breve{a})[-60]d\breve{a} \Leftrightarrow -50 \leq -17.5 - 30$$

Thus a leveled commitment contract with $\rho = 60$, $a = 10$ satisfies both PIR constraints. $\square$

Proof. (Theorem 3.7). The contractor's expected payoff for not accepting either contract is $\int_{-\infty}^{\infty} f(\breve{a})[-\breve{a}]d\breve{a}$. The contractor's payoff for the full commitment contract is $-\rho_F$, and its expected payoff for the leveled commitment contract is $\int_{-\infty}^{\rho-a} f(\breve{a})[-\breve{a} - a]d\breve{a} + \int_{\rho-a}^{\infty} f(\breve{a})[-\rho]d\breve{a}$. None of these depend on the contractee's information.

The contractee's payoff for not accepting either contract is $\breve{b}$. Its payoff for the full commitment contract is ρ_F, and its expected payoff for the leveled commitment contract is $[1 - (\int_{-\infty}^{\rho-a} f(\breve{a})d\breve{a})]\rho + (\int_{-\infty}^{\rho-a} f(\breve{a})d\breve{a})[\bar{b} + a]$. None of these depend on the contractor's information. $\square$

Proof. (Corollary 3.1). By definition, a contract is IR for the contractor if it is preferred over the null deal. But by Theorem 3.7 the preference ordering is unaffected by the contractee's information. Similarly, by definition, a contract is IR for the contractee if it is preferred over the null deal. But by Theorem 3.7 the preference ordering is not affected by the contractor's information. □

Proof. (Theorem 3.8). Let $f(\breve{a}) = \begin{cases} \frac{1}{100} & \text{if } 0 \le \breve{a} \le 100 \\ 0 & \text{otherwise} \end{cases}$ and $g(\breve{b}) = \begin{cases} \frac{1}{110} & \text{if } 0 \le \breve{b} \le 110 \\ 0 & \text{otherwise.} \end{cases}$ Now a full commitment contract F does not satisfy both IR constraints because that would require $E[\breve{b}] \le \rho_F \le E[\breve{a}]$ which is impossible because $55 = E[\breve{b}] > E[\breve{a}] = 50$. Let us choose a leveled commitment contract where $\rho = 52.5$, $a = 30$, and $b = 20$. Now $\breve{b}^*(\rho, a, b) = \rho + \frac{b + \int_{-\infty}^{\rho-a} f(\breve{a}) d\breve{a}[a]}{\int_{\rho-a}^{\infty} f(\breve{a}) d\breve{a}} = 52.5 + \frac{20 + 0.225 \cdot 30}{0.775} \approx 87.0$. The contractor's IR constraint becomes

$$
\begin{aligned}
& \int_{\breve{b}^*(\rho,a,b)}^{\infty} g(\breve{b}) \int_{-\infty}^{\infty} f(\breve{a})[-\breve{a} + b] d\breve{a} d\breve{b} \\
& + \int_{-\infty}^{\breve{b}^*(\rho,a,b)} g(\breve{b}) [\int_{-\infty}^{\rho-a} f(\breve{a})[-\breve{a} - a] d\breve{a} + \int_{\rho-a}^{\infty} f(\breve{a})[-\rho] d\breve{a}] d\breve{b} \\
\ge \; & \int_{-\infty}^{\infty} f(\breve{a})[-\breve{a}] \\
\Leftrightarrow \; & \int_{\breve{b}^*(\rho,a,b)}^{110} g(\breve{b}) \int_{0}^{100} f(\breve{a})[-\breve{a} + 20] d\breve{a} d\breve{b} \\
& + \int_{0}^{\breve{b}^*(\rho,a,b)} g(\breve{b}) [\int_{0}^{52.5-30} f(\breve{a})[-\breve{a} - 30] d\breve{a} + \int_{52.5-30}^{100} f(\breve{a})[-52.5] d\breve{a}] d\breve{b} \\
\ge \; & \int_{0}^{100} f(\breve{a})[-\breve{a}] \\
\Leftrightarrow \; & \int_{\breve{b}^*(\rho,a,b)}^{110} \frac{1}{110} \frac{1}{100} [\frac{-(100)^2}{2} + 100 \cdot 20] d\breve{b} + \\
& \int_{0}^{\breve{b}^*(\rho,a,b)} \frac{1}{110} [\frac{1}{100} [\frac{-(22.5)^2}{2} - 22.5 \cdot 30] + \frac{1}{100} [-52.5 \cdot (100 - 22.5)]] d\breve{b} \\
\ge \; & -50 \\
\Leftrightarrow \; & \int_{\breve{b}^*(\rho,a,b)}^{110} \frac{15.5}{110} d\breve{b} + \int_{0}^{\breve{b}^*(\rho,a,b)} - \frac{49.96875}{110} d\breve{b} \ge -50
\end{aligned}
$$

Substituting $\breve{b}^*(\rho, a, b) = 87.0$ gives approximately $-6.3 - 39.5 \ge -50$ for the above inequality. Thus the contractor's IR constraint is satisfied.

The contractee's IR constraint becomes

$$\begin{aligned}
&\int_{\breve{b}^*(\rho,a,b)}^{\infty} g(\breve{b})[\breve{b}-b]d\breve{b} + \int_{-\infty}^{\breve{b}^*(\rho,a,b)} g(\breve{b})[\int_{-\infty}^{\rho-a} f(\breve{a})[\breve{b}+a]d\breve{a} + \int_{\rho-a}^{\infty} f(\breve{a})\rho d\breve{a}]d\breve{b} \\
&\geq \int_{-\infty}^{\infty} g(\breve{b})\breve{b}d\breve{b} \\
\Leftrightarrow \quad & \int_{\breve{b}^*(\rho,a,b)}^{110} \frac{1}{110}[\breve{b}-20]d\breve{b} \\
& + \int_{0}^{\breve{b}^*(\rho,a,b)} \frac{1}{110}[\int_{0}^{52.5-30} \frac{1}{100}[\breve{b}+30]d\breve{a} + \int_{52.5-30}^{100} \frac{1}{100} 52.5 d\breve{a}]d\breve{b} \geq 55 \\
\Leftrightarrow \quad & \int_{\breve{b}^*(\rho,a,b)}^{110} \frac{1}{110}[\breve{b}-20]d\breve{b} + \int_{0}^{\breve{b}^*(\rho,a,b)} \frac{1}{110}[\frac{22.5}{100}[\breve{b}+30] + \frac{77.5}{100} 52.5]d\breve{b} \geq 55 \\
\Leftrightarrow \quad & \int_{\breve{b}^*(\rho,a,b)}^{110} \frac{1}{110}[\breve{b}-20]d\breve{b} + \int_{0}^{\breve{b}^*(\rho,a,b)} 0.43125 + \frac{0.225}{110}\breve{b}d\breve{b} \geq 55
\end{aligned}$$

Substituting $\breve{b}^*(\rho,a,b) = 87.0$ gives approximately $16.4 + 45.3 \geq 55$ for the above inequality. Thus the contractee's IR constraint is satisfied. □

Proof. (Theorem 3.9). We prove this under condition 1. The proof under 2 is analogous. With F, the contractor's payoff is $-\rho_F$, and the contractee's ρ_F. We construct a leveled commitment contract where the *contractee will surely not decommit* because its penalty is chosen high and $\breve{b}$ is bounded from above. Choose $\rho = \rho_F$, and $a = \rho_F - E[\breve{b}] + \epsilon$. The contractor decommits if $-\breve{a} - a > -\rho \Leftrightarrow \breve{a} < \rho - a = E[\breve{b}] - \epsilon$. This has nonzero probability because bounded f and $\int_{-\infty}^{E[\breve{b}]} f(\breve{a})d\breve{a} > 0$ imply $\exists \epsilon > 0$ s.t. $\int_{-\infty}^{E[\breve{b}]-\epsilon} f(\breve{a})d\breve{a} > 0$. The contractee's expected payoff increased: it is ρ_F if the contractor does not decommit, and $E[\breve{b}] + a = \rho_F + \epsilon > \rho_F$ if the contractor does. The contractor's expected payoff also increased:

$$\begin{aligned}
-\rho_F < \int_{-\infty}^{\rho-a} f(\breve{a})[-\breve{a}-a]d\breve{a} + \int_{\rho-a}^{\infty} f(\breve{a})[-\rho]d\breve{a} \\
\Leftrightarrow 0 < \int_{-\infty}^{\rho-a} f(\breve{a})[-\breve{a}-a+\rho_F]d\breve{a}
\end{aligned}$$

which is implied by $\int_{-\infty}^{E[\breve{b}]-\epsilon} f(\breve{a})d\breve{a} > 0$ (Because f is bounded, all of this probability mass cannot be on a single point $\breve{a} = \rho - a$ $(= E[\breve{b}] - \epsilon)$). □

Proof. (Theorem 3.11). In the proof of Theorem 3.9, a leveled commitment contract was constructed where one agent was sure not to decommit. When one agent is known not to decommit, SIMUDBP games are equivalent to SEQD games. Therefore, the proof of Theorem 3.9 applies. □

Proof. (Theorem 3.13). In the proof of Theorem 3.9, a leveled commitment contract was constructed where one agent was sure not to decommit. When one agent is known not to decommit, SIMUDNP games are equivalent to SEQD games. Therefore, the proof of Theorem 3.9 applies. □

Proof. (Theorem 3.14). A CSM-contract can move the iterative refinement search focus from any task allocation to any other in a single step (i.e. in a single contract). Thus the optimal task allocation can be reached from any task allocation: there are no local optima. Therefore the hill-climbing algorithm does not need to backtrack in order to reach the global optimum.

Because agents only make individually rational contracts, every contract is an improvement of social welfare. Therefore, no task allocation is visited more than once. Because there are a finite number of tasks and agents, there is only a finite number of task allocations. It follows that the global optimum is reached in a finite number of steps. □

Proof. (Theorem 4.1). Let us analyze two arbitrary potential coalitions S and T, where $S, T \subseteq A$ and $S \cap T = \emptyset$. The conditions in the theorem state

$$\forall r_S \geq 0, \forall r_T \geq 0, c_{S \cup T}(r_S + r_T) \leq c_S(r_S) + c_T(r_T)$$

and obviously

$$\exists r'_S, r'_T \geq 0 \text{ s.t. } c_S(r'_S) + c_{comp} \cdot r'_S + c_T(r'_T) + c_{comp} \cdot r'_T$$
$$= \min_r [c_S(r) + c_{comp} \cdot r] + \min_r [c_T(r) + c_{comp} \cdot r]$$

It follows that

$$\exists r'_S, r'_T \geq 0 \text{ s.t. } c_{S \cup T}(r'_S + r'_T) + c_{comp} \cdot (r'_S + r'_T)$$

$$\le \min_r[c_S(r) + c_{comp} \cdot r] + \min_r[c_T(r) + c_{comp} \cdot r]$$

$$\Leftrightarrow \quad \exists r' \ge 0 \text{ s.t. } c_{S\cup T}(r') + c_{comp} \cdot r' \le \min_r[c_S(r) + c_{comp} \cdot r] + \min_r[c_T(r) + c_{comp} \cdot r]$$

$$\Leftrightarrow \quad \min_r[c_{S\cup T}(r) + c_{comp} \cdot r] \le \min_r[c_S(r) + c_{comp} \cdot r] + \min_r[c_T(r) + c_{comp} \cdot r]$$

$$\Leftrightarrow \quad v_{S\cup T}(c_{comp}) \ge v_S(c_{comp}) + v_T(c_{comp})$$

which completes the proof. □

Proof. (Theorem 4.2). **Counterexample.** Let us analyze a 2-agent game where $A = \{1, 2\}$. Let the performance profiles of the algorithms be

$$c_{\{1\}}(r) = c_{\{2\}}(r) = \begin{cases} \frac{1}{2} - \frac{1}{2}r & \text{if } 0 \le r \le 1 \\ 0 & \text{if } r > 1 \end{cases} \quad \text{and } c_{\{1,2\}}(r) = \begin{cases} 1 & \text{if } 0 \le r \le 1 \\ 2 - r & \text{if } 1 < r \le 2 \\ 0 & \text{if } r > 2 \end{cases}$$

Thus (see also Figure A.1),

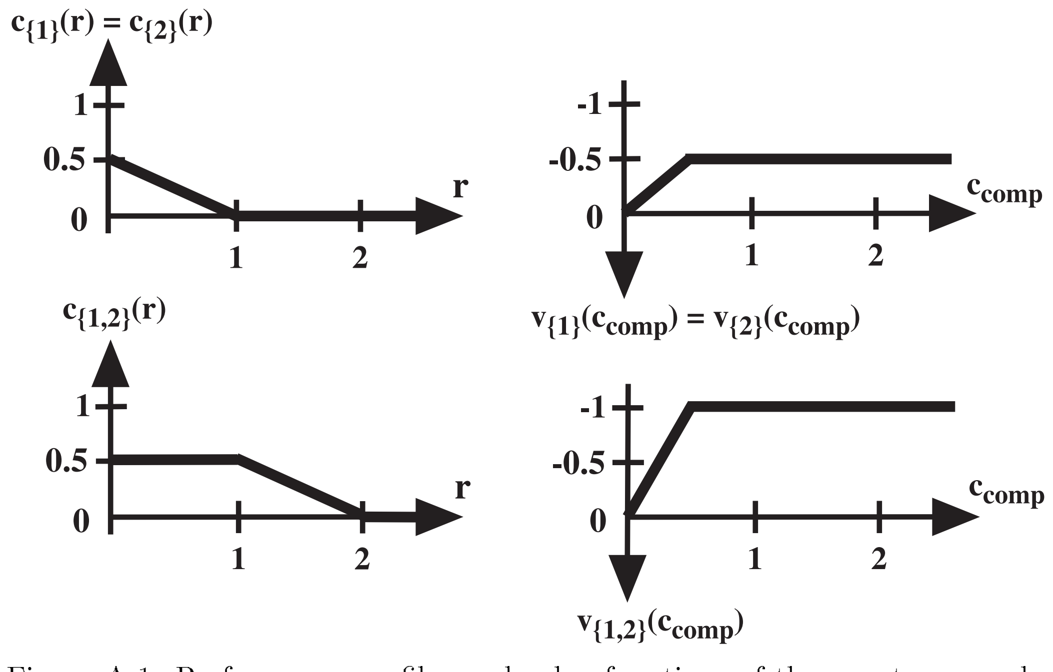

Figure A.1. Performance profiles and value functions of the counterexample.

$$v_{\{1\}}(c_{comp}) = v_{\{2\}}(c_{comp}) = -\min_r[c_{\{2\}}(r) + c_{comp} \cdot r] = \begin{cases} -c_{comp} & \text{if } c_{comp} \le \frac{1}{2} \\ -\frac{1}{2} & \text{if } c_{comp} > \frac{1}{2} \end{cases}$$

$$\text{and } v_{\{1,2\}}(c_{comp}) = -\min_r[c_{\{1,2\}}(r) + c_{comp}\cdot r] = \begin{cases} -2c_{comp} & \text{if } c_{comp} \le \frac{1}{2} \\ -1 & \text{if } c_{comp} > \frac{1}{2} \end{cases}$$

So when $c_{comp} \le \frac{1}{2}$,

$$v_{\{1,2\}}(c_{comp}) = -2c_{comp} = -c_{comp} + -c_{comp} = v_{\{1\}}(c_{comp}) + v_{\{2\}}(c_{comp})$$

and when $c_{comp} > \frac{1}{2}$,

$$v_{\{1,2\}}(c_{comp}) = -1 = -\frac{1}{2} + -\frac{1}{2} = v_{\{1\}}(c_{comp}) + v_{\{2\}}(c_{comp})$$

Thus, $(\forall c_{comp}, \forall S, T \subseteq A, S \cap T = \emptyset), v_{S\cup T}(c_{comp}) \ge v_S(c_{comp}) + v_T(c_{comp})$, i.e. the game is BRSUP for all c_{comp}. But $c_{\{1,2\}}(\frac{1}{2}+\frac{1}{2}) = 1 > \frac{1}{4}+\frac{1}{4} = c_{\{1\}}(\frac{1}{2}) + c_{\{2\}}(\frac{1}{2})$. □

The proof of Theorem 4.3 relies on the following Lemma:

Lemma A.1 *Let $f(x)$ be a decreasing, convex function. For any given x^*, $\exists c \ge 0$ s.t.*

$$\min_x[f(x) + cx] = f(x^*) + cx^*$$

Proof. (Lemma A.1). Let us define $x' = \text{argmin}_x[f(x) + cx]$. Assume—for contradiction—that $\exists x^*$ s.t. $\forall c \ge 0$,

$$\begin{aligned} \min_x[f(x) + cx] &\neq f(x^*) + cx^* \\ \Leftrightarrow f(x') + cx' &\neq f(x^*) + cx^* \end{aligned}$$

Because $f(x)$ is convex,

$$\begin{aligned} f(x^*) &\le \frac{f(x^*-\delta) + f(x^*+\delta)}{2} \\ \Rightarrow \lim_{\delta\to 0}\frac{f(x^*) - f(x^*-\delta)}{\delta} &\le \lim_{\delta\to 0}\frac{f(x^*+\delta) - f(x^*)}{\delta} \end{aligned}$$

Thus $c \ge 0$ is well-defined when chosen as follows:

$$\lim_{\delta\to 0}\frac{f(x^*) - f(x^*-\delta)}{\delta} \le -c \le \lim_{\delta\to 0}\frac{f(x^*+\delta) - f(x^*)}{\delta}$$

Now there are two cases:

Case 1: $x' < x^*$**:**

$$\begin{aligned}
x' &< x^* \\
\Leftrightarrow \operatorname*{argmin}_x[f(x)+cx] &< x^* \\
\Leftrightarrow f(\operatorname*{argmin}_x[f(x)+cx]) + c\cdot \operatorname*{argmin}_x[f(x)+cx] &< f(x^*)+cx^* \\
\Leftrightarrow f(x')+cx' &< f(x^*)+cx^* \\
\Leftrightarrow f(x^*-\epsilon)+c\cdot(x^*-\epsilon) &< f(x^*)+cx^* \\
\Leftrightarrow f(x^*)-f(x^*-\epsilon) &> -c\epsilon \\
\Leftrightarrow \frac{f(x^*)-f(x^*-\epsilon)}{\epsilon} &> -c \\
\Rightarrow \frac{f(x^*)-f(x^*-\epsilon)}{\epsilon} &> \lim_{\delta\to 0}\frac{f(x^*)-f(x^*-\delta)}{\delta}
\end{aligned}$$

This violates convexity. Contradiction.

Case 2: $x' > x^*$**:**

$$\begin{aligned}
x' &> x^* \\
\Leftrightarrow \operatorname*{argmin}_x[f(x)+cx] &> x^* \\
\Leftrightarrow f(\operatorname*{argmin}_x[f(x)+cx]) + c\cdot \operatorname*{argmin}_x[f(x)+cx] &< f(x^*)+cx^* \\
\Leftrightarrow f(x')+cx' &< f(x^*)+cx^* \\
\Leftrightarrow f(x^*+\epsilon)+c\cdot(x^*+\epsilon) &< f(x^*)+cx^* \\
\Leftrightarrow \frac{f(x^*+\epsilon)-f(x^*)}{\epsilon} &< -c \\
\Rightarrow \frac{f(x^*+\epsilon)-f(x^*)}{\epsilon} &< \lim_{\delta\to 0}\frac{f(x^*+\delta)-f(x^*)}{\delta}
\end{aligned}$$

This also violates convexity. Contradiction. Because both cases lead to a contradiction, the original assumption is false. □

Proof. (Theorem 4.3). The *if*-part was proven in Theorem 4.1. Now the *only if*-part is proven.

Game is BRSUP $\forall c_{comp}$

$$\Leftrightarrow \quad (\forall c_{comp}, \forall S, T \subseteq A, S \cap T = \emptyset), v_{S\cup T}(c_{comp}) \geq v_S(c_{comp}) + v_T(c_{comp})$$

$$\Leftrightarrow \quad (\forall c_{comp}, \forall S, T \subseteq A, S \cap T = \emptyset),$$

$$\min_r[c_{S\cup T}(r) + c_{comp} \cdot r] \leq \min_r[c_S(r) + c_{comp} \cdot r] + \min_r[c_T(r) + c_{comp} \cdot r]$$

$$\Leftrightarrow \quad (\forall c_{comp}, \forall S, T \subseteq A, S \cap T = \emptyset, \forall r_S, r_T \geq 0),$$

$$\min_r[c_{S\cup T}(r) + c_{comp} \cdot r] \leq c_S(r_S) + c_{comp} \cdot r_S + c_T(r_T) + c_{comp} \cdot r_T$$

Now, by Lemma A.1, for any $r_S + r_T \geq 0$, $\exists c_{comp} \geq 0$ s.t. $\min_r[c_{S\cup T}(r) + c_{comp} \cdot r] = c_{S\cup T}(r_S + r_T) + c_{comp} \cdot (r_S + r_T)$. Thus,

$$(\forall S, T \subseteq A, S \cap T = \emptyset, \forall r_S, r_T \geq 0, \exists c_{comp} \geq 0),$$

$$c_{S\cup T}(r_S + r_T) + c_{comp} \cdot (r_S + r_T) \leq c_S(r_S) + c_{comp} \cdot r_S + c_T(r_T) + c_{comp} \cdot r_T$$

$$\Leftrightarrow \quad (\forall S, T \subseteq A, S \cap T = \emptyset, \forall r_S, r_T \geq 0, \exists c_{comp} \geq 0), c_{S\cup T}(r_S + r_T) \leq c_S(r_S) + c_T(r_T)$$

$$\Leftrightarrow \quad (\forall S, T \subseteq A, S \cap T = \emptyset, \forall r_S, r_T \geq 0), c_{S\cup T}(r_S + r_T) \leq c_S(r_S) + c_T(r_T)$$

This completes the proof. □

Proof. (Theorem 4.4).

$$(\forall S, T \subseteq A, S \cap T = \emptyset, \forall r_S, r_T \geq 0), c_{S\cup T}(r_S + r_T) > c_S(r_S) + c_T(r_T)$$

$$\Leftrightarrow \quad (\forall S, T \subseteq A, S \cap T = \emptyset, \forall r_S, r_{S\cup T} \geq 0), c_{S\cup T}(r_{S\cup T}) > c_S(r_S) + c_T(r_{S\cup T} - r_S)$$

$$\Leftrightarrow \quad (\forall c_{comp}, \forall S, T \subseteq A, S \cap T = \emptyset, \forall r_S, r_{S\cup T} \geq 0), c_{S\cup T}(r_{S\cup T}) + c_{comp} \cdot r_{S\cup T}$$

$$> c_S(r_S) + c_{comp} \cdot r_S + c_T(r_{S\cup T} - r_S) + c_{comp} \cdot (r_{S\cup T} - r_S)$$

$$\Rightarrow \quad (\forall c_{comp}, \forall S, T \subseteq A, S \cap T = \emptyset, \forall r_S, r_{S\cup T} \geq 0),$$

$$\min_r[c_{S\cup T}(r) + c_{comp} \cdot r]$$

$$> c_S(r_S) + c_{comp} \cdot r_S + c_T(r_{S\cup T} - r_S) + c_{comp} \cdot (r_{S\cup T} - r_S)$$

$$\geq \min_r[c_S(r) + c_{comp} \cdot r] + \min_r[c_T(r) + c_{comp} \cdot (r)]$$

$$\Leftrightarrow \quad (\forall c_{comp}, \forall S, T \subseteq A, S \cap T = \emptyset), v_{S\cup T}(c_{comp}) < v_S(c_{comp}) + v_T(c_{comp})$$

$\Leftrightarrow$ Game is bounded rational subadditive $\forall c_{comp}$

This completes the proof. □

Proof. (Theorem 4.5). Let us analyze a game that is bounded rational subadditive for some c_{comp}, i.e. $(\forall S, T \subseteq A, S \cap T = \emptyset), v_{S\cup T}(c_{comp}) < v_S(c_{comp}) + v_T(c_{comp})$. Let us study a coalition structure $CS^* = \{\{1\}, \{2\}, ..., \{|A|\}\}$. Let us choose $\vec{x}$ s.t. $\forall i \in A$, $x_i = v_{\{i\}}(c_{comp})$. Now,

$$\sum_{i\in A} x_i = \sum_{i\in A} v_{\{i\}}(c_{comp}) = \sum_{j\in CS^*} v_{S_j}(c_{comp})$$

and

$$\forall S \subset A, \sum_{i\in S} x_i = \sum_{i\in S} v_{\{i\}}(c_{comp}) \geq v_S(c_{comp})$$

Thus $\vec{x} \in BRC(c_{comp})$ which implies $BRC(c_{comp}) \neq \emptyset$. □

Proof. (Theorem 4.6). Shapley [Shapley, 1967] proved the following fact (his Theorem 2) for rational agents. In games where $CS^{R*} = \{A\}$, $C \neq \emptyset$ iff for every minimal balanced set $\mathcal{B} = \{B_1, ..., B_p\}, \sum_{j=1}^{p} \lambda_j v^R_{B_j} \leq v^R_A$. Theorem 4.6 follows by analogy. □

Proof. (Theorem 4.7). Shapley [Shapley, 1967] proved the following fact (his Theorem 3) for rational agents. In a superadditive game, $C \neq \emptyset$ iff for every proper minimal balanced set $\mathcal{B} = \{B_1, ..., B_p\}, \sum_{j=1}^{p} \lambda_j v^R_{B_j} \leq v^R_A$. Charnes and Kortanek [Charnes and Kortanek, 1966] proved that this set of inequalities is minimal. Theorem 4.7 follows by analogy. □

Proof. (Theorem 4.8). Let us analyze an arbitrary minimal balanced set $\mathcal{B} = \{B_1, ..., B_p\}$.

$$\begin{aligned}
& (\forall B \in \mathcal{B}, \forall r_B \geq 0), \sum_{j=1}^{p} \lambda_j c_{B_j}(r_{B_j}) \geq c_A(\sum_{j=1}^{p} \lambda_j r_{B_j}) \\
\Rightarrow \quad & (\forall c_{comp}, \forall B \in \mathcal{B}, \forall r_B \geq 0, \exists r_A \geq 0), \\
& \sum_{j=1}^{p} \lambda_j c_{B_j}(r_{B_j}) + c_{comp} \cdot (-r_A + \sum_{j=1}^{p} \lambda_j r_{B_j}) \geq c_A(r_A) \\
\Leftrightarrow \quad & (\forall c_{comp}, \forall B \in \mathcal{B}, \forall r_B \geq 0, \exists r_A \geq 0), \\
& \sum_{j=1}^{p} \lambda_j c_{B_j}(r_{B_j}) + c_{comp} \cdot \sum_{j=1}^{p} \lambda_j r_{B_j} \geq c_A(r_A) + c_{comp} \cdot r_A
\end{aligned}$$

$$
\begin{aligned}
\Leftrightarrow \quad & (\forall c_{comp}, \forall B \in \mathcal{B}, \forall r_B \geq 0), \\
& \sum_{j=1}^{p} \lambda_j c_{B_j}(r_{B_j}) + c_{comp} \cdot \sum_{j=1}^{p} \lambda_j r_{B_j} \geq \min_r [c_A(r) + c_{comp} \cdot r] \\
\Leftrightarrow \quad & (\forall c_{comp}, \forall B \in \mathcal{B}, \forall r_B \geq 0), \\
& \sum_{j=1}^{p} \lambda_j [c_{B_j}(r_{B_j}) + c_{comp} \cdot r_{B_j}] \geq \min_r [c_A(r) + c_{comp} \cdot r] \\
\Leftrightarrow \quad & (\forall c_{comp}), \sum_{j=1}^{p} \lambda_j \min_{r'_{B_j}} [c_{B_j}(r'_{B_j}) + c_{comp} \cdot r'_{B_j}] \geq \min_r [c_A(r) + c_{comp} \cdot r] \\
\Leftrightarrow \quad & (\forall c_{comp}), \sum_{j=1}^{p} \lambda_j v_{B_j}(c_{comp}) \leq v_A(c_{comp})
\end{aligned}
$$

Since this holds for an arbitrary minimal balanced set, it has to hold for every minimal balanced set. Thus, by Theorem 4.6, $BRC(c_{comp}) \neq \emptyset$. □

Proof. (Theorem 4.9). Analogous to the proof of Theorem 4.8, except that now an arbitrary *proper* minimal balanced set is considered. Furthermore, the reference to Theorem 4.6 should be changed to a reference to Theorem 4.7. □

Proof. (Theorem 5.1). First it is proven that the exchange is completed starting from any subgame (x_n, p_n) where $x_n \in [0, 1)$ and $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$. The two inequalities of p_n are preserved by S_s and S_d for all remaining n. An example path starting from $(x_n, p_n) = (0, 0)$ is presented in Fig. 5.2 middle. At a point (x_n, p_n), the agents' strategies prescribe that $(x_{n+1}, p_{n+1}) = (max\{x \in X | p^{min}(x) \leq p_n\}, p^{max}(x_n))$ is reached. Let us denote the size of the smallest possible (due to a finite number of countable goods) delivery from x_n by Δx. Now there are two cases. In the first case (demander "behind"), $p_n < p^{min}(x_n + \Delta x)$, and $(x_{n+1}, p_{n+1}) = (x_n, p^{max}(x_n))$. The result is that now $p_{n+1} = p^{max}(x_n) \geq p^{min}(x_n + \Delta x)$ and the conditions for the second case are fulfilled except that the indexes are incremented by one. In the second case (supplier "behind"), $p_n \geq p^{min}(x_n + \Delta x)$, and $(x_{n+1}, p_{n+1}) = (max\{x \in X | p^{min}(x) \leq p_n\}, p^{max}(x_n))$. In this case $x_{n+1} \geq x_n + \Delta x$ (for $x_n < 1$) because $p_n \geq p^{min}(x_n + \Delta x)$. So from any point $(x_n, p_n), p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$ the exchange can proceed along the x-axis at least by Δx (using one or two steps depending whether the first case from above is reached or not). With a finite number of countable goods, there are only a finite number of Δx's (of potentially different sizes), and thus using a finite number of steps $x = 1$ is reached. But the strategies preserved $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$. Thus it follows from the fact $p^{min}(1) = p^{max}(1) = p^{contr}$ that the point $(1, p^{contr})$ has been reached (in a finite number of steps).

Next we show that S_s is a best response to S_d. In any subgame where $p^{min}(x_n) > p_n$ or $p_n > p^{max}(x_n)$, S_d specifies that the demander is going to pay no further. The supplier's best response to this is obviously to not deliver anything further, i.e. to exit. Next we analyze subgames where $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$. Defecting gives the supplier a remaining benefit of $p^{max}(x_n) - p_n$, because the supplier does not deliver anything further and the demander still pays during one step, increasing total payment to $p^{max}(x_n)$ according to S_d. Cooperating throughout the rest of the exchange according to S_s gives the supplier a remaining benefit of $p^{contr} - p_n - v_s(1) + v_s(x_n)$ because the exchange will be completed as shown earlier. By definition, $p^{max}(x_n) = p^{contr} - v_s(1) + v_s(x_n)$, and thus the benefit from exiting $(p^{max}(x_n) - p_n)$ equals the benefit from cooperating throughout the rest of the exchange $(p^{contr} - p_n - v_s(1) + v_s(x_n))$. Starting from a point where $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$, and following strategies S_s and S_d, the agents preserve the property $p^{min}(x_k) \leq p_k \leq p^{max}(x_k)$ for every step k. Because—as shown above—in no such subgame is the supplier motivated to exit, cooperating throughout the rest of the exchange is the supplier's best response to S_d starting from any subgame where $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$. Put together, S_s is the supplier's best response to demander's strategy S_d in every subgame.

Now we show that S_d is a best response to S_s. In any subgame where $p^{min}(x_n) > p_n$ or $p_n > p^{max}(x_n)$, S_s specifies that the supplier is going to deliver no further. The demander's best response to this is obviously to not pay anything further, i.e. to exit. Next we analyze subgames where $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$. Defecting gives the demander a remaining benefit of $v_d(max\{x \in X|p^{min}(x) \leq p_n\}) - v_d(x_n)$ because the demander does not pay anything and the supplier still delivers (according to S_s) during one step. From the definition of $p^{min}(x)$, this benefit can be rewritten as $v_d(max\{x \in X|p^{contr} - v_d(1) + v_d(x) \leq p_n\}) - v_d(x_n) \leq p_n - p^{contr} + v_d(1) - v_d(x_n)$. Cooperating throughout the rest of the exchange according to S_d gives the demander a remaining benefit of $v_d(1) - v_d(x_n) - p^{contr} + p_n$ because the exchange will be completed as shown earlier. Therefore, the demander's benefit from exiting is no greater than its benefit from cooperating throughout the rest of the exchange. Starting from a point where $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$, and following strategies S_s and S_d, the agents preserve the property $p^{min}(x_k) \leq p_k \leq p^{max}(x_k)$ for every step k. Because—as shown above—in no such subgame is the demander motivated to exit, cooperating throughout the rest of the exchange is the demander's best response to S_s starting from any subgame where $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$. Put together, S_d is the demander's best response to supplier's strategy S_s in every subgame.

Now we show that there is no subgame perfect Nash equilibrium leading to completion in fewer steps. For any supplier's strategy that specifies $x_{n+1} > max\{x \in X|p^{min}(x) \leq p_n\} = max\{x \in X|p^{contr} - v_d(1) + v_d(x) \leq p_n\}$, the demander's payoff from defecting and keeping total payment at p_n is $v_d(x_{n+1}) - v_d(x_n) > p_n - p^{contr} + v_d(1) - v_d(x_n)$. Because the demander's payoff from cooperating throughout the exchange is only $v_d(1) - v_d(x_n) - p^{contr} + p_n$, the demander will defect. Thus, in subgame perfect Nash equilibrium, at no point can a supplier deliver more than was specified by S_s. Likewise, for any demander's strategy that specifies $p_{n+1} > p^{max}(x_n)$, the supplier's payoff from defecting and keeping total delivery at x_n is $p_{n+1} - p_n > p^{max}(x_n) - p_n = p^{contr} - v_s(1) + v_s(x_n) - p_n$. Because the supplier's payoff from cooperating throughout the exchange is only $p^{contr} - v_s(1) + v_s(x_n) - p_n$, the supplier will defect. Thus, in subgame perfect Nash equilibrium, at no point can a demander pay more than was prescribed by S_d. When the exchange begins, $(x_0, p_0) = (0, 0)$. For the induction step, assume that a point (x_q, p_q) has been reached, where $x_q \leq x_q^{(S_s,S_d)}$ and $p_q \leq p_q^{(S_s,S_d)}$. Now, $x_{q+1} \leq max\{x \in X | p^{min}(x) \leq p_q\} \leq max\{x \in X | p^{min}(x) \leq p_q^{(S_s,S_d)}\} = x_{q+1}^{(S_s,S_d)}$. Similarly $p_{q+1} \leq p^{max}(x_q) \leq p^{max}(x_q^{S_s}) = p_{q+1}^{S_s}$. This proves that no subgame perfect Nash equilibrium path leads to completion in fewer steps than the one induced by S_s and S_d. □

Proof. (Theorem 5.2). Let $x_1, x_2 \in X$ be the first consecutive amounts of cumulative delivery for which $p^{max}(x_1) < p^{min}(x_2)$. Now from any point (x, p) where $p > p^{max}(x_1)$ and $x \leq x_1$, the exchange cannot proceed to completion $(1, p^{contr})$ because the supplier's remaining payoff from cooperating up to completion is only $p^{contr} - p - v_s(1) + v_s(x) < p^{contr} - p^{max}(x_1) - v_s(1) + v_s(x_1) = 0$ and its remaining payoff from defecting is at least 0.

From any point (x, p) where $p < p^{min}(x_2)$ and $x \geq x_2$, the exchange cannot proceed to completion $(1, p^{contr})$ because the demander's remaining payoff from cooperating up to completion is only $v_d(1) - v_d(x) - p^{contr} + p < v_d(1) - v_d(x_2) - p^{contr} + p^{min}(x_2) = 0$, and its remaining payoff from defecting is at least 0.

Now let us analyze an arbitrary point (x, p), where $p \leq p^{max}(x_1)$ and $x \leq x_1$. From this point, any move by the supplier that increases cumulative delivery to or beyond x_2 leads to a point (x^{sup}, p^{sup}), where $x^{sup} \geq x_2$, $p^{sup} \leq p^{max}(x_1) < p^{min}(x_2)$, i.e. to the region where the demander is motivated to not complete the exchange. The demander knows this, so his best response strategy must prescribe his simultaneous move to be to pay nothing so that this region is actually reached. Similarly, from (x, p), any move by the demander that increases cumulative payment beyond $p^{max}(x_1)$ leads to a point (x^{dem}, p^{dem}), where $x^{dem} \leq x_1$, $p^{dem} > p^{max}(x_1)$, i.e. to the region where the supplier is motivated to not complete the exchange. The supplier knows this, so his best response strategy must prescribe his simultaneous move to be to deliver nothing so that this region is actually reached. Therefore neither agent can make a move beyond $(x_1, p^{max}(x_1))$ without causing the other agent to defect. Thus the exchange will not be completed. □

Proof. (Theorem 5.3). It is first proven that the exchange is completed in a finite number of steps—given that it starts from a subgame (x_n, p_n) where $0 \leq x_n \leq 1$ and $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$. Obviously, these two inequalities are preserved for all remaining n by strategies $S_{s_{un}}$ and S_d throughout the exchange. At any such point (x_n, p_n), the agents' strategies prescribe the next point (x_{n+1}, p_{n+1}). There are two cases depending whether $max\{x \in X | p^{min}(x) \leq p_n\}$ is well-defined, and several subcases:

Max defined, Case 1: (neither passes on current move) In this case, $x_n < max\{x \in X | p^{min}(x) \leq p_n\}$, and $p_n < p^{max}(x_n)$. Thus, $(x_{n+1}, p_{n+1}) = (x_n + \Delta x, p^{max}(x_n))$ for some $\Delta x > 0$. From there the point (x_{n+2}, p_{n+2}) is reached where $x_{n+2} \geq sup\{x \in X | p^{min}(x) \leq p_{n+1}\} - \underline{\epsilon} = sup\{x \in X | p^{min}(x) \leq p^{max}(x_n)\} - \underline{\epsilon} \geq min(x_n + \epsilon - \underline{\epsilon}, 1)$. Thus, an amount of at least $\epsilon - \underline{\epsilon}$ is delivered in two steps, or the exchange has reached $x = 1$.

Max defined, Case 2: (supplier passes on current move) In this case, $x_n = max\{x \in X | p^{min}(x) \leq p_n\}$, and $p_n < p^{max}(x_n)$. Thus, $(x_{n+1}, p_{n+1}) = (x_n, p^{max}(x_n))$. From there the point (x_{n+2}, p_{n+2}) is reached where $p_{n+2} = p_{n+1} = p^{max}(x_n)$, and $x_{n+2} \geq sup\{x \in X | p^{min}(x) \leq p^{max}(x_n)\} - \underline{\epsilon} \geq min(x_n + \epsilon - \underline{\epsilon}, 1)$. Thus, an amount of at least $\epsilon - \underline{\epsilon}$ is delivered in two steps, or the exchange has reached $x = 1$.

Max defined, Case 3: (demander passes on current move) In this case, $p_n = p^{max}(x_n)$, and $x_n < max\{x \in X | p^{min}(x) \leq p_n\}$. Thus the point $(x_{n+1}, p_{n+1}) = (x_{n+1}, p_n)$ is reached where $x_{n+1} \geq sup\{x \in X | p^{min}(x) \leq p^{max}(x_n)\} - \underline{\epsilon} \geq min(x_n + \epsilon - \underline{\epsilon}, 1)$. Thus, an amount of at least $\epsilon - \underline{\epsilon}$ is delivered in one step, or the exchange has reached $x = 1$. Note that at any point (x_n, p^{contr}) where $p^{max}(x_n) = p^{contr}$, $x_{n+1} = max\{x \in X | p^{min}(x) \leq p_n\} = 1$, and the exchange will be completed.

Max defined, Case 4: (completion) In this case, $p_n = p^{max}(x_n)$, and $x_n = max\{x \in X | p^{min}(x) \leq p_n\}$. Here, $x_n = 1$, and $p_n = p^{contr}$: the exchange has already been completed.

Max undefined, Case 1: (neither passes on current move) In this case, $x_n < sup\{x \in X | p^{min}(x) \leq p_n\} - \underline{\epsilon}$, and $p_n < p^{max}(x_n)$. Thus, $(x_{n+1}, p_{n+1}) = (x_n + \Delta x, p^{max}(x_n))$ for some $\Delta x > 0$. From there the point (x_{n+2}, p_{n+2}) is reached where $x_{n+2} \geq sup\{x \in X | p^{min}(x) \leq p_{n+1}\} - \underline{\epsilon} = sup\{x \in X | p^{min}(x) \leq p^{max}(x_n)\} - \underline{\epsilon} \geq min(x_n + \epsilon - \underline{\epsilon}, 1)$. Thus, an amount of at least $\epsilon - \underline{\epsilon}$ is delivered in two steps, or the exchange has reached $x = 1$.

Max undefined, Case 2: (supplier passes on current move) In this case, $x_n \geq sup\{x \in X | p^{min}(x) \leq p_n\} - \underline{\epsilon}$, and $p_n < p^{max}(x_n)$. Note that because $p^{min}(x_n) \leq p_n$, it holds that $x_n \leq sup\{x \in X | p^{min}(x) \leq p_n\}$. Now, $(x_{n+1}, p_{n+1}) = (x_n, p^{max}(x_n))$. From there the point (x_{n+2}, p_{n+2}) is reached where $p_{n+2} = p_{n+1} = p^{max}(x_n)$, and $x_{n+2} \geq sup\{x \in X | p^{min}(x) \leq p^{max}(x_n)\} - \underline{\epsilon} \geq min(x_n + \epsilon - \underline{\epsilon}, 1)$. Thus, an amount of at least $\epsilon - \underline{\epsilon}$ is delivered in two steps, or the exchange has reached $x = 1$.

Max undefined, Case 3: (demander passes on current move) In this case, $p_n = p^{max}(x_n)$, and $x_n < sup\{x \in X | p^{min}(x) \leq p_n\} - \underline{\epsilon}$. Thus the point $(x_{n+1}, p_{n+1}) = (x_{n+1}, p_n)$ is reached where $x_{n+1} = sup\{x \in X | p^{min}(x) \leq p^{max}(x_n)\} - \underline{\epsilon} \geq min(x_n + \epsilon - \underline{\epsilon}, 1)$. Thus, an amount of at least $\epsilon - \underline{\epsilon}$ is delivered in one step, or the exchange has reached $x = 1$.

Because each case extends cumulative delivery by at least a fixed amount $\epsilon - \underline{\epsilon}$ in no more than two steps, $x = 1$ is reached in a finite number of steps. But the strategies preserved $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$. Thus it follows from the fact $p^{min}(1) = p^{max}(1) = p^{contr}$ that the point $(1, p^{contr})$ has been reached, i.e. the exchange has been completed in a finite number of steps.

Now that we have proven that the exchange is completed in a finite number of steps, the arguments in the proof of Theorem 5.1 suffice to prove that $S_{s_{un}}$ is a best response to S_d in every subgame.

Now we show that S_d is a best response to $S_{s_{un}}$. In any subgame where $p^{min}(x_n) > p_n$ or $p_n > p^{max}(x_n)$, $S_{s_{un}}$ specifies that the supplier is going to deliver no further. The demander's best response to this is obviously to not pay anything further, i.e. to exit.

Next we analyze subgames where $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$. If the demander defects, it pays no further but the supplier, according to $S_{s_{un}}$, still delivers during one step. Thus the demander's remaining payoff is $v_d(max\{x \in X | p^{min}(x) \leq p_n\}) - v_d(x_n) = v_d(max\{x \in X | p^{contr} - v_d(1) + v_d(x) \leq p_n\}) - v_d(x_n) \leq p_n - p^{contr} + v_d(1) - v_d(x_n)$ if the result of the max-operator is defined. If it is not, the demander's payoff is $v_d(max(x_n, sup\{x \in X | p^{min}(x) \leq p_n\} - \underline{\epsilon})) - v_d(x_n) = v_d(max(x_n, sup\{x \in X | p^{contr} - v_d(1) + v_d(x) \leq p_n\} - \underline{\epsilon})) - v_d(x_n) < p_n - p^{contr} + v_d(1) - v_d(x_n)$. So, in either case the demander's payoff from defecting is no greater than $p_n - p^{contr} + v_d(1) - v_d(x_n)$. Cooperating throughout the rest of the exchange according to S_d gives the demander a remaining benefit of $v_d(1) - v_d(x_n) - p^{contr} + p_n$ because the exchange will be completed as shown earlier. Therefore, the demander's benefit from exiting is no greater than its benefit from cooperating throughout the rest of the exchange. Starting from a point where $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$, and following strategies $S_{s_{un}}$ and S_d, the agents preserve the property $p^{min}(x_k) \leq p_k \leq p^{max}(x_k)$ for every step k. Because—as shown above—in no such subgame is the demander motivated to exit, cooperating throughout the rest of the exchange is the demander's best response to $S_{s_{un}}$ starting from any subgame where $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$. Put together, S_d is the demander's best response to supplier's strategy $S_{s_{un}}$ in every subgame.

Now we show that $\exists \underline{\epsilon}^* > 0$ s.t. $\forall \underline{\epsilon}$ where $0 < \underline{\epsilon} \leq \underline{\epsilon}^*$, the exchange has a minimal number of steps out of all subgame perfect Nash equilibrium exchanges. Let S'_s and S'_d be strategies that lead to subgame perfect completion of the exchange in a minimal number of steps. A minimum exists because the number of steps is bounded below by zero. For any supplier's strategy that specifies $x_{n+1} > max\{x \in X | p^{min}(x) \leq p_n\} = max\{x \in X | p^{contr} - v_d(1) + v_d(x) \leq p_n\}$, the demander's payoff from defecting and keeping total payment at p_n is $v_d(x_{n+1}) - v_d(x_n) > p_n - p^{contr} + v_d(1) - v_d(x_n)$. For any supplier's strategy that specifies (when the above max is undefined) $x_{n+1} \geq sup\{x \in X | p^{min}(x) \leq p_n\} = sup\{x \in X | p^{contr} - v_d(1) + v_d(x) \leq p_n\}$, the demander's payoff from defecting and keeping total payment at p_n is $v_d(x_{n+1}) - v_d(x_n) > p_n - p^{contr} + v_d(1) - v_d(x_n)$ (the inequality is strict because the max was undefined). Because the demander's payoff from cooperating throughout the exchange is only $v_d(1) - v_d(x_n) - p^{contr} + p_n$, the demander will defect. Thus, in subgame perfect Nash equilibrium, at no point can a supplier deliver more than is specified by some $S_{s_{un}}$ (parameterized by an arbitrarily small $\underline{\epsilon} > 0$). Likewise, for any demander's strategy that specifies $p_{n+1} > p^{max}(x_n)$, the supplier's payoff from defecting and keeping total delivery at x_n is $p_{n+1} - p_n > p^{max}(x_n) - p_n = p^{contr} - v_s(1) + v_s(x_n) - p_n$. Because the supplier's payoff from cooperating throughout the exchange is only $p^{contr} - v_s(1) + v_s(x_n) - p_n$, the supplier will defect. Thus, in subgame perfect Nash equilibrium, at no point can a demander pay more than was prescribed by S_d.

When the exchange begins, $(x_0, p_0) = (0, 0)$. For the induction step, assume that a point $(x_q^{(S'_s,S'_d)}, p_q^{(S'_s,S'_d)})$ has been reached, where $x_q^{(S'_s,S'_d)} \leq x_q^{(S_{s_{un}},S_d)}$ and $p_q^{(S'_s,S'_d)} \leq p_q^{(S_{s_{un}},S_d)}$. Now,

$$p_{q+1}^{(S'_s,S'_d)} \leq p^{max}(x_q^{(S'_s,S'_d)}) \leq p^{max}(x_q^{(S_{s_{un}},S_d)}) = p_{q+1}^{(S_{s_{un}},S_d)}$$

Similarly,

$$\begin{aligned} x_{q+1}^{(S'_s,S'_d)} &\leq max\{x \in X | p^{min}(x) \leq p_q^{(S'_s,S'_d)}\} \\ &\leq max\{x \in X | p^{min}(x) \leq p_q^{(S_{s_{un}},S_d)}\} \\ &= x_{q+1}^{(S_{s_{un}},S_d)} \end{aligned}$$

if the max is defined. If the max is undefined,

$$\begin{aligned} x_{q+1}^{(S'_s,S'_d)} &< sup\{x \in X | p^{min}(x) \leq p_q^{(S'_s,S'_d)}\} \\ &= sup\{x \in X | p^{min}(x) \leq p_q^{(S'_s,S'_d)}\} - \delta \\ &\leq sup\{x \in X | p^{min}(x) \leq p_q^{(S_{s_{un}},S_d)}\} - \delta \end{aligned}$$

Now $\underline{\epsilon}$ can be chosen to equal δ, and it follows that $x_{q+1}^{(S'_s,S'_d)} \leq x_{q+1}^{(S_{s_{un}},S_d)}$. Because the path of play induced by $S_{s_{un}}$ and S_d has a finite number of steps, and the path of play induced by S_s' and S_d' is minimal, the latter also has just a finite number of steps. Thus, there is a finite number of δ's. Let us choose $\underline{\epsilon}^*$ to equal the smallest δ. Now, choosing $\underline{\epsilon} = \underline{\epsilon}^*$ at every step of the above induction, it follows that at every step of the exchange, the point $(x_q^{(S_s',S_d')}, p_q^{(S_s',S_d')})$ is no further in the exchange than $p_q^{(S_s',S_d')} \leq p_q^{(S_{sun},S_d)}$. Thus S_{sun} and S_d also induce a path that is minimal in steps.

What remains to be proven is that choosing $\underline{\epsilon} < \underline{\epsilon}^*$ does not increase the number of steps in the exchange. Let us denote S_{sun} by $S_{sun}^{=}$ when $\underline{\epsilon} = \underline{\epsilon}^*$ and by $S_{sun}^{<}$ when $\underline{\epsilon} < \underline{\epsilon}^*$. When the exchange begins, $(x_0, p_0) = (0, 0)$. For the induction step, assume that a point $(x_q^{(S_{sun}^{=},S_d')}, p_q^{(S_{sun}^{=},S_d')})$ has been reached, where $x_q^{(S_{sun}^{=},S_d')} \leq x_q^{(S_{sun}^{<},S_d)}$ and $p_q^{(S_{sun}^{=},S_d')} \leq p_q^{(S_{sun}^{<},S_d)}$. Now,

$$p_{q+1}^{(S_{sun}^{=},S_d')} = p^{max}(x_q^{(S_{sun}^{=},S_d')}) \leq p^{max}(x_q^{(S_{sun}^{<},S_d)}) = p_{q+1}^{(S_{sun}^{<},S_d)}$$

Similarly,

$$\begin{aligned} x_{q+1}^{(S_{sun}^{=},S_d')} &= max\{x \in X | p^{min}(x) \leq p_q^{(S_{sun}^{=},S_d')}\} \\ &\leq max\{x \in X | p^{min}(x) \leq p_q^{(S_{sun}^{<},S_d)}\} \\ &= x_{q+1}^{(S_{sun}^{<},S_d)} \end{aligned}$$

if the max is defined. If the max is undefined,

$$\begin{aligned} x_{q+1}^{(S_{sun}^{=},S_d')} &= sup\{x \in X | p^{min}(x) \leq p_q^{(S_{sun}^{=},S_d')}\} - \underline{\epsilon}^* \\ &\leq x_{q+1}^{(S_{sun}^{<},S_d')} \end{aligned}$$

This completes the proof. $\square$

Proof. (Theorem 5.4). For there to be a subgame perfect Nash equilibrium leading to completion of the exchange in a finite number of steps, there must be some $\epsilon > 0$ that is the supplier's smallest move on the equilibrium path. Now, if $p^{max}(x^*) < p^{min}(x^* + \epsilon)$, the proof of Theorem 5.2 can be used (by substituting $x_1 = x^*, x_2 = x^* + \epsilon$) to show that there is no subgame perfect Nash equilibrium leading to completion. But according to Theorem 5.4, such an x^* can be found for *any* $\epsilon > 0$. $\square$

Proof. (Theorem 5.5). The proof of Theorem 5.1 suffices to show that the exchange is completed in a minimal, finite number of steps using strategies S_s^{timed} and S_d^{timed} because these strategies are identical to S_s and S_d except that the former strategies explicitly specify *immediate* moves. Due to this prescription, the exchange is completed immediately, i.e. for all n, $t_n = 0$.

Now it is shown that S_s^{timed} is a best response to S_d^{timed}, and vice versa, in each subgame (x_n, p_n, t_n) where $x_n \in [0, 1]$, $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$, and $t_n = 0$. By following strategies S_s^{timed} and S_d^{timed}, only subgames of this type are reached. Thus the equilibrium will remain an equilibrium in each subgame that is reached.

First, it is proven that S_s^{timed} is the supplier's best response to S_d^{timed}. In a subgame where $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$ and $t_n = 0$, strategy S_s^{timed} has remaining payoff

$$
\begin{aligned}
&\pi_s^{(x_n, p_n, 0)}(S_s^{timed}, S_d^{timed}) \\
&= p^{contr} - p_n - v_s(1) + v_s(x_n) \\
&= p^{contr} - p^{max}(x_{n-1}) - v_s(1) + v_s(x_n) \\
&= p^{contr} - [p^{contr} - v_s(1) + v_s(x_{n-1})] - v_s(1) + v_s(x_n) \\
&= v_s(x_n) - v_s(x_{n-1})
\end{aligned}
$$

because the exchange is completed immediately. To avoid handling the beginning of the exchange $(x_0, p_0, 0) = (0, 0, 0)$ as a special case, we define $v_s(x_{-1})$ s.t. $p_0 = p^{max}(x_{-1}) = 0$. Thus, $v_s(x_{-1}) = v_s(1) - p^{contr}$.

Now, let us analyze the remaining payoff of an arbitrarily chosen strategy S_s' for the supplier. Let the path of play induced by S_s' against S_d^{timed} be defined by $x_{n+1}, x_{n+2}, ...$ and $t_{n+1}, t_{n+2}, ...$ where $x_0 = t_0 = 0$ and $\forall k \in \{n+1, n+2, ...\}, x_k \leq x_{k+1}, t_k \leq t_{k+1}$. This model incorporates the possibility that for some q, $x_q = x_{q+1} = ... = \lim_{M \to \infty} x_M$ or $t_q = t_{q+1} = ... = \lim_{M \to \infty} t_M$, i.e. the deliveries can stop at some finite step, or the exchange can stop at some (possibly finite) time. Note that the demander, according to S_d^{timed}, always makes its move $p_{k+1} = p^{max}(x_k)$ immediately after the supplier has made move k, i.e. at time t_k.

$$
\begin{aligned}
&\pi_s^{(x_n,p_n,0)}(S_s', S_d^{timed}) \\
&= \sum_{k=n}^{\infty} f_s^p(t_k)[p_{k+1} - p_k] - f_s^v(t_{k+1})[v_s(x_{k+1}) - v_s(x_k)] \\
&= \sum_{k=n}^{\infty} f_s^p(t_k)[v_s(x_k) - v_s(x_{k-1})] - f_s^v(t_{k+1})[v_s(x_{k+1}) - v_s(x_k)] \\
&\leq \sum_{k=n}^{\infty} f_s^p(t_k)[v_s(x_k) - v_s(x_{k-1})] - f_s^p(t_{k+1})[v_s(x_{k+1}) - v_s(x_k)] \\
&= \lim_{M\to\infty} \sum_{k=n}^{M} f_s^p(t_k)[v_s(x_k) - v_s(x_{k-1})] - f_s^p(t_{k+1})[v_s(x_{k+1}) - v_s(x_k)] \\
&= \lim_{M\to\infty} f_s^p(t_n)\cdot(-v_s(x_{n-1})) + f_s^p(t_M)v_s(x_M) \\
&\quad + f_s^p(t_{n+1})v_s(x_n) - f_s^p(t_{M+1})v_s(x_{M+1}) \\
&\quad + \sum_{k=n}^{M-1} [f_s^p(t_k) - f_s^p(t_{k+1})]v_s(x_k) + [f_s^p(t_{k+2}) - f_s^p(t_{k+1})]v_s(x_{k+1}) \\
&= \lim_{M\to\infty} f_s^p(t_n)\cdot(-v_s(x_{n-1})) + f_s^p(t_M)v_s(x_M) \\
&\quad + f_s^p(t_{n+1})v_s(x_n) - f_s^p(t_{M+1})v_s(x_{M+1}) \\
&\quad + [f_s^p(t_n) - f_s^p(t_{n+1})]v_s(x_n) + [f_s^p(t_{M+1}) - f_s^p(t_M)]v_s(x_M) \\
&= \lim_{M\to\infty} f_s^p(t_n)[v_s(x_n) - v_s(x_{n-1})] + f_s^p(t_{M+1})[\underbrace{v_s(x_M) - v_s(x_{M+1})}_{\leq 0}] \\
&\leq f_s^p(t_n)[v_s(x_n) - v_s(x_{n-1})] \\
&= v_s(x_n) - v_s(x_{n-1}) \\
&= \pi_s^{(x_n,p_n,0)}(S_s^{timed}, S_d^{timed})
\end{aligned}
$$

Because the remaining payoff to an arbitrary strategy S_s' is no better than the remaining payoff to S_s^{timed}, the latter is a best response to S_d^{timed}.

Next, it is shown that S_d^{timed} is the demander's best response to S_s^{timed}. In a subgame where $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$ and $t_n = 0$, strategy S_d^{timed} has remaining payoff

$$\pi_d^{(x_n,p_n,0)}(S_s^{timed}, S_d^{timed}) = v_d(1) - v_d(x_n) - [p^{contr} - p_n]$$

because the exchange is completed immediately.

Now, let us analyze the remaining payoff of an arbitrarily chosen strategy S'_d for the demander. Let the path of play induced by S'_d against S_s^{timed} be defined by $p_{n+1}, p_{n+2}, ...$ and $t_{n+1}, t_{n+2}, ...$ where $p_0 = t_0 = 0$ and $\forall k \in \{n+1, n+2, ...\}, p_k \leq p_{k+1}, t_k \leq t_{k+1}$. This model incorporates the possibility that for some q, $p_q = p_{q+1} = ... = \lim_{M\to\infty} p_M$ or $t_q = t_{q+1} = ... = \lim_{M\to\infty} t_M$, i.e. the payments can stop at some finite step, or the exchange can stop at some (possibly finite) time. Note that the supplier, according to S_s^{timed}, always moves to x_{k+1} immediately after the demander has moved to p_k, i.e. at time t_k.

$$
\begin{aligned}
&\pi_d^{(x_n,p_n,0)}(S_s^{timed}, S'_d) \\
&= \sum_{k=n}^{\infty} f_d^v(t_k)[v_d(x_{k+1}) - v_d(x_k)] - f_d^p(t_{k+1})[p_{k+1} - p_k] \\
&\leq \sum_{k=n}^{\infty} f_d^p(t_k)[v_d(x_{k+1}) - v_d(x_k)] - f_d^p(t_{k+1})[p_{k+1} - p_k] \\
&= \lim_{M\to\infty} \sum_{k=n}^{M} f_d^p(t_k)[v_d(x_{k+1}) - v_d(x_k)] - f_d^p(t_{k+1})[p_{k+1} - p_k] \\
&= \lim_{M\to\infty} -f_d^p(t_n)v_d(x_n) + f_d^p(t_M)v_d(x_{M+1}) + f_d^p(t_{n+1})p_n - f_d^p(t_{M+1})p_{M+1} \\
&\quad + \sum_{k=n}^{M-1} [f_d^p(t_k) - f_d^p(t_{k+1})]v_d(x_{k+1}) - [f_d^p(t_{k+1}) - f_d^p(t_{k+2})]p_{k+1}
\end{aligned}
$$

Now we use the fact $v_d(x_{k+1}) = v_d(max\{x \in X | p^{min}(x) \leq p_k\}) = v_d(max\{x \in X | p^{contr} - v_d(1) + v_d(x) \leq p_k\}) \leq p_k - p^{contr} + v_d(1)$.

$$
\begin{aligned}
&\leq \lim_{M\to\infty} -f_d^p(t_n)v_d(x_n) + f_d^p(t_M)v_d(x_{M+1}) + f_d^p(t_{n+1})p_n - f_d^p(t_{M+1})p_{M+1} \\
&\quad + \sum_{k=n}^{M-1} [f_d^p(t_k) - f_d^p(t_{k+1})][p_k - p^{contr} + v_d(1)] - [f_d^p(t_{k+1}) - f_d^p(t_{k+2})]p_{k+1} \\
&= \lim_{M\to\infty} -f_d^p(t_n)v_d(x_n) + f_d^p(t_M)v_d(x_{M+1}) + f_d^p(t_{n+1})p_n - f_d^p(t_{M+1})p_{M+1} \\
&\quad + \sum_{k=n}^{M-1} [f_d^p(t_k) - f_d^p(t_{k+1})][v_d(1) - p^{contr}] \\
&\quad + \sum_{k=n}^{M-1} [f_d^p(t_k) - f_d^p(t_{k+1})]p_k - [f_d^p(t_{k+1}) - f_d^p(t_{k+2})]p_{k+1} \\
&= \lim_{M\to\infty} -f_d^p(t_n)v_d(x_n) + f_d^p(t_M)v_d(x_{M+1}) + f_d^p(t_{n+1})p_n - f_d^p(t_{M+1})p_{M+1} \\
&\quad + [f_d^p(t_n) - f_d^p(t_M)][v_d(1) - p^{contr}]
\end{aligned}
$$

$$
\begin{aligned}
& +[f_d^p(t_n) - f_d^p(t_{n+1})]p_n - [f_d^p(t_M) - f_d^p(t_{M+1})]p_M \\
\leq \; & \lim_{M\to\infty} -f_d^p(t_n)v_d(x_n) + f_d^p(t_M)[p_M - p^{contr} + v_d(1)] \\
& +f_d^p(t_{n+1})p_n - f_d^p(t_{M+1})p_{M+1} \\
& +[f_d^p(t_n) - f_d^p(t_M)][v_d(1) - p^{contr}] \\
& +[f_d^p(t_n) - f_d^p(t_{n+1})]p_n - [f_d^p(t_M) - f_d^p(t_{M+1})]p_M \\
= \; & \lim_{M\to\infty} f_d^p(t_n)[-v_d(x_n) - p^{contr} + v_d(1) + p_n] + f_d^p(t_{M+1})[\underbrace{p_M - p_{M+1}}_{\leq 0}] \\
\leq \; & f_d^p(t_n)[-v_d(x_n) - p^{contr} + v_d(1) + p_n] \\
= \; & -v_d(x_n) - p^{contr} + v_d(1) + p_n \\
= \; & \pi_d^{(x_n,p_n,0)}(S_s^{timed}, S_d^{timed})
\end{aligned}
$$

Because the remaining payoff to an arbitrary strategy S'_d is no better than the remaining payoff to S_d^{timed}, the latter is a best response to S_s^{timed}. □

Proof. (Theorem 5.6). For example, in a subgame (x_n, p_n, t_n) where $t_n > 0$, $f_s^p(t_n) = 0$, $f_s^v(t_n) > 0$, $v_s(x_n) < v_s(1)$, and $p^{min}(x_n) \leq p_n \leq p^{max}(x_n) < p^{contr}$, the supplier is not motivated to proceed immediately because no payment by the demander can compensate for any cost incurred by the supplier's delivering (because $f_s^p(t) = 0$ for all $t \geq t_n$). Therefore, S_s^{timed} is not the supplier's best strategy in the subgame. □

Proof. (Theorem 5.7). The proof of Theorem 5.3 suffices to show that the exchange is completed in a minimal, finite number of steps using strategies $S_{s_{un}}^{timed}$ and S_d^{timed} because these strategies are identical to $S_{s_{un}}$ and S_d except that the former strategies explicitly specify *immediate* moves. Due to this prescription, the exchange is completed immediately, i.e. for all n, $t_n = 0$.

In the proof of Theorem 5.5 it was shown that S_s^{timed} is a best response to S_d^{timed} in each subgame (x_n, p_n, t_n) where $x_n \in [0, 1]$, $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$, and $t_n = 0$.

That proof can be used to show the same fact for $S_{s_{un}}^{timed}$ and S_d^{timed} by just replacing $S_{s_{un}}^{timed}$ in place of S_s^{timed}.

By following strategies $S_{s_{un}}^{timed}$ and S_d^{timed}, only subgames of this type are reached. Thus the equilibrium will remain an equilibrium in each subgame that is reached. What remains to be shown is that S_d^{timed} is the demander's best response to $S_{s_{un}}^{timed}$ in each subgame (x_n, p_n, t_n) where $x_n \in [0,1]$, $p^{min}(x_n) \leq p_n \leq p^{max}(x_n)$, and $t_n = 0$.

In a subgame of the mentioned type, S_d^{timed} has remaining payoff

$$\pi_d^{(x_n,p_n,0)}(S_{s_{un}}^{timed}, S_d^{timed}) = v_d(1) - v_d(x_n) - [p^{contr} - p_n]$$

because the exchange is completed immediately.

Now, let us analyze the remaining payoff of an arbitrarily chosen strategy S_d' for the demander. Let the path of play induced by S_d' against $S_{s_{un}}^{timed}$ be defined by $p_{n+1}, p_{n+2}, \ldots$ and $t_{n+1}, t_{n+2}, \ldots$ where $p_0 = t_0 = 0$ and $\forall k \in \{n+1, n+2, \ldots, N-1\}, p_k \leq p_{k+1}, t_k \leq t_{k+1}$. This model allows for the possibility that for some q, $p_q = p_{q+1} = \ldots = \lim_{M\to\infty} p_M$ or $t_q = t_{q+1} = \ldots = \lim_{M\to\infty} t_M$, i.e. the payments can stop at some finite step, or the exchange can stop at some (possibly finite) time. Note that the supplier, according to $S_{s_{un}}^{timed}$, always moves to x_{k+1} immediately after the demander has moved to p_k, i.e. at time t_k.

$$\begin{aligned}
&\pi_d^{(x_n,p_n,0)}(S_{s_{un}}^{timed}, S_d') \\
&= \sum_{k=n}^{\infty} f_d^v(t_k)[v_d(x_{k+1}) - v_d(x_k)] - f_d^p(t_{k+1})[p_{k+1} - p_k] \\
&\leq \sum_{k=n}^{\infty} f_d^p(t_k)[v_d(x_{k+1}) - v_d(x_k)] - f_d^p(t_{k+1})[p_{k+1} - p_k] \\
&= \lim_{M\to\infty} \sum_{k=n}^{M} f_d^p(t_k)[v_d(x_{k+1}) - v_d(x_k)] - f_d^p(t_{k+1})[p_{k+1} - p_k] \\
&= \lim_{M\to\infty} -f_d^p(t_n)v_d(x_n) + f_d^p(t_M)v_d(x_{M+1}) + f_d^p(t_{n+1})p_n - f_d^p(t_{M+1})p_{M+1} \\
&\quad + \sum_{k=n}^{M-1} [f_d^p(t_k) - f_d^p(t_{k+1})]v_d(x_{k+1}) - [f_d^p(t_{k+1}) - f_d^p(t_{k+2})]p_{k+1}
\end{aligned}$$

Now we can use knowledge of how $S_{s_{un}}^{timed}$ prescribes $v_d(x_{k+1})$. If the max-operator has a defined value, $v_d(x_{k+1}) = v_d(max\{x \in X | p^{min}(x) \leq p_k\}) = v_d(max\{x \in X | p^{contr} -$

$v_d(1) + v_d(x) \leq p_k\}) \leq p_k - p^{contr} + v_d(1)$. If the max-operator has an undefined value, $v_d(x_{k+1}) = v_d(max(x_k, sup\{x \in X|p^{min}(x) \leq p_k\} - \underline{\epsilon})) = v_d(max(x_k, sup\{x \in X|p^{contr} - v_d(1) + v_d(x) \leq p_k\} - \underline{\epsilon})) < p_k - p^{contr} + v_d(1)$. So, in either case, $v_d(x_{k+1}) \leq p_k - p^{contr} + v_d(1)$. Now we can substitute this:

$$
\begin{aligned}
\leq\ & \lim_{M\to\infty} -f_d^p(t_n)v_d(x_n) + f_d^p(t_M)v_d(x_{M+1}) + f_d^p(t_{n+1})p_n - f_d^p(t_{M+1})p_{M+1} \\
& + \sum_{k=n}^{M-1} [f_d^p(t_k) - f_d^p(t_{k+1})][p_k - p^{contr} + v_d(1)] - [f_d^p(t_{k+1}) - f_d^p(t_{k+2})]p_{k+1} \\
=\ & \lim_{M\to\infty} -f_d^p(t_n)v_d(x_n) + f_d^p(t_M)v_d(x_{M+1}) + f_d^p(t_{n+1})p_n - f_d^p(t_{M+1})p_{M+1} \\
& + \sum_{k=n}^{M-1} [f_d^p(t_k) - f_d^p(t_{k+1})][v_d(1) - p^{contr}] \\
& + \sum_{k=n}^{M-1} [f_d^p(t_k) - f_d^p(t_{k+1})]p_k - [f_d^p(t_{k+1}) - f_d^p(t_{k+2})]p_{k+1} \\
=\ & \lim_{M\to\infty} -f_d^p(t_n)v_d(x_n) + f_d^p(t_M)v_d(x_{M+1}) + f_d^p(t_{n+1})p_n - f_d^p(t_{M+1})p_{M+1} \\
& +[f_d^p(t_n) - f_d^p(t_M)][v_d(1) - p^{contr}] \\
& +[f_d^p(t_n) - f_d^p(t_{n+1})]p_n - [f_d^p(t_M) - f_d^p(t_{M+1})]p_M \\
\leq\ & \lim_{M\to\infty} -f_d^p(t_n)v_d(x_n) + f_d^p(t_M)[p_M - p^{contr} + v_d(1)] \\
& +f_d^p(t_{n+1})p_n - f_d^p(t_{M+1})p_{M+1} \\
& +[f_d^p(t_n) - f_d^p(t_M)][v_d(1) - p^{contr}] \\
& +[f_d^p(t_n) - f_d^p(t_{n+1})]p_n - [f_d^p(t_M) - f_d^p(t_{M+1})]p_M \\
=\ & \lim_{M\to\infty} f_d^p(t_n)[-v_d(x_n) - p^{contr} + v_d(1) + p_n] + f_d^p(t_{M+1})[\underbrace{p_M - p_{M+1}}_{\leq 0}] \\
\leq\ & f_d^p(t_n)[-v_d(x_n) - p^{contr} + v_d(1) + p_n] \\
=\ & -v_d(x_n) - p^{contr} + v_d(1) + p_n \\
=\ & \pi_d^{(x_n,p_n,0)}(S_{sun}^{timed}, S_d^{timed})
\end{aligned}
$$

Because the remaining payoff to an arbitrary strategy S'_d is no better than the remaining payoff to S_d^{timed}, the latter is a best response to S_{sun}^{timed}. □

Proof. (Theorem 5.8). The counterexample presented as proof of Theorem 5.6 applies. □

Proof. (Theorem 5.9). First, the case where the supplier delays is proven. Assume—for contradiction—that in a subgame (x_n, p_n, t_n), where $p^{max}(x_n) < p_n \leq p^{max'}(x_n)$, there is a Nash equilibrium (by some arbitrary strategies S'_s and S'_d) that remains an equilibrium in every subgame that is reached and results in reaching $(1, p^{contr})$ when $f^p_s(t) > 0$, i.e. $\lim_{M\to\infty} f^p_s(t_M) > 0$. Then there is an equilibrium path defined by $x_{n+1}, x_{n+2}, ...$ and $p_{n+1}, p_{n+2}, ...$ and $t_{n+1}, t_{n+2}, ...$ where $\lim_{M\to\infty} t_M = T$ (i.e. finite because otherwise $\lim_{M\to\infty} f^p_s(t_M) = 0$), $\lim_{M\to\infty} f^v_s(t_M) > 0$, $\lim_{M\to\infty} x_M = 1$, and $\lim_{M\to\infty} p_M = p^{contr}$. Let us define a shorthand notation $s(q)$:

$$\begin{aligned}
s(q) &= \pi_s^{(x_q,p_q,t_q)}(S'_s, S'_d) \\
&= \sum_{k=q}^{\infty} f^p_s(t_{k+1})[p_{k+1} - p_k] - f^v_s(t_{k+1})[v_s(x_{k+1}) - v_s(x_k)] \\
\Leftrightarrow & \\
p_q &= p_{q+1} - \frac{s(q)}{f^p_s(t_{q+1})} - \frac{f^v_s(t_{q+1})[v_s(x_{q+1}) - v_s(x_q)]}{f^p_s(t_{q+1})} + \frac{s(q+1)}{f^p_s(t_{q+1})} \\
&\leq p_{q+1} - [v_s(x_{q+1}) - v_s(x_q)] + \frac{s(q+1) - s(q)}{f^p_s(t_{q+1})}
\end{aligned}$$

Applying this recursively for all q from n to infinity gives

$$\begin{aligned}
p_n &\leq \lim_{M\to\infty} p_{M+1} + \sum_{q=n}^{M} -[v_s(x_{q+1}) - v_s(x_q)] + \sum_{q=n}^{M} \frac{s(q+1) - s(q)}{f^p_s(t_{q+1})} \\
&= \lim_{M\to\infty} p_{M+1} - v_s(x_{M+1}) + v_s(x_n) + \sum_{q=n}^{M} \frac{s(q+1) - s(q)}{f^p_s(t_{q+1})} \\
&= \lim_{M\to\infty} p^{contr} - v_s(1) + v_s(x_n) \\
&\quad - \frac{s(n)}{f^p_s(t_{n+1})} + \frac{s(M+1)}{f^p_s(t_{M+1})} + \sum_{q=n+1}^{M} [\frac{1}{f^p_s(t_q)} - \frac{1}{f^p_s(t_{q+1})}] s(q)
\end{aligned}$$

On the other hand, by postponing infinitely, the supplier does not incur any costs from delivering because $\lim_{t\to\infty} f^v_s(t) = 0$. Thus,

$$\pi_s^{(x_q,p_q,t_q)}(S_s^{postpone}, S'_d) \geq f^p_s(t_{q+1})[p_{q+1} - p_q] \geq 0$$

For the equilibrium (S'_s, S'_d) to remain an equilibrium in every subgame that is reached, it has to hold in such subgames that following S'_s is no worse than postponing infinitely:

$$\forall q \geq n, s(q) = \pi_s^{(x_q,p_q,t_q)}(S'_s, S'_d) \geq \pi_s^{(x_q,p_q,t_q)}(S_s^{postpone}, S'_d) \geq 0$$

Now we can use the fact $s(q) \geq 0$:

$$\begin{aligned}
p_n &\leq \lim_{M\to\infty} p^{contr} - v_s(1) + v_s(x_n) \\
&\quad - \underbrace{\frac{s(n)}{f_s^p(t_{n+1})}}_{\geq 0} + \frac{s(M+1)}{f_s^p(t_{M+1})} + \sum_{q=n+1}^{M} \underbrace{[\frac{1}{f_s^p(t_q)} - \frac{1}{f_s^p(t_{q+1})}]}_{\leq 0} \underbrace{s(q)}_{\geq 0} \\
&\leq p^{contr} - v_s(1) + v_s(x_n) + \lim_{M\to\infty} \frac{s(M+1)}{f_s^p(t_{M+1})} \\
&= p^{contr} - v_s(1) + v_s(x_n) + \lim_{M\to\infty} \frac{s(M)}{f_s^p(t_M)} \\
&= p^{contr} - v_s(1) + v_s(x_n) \\
&\quad + \lim_{M\to\infty} \frac{\sum_{k=M}^{\infty} f_s^p(t_{k+1})[p_{k+1} - p_k] - f_s^v(t_{k+1})[v_s(x_{k+1}) - v_s(x_k)]}{f_s^p(t_M)} \\
&\leq p^{contr} - v_s(1) + v_s(x_n) + \lim_{M\to\infty} \frac{\sum_{k=M}^{\infty}[p_{k+1} - p_k]}{f_s^p(t_M)} \\
&= p^{contr} - v_s(1) + v_s(x_n) + \lim_{M\to\infty} \frac{\lim_{N\to\infty} p_{N+1} - p_M}{f_s^p(t_M)} \\
&= p^{contr} - v_s(1) + v_s(x_n) + \frac{p^{contr} - p^{contr}}{\lim_{M\to\infty} f_s^p(t_M)} \\
&= p^{contr} - v_s(1) + v_s(x_n) \\
&= p^{max}(x_n)
\end{aligned}$$

This contradicts the assumption $p^{max}(x_n) < p_n$. Therefore no such equilibrium exists.

Next, we analyze the case where the demander postpones. Assume—for contradiction—that in a subgame (x_n, p_n, t_n), where $p^{min'}(x_n) \leq p_n < p^{min}(x_n)$, there is a Nash equilibrium (by some arbitrary strategies S'_s and S'_d) that remains an equilibrium in every subgame that is reached and results in reaching $(1, p^{contr})$ when $f_d^v(t) > 0$, i.e. $\lim_{M\to\infty} f_d^v(t_M) > 0$. Then there is an equilibrium path defined by $x_{n+1}, x_{n+2}, ...$ and $p_{n+1}, p_{n+2}, ...$ and $t_{n+1}, t_{n+2}, ...$ where $\lim_{M\to\infty} t_M = T$ (i.e. finite because otherwise $\lim_{M\to\infty} f_d^v(t_M) = 0$), $\lim_{M\to\infty} f_d^p(t_M) > 0$, $\lim_{M\to\infty} x_M = 1$, and $\lim_{M\to\infty} p_M = p^{contr}$. Let us define a shorthand $z(q)$:

$$\begin{aligned}
z(q) &= \pi_d^{(x_q,p_q,t_q)}(S'_s, S'_d) \\
&= \sum_{k=q}^{\infty} f_d^v(t_{k+1})[v_d(x_{k+1}) - v_d(x_k)] - f_d^p(t_{k+1})[p_{k+1} - p_k]
\end{aligned}$$

$\Leftrightarrow$

$$\begin{aligned} p_q &= p_{q+1} + \frac{z(q)}{f_d^p(t_{q+1})} - \frac{f_d^v(t_{q+1})[v_d(x_{q+1}) - v_d(x_q)]}{f_d^p(t_{q+1})} - \frac{z(q+1)}{f_d^p(t_{q+1})} \\ &\geq p_{q+1} - [v_d(x_{q+1}) - v_d(x_q)] + \frac{z(q) - z(q+1)}{f_d^p(t_{q+1})} \end{aligned}$$

Applying this recursively for all q from n to infinity gives

$$\begin{aligned} p_n &\geq \lim_{M\to\infty} p_{M+1} + \sum_{q=n}^{M} -[v_d(x_{q+1}) - v_d(x_q)] + \sum_{q=n}^{M} \frac{z(q) - z(q+1)}{f_d^p(t_{q+1})} \\ &= \lim_{M\to\infty} p_{M+1} - v_d(x_{M+1}) + v_d(x_n) + \sum_{q=n}^{M} \frac{z(q) - z(q+1)}{f_d^p(t_{q+1})} \\ &= \lim_{M\to\infty} p^{contr} - v_d(1) + v_d(x_n) \\ &\quad + \frac{z(n)}{f_d^p(t_{n+1})} - \frac{z(M+1)}{f_d^p(t_{M+1})} + \sum_{q=n+1}^{M} [\frac{1}{f_d^p(t_{q+1})} - \frac{1}{f_d^p(t_q)}] z(q) \end{aligned}$$

On the other hand, by postponing infinitely, the demander does not incur any costs from paying because $\lim_{t\to\infty} f_d^p(t) = 0$. Thus,

$$\pi_d^{(x_q,p_q,t_q)}(S'_s, S_d^{postpone}) \geq f_d^p(t_{q+1})[v_d(x_{q+1}) - v_d(x_q)] \geq 0$$

For the equilibrium (S'_s, S'_d) to remain an equilibrium in every subgame that is reached, it has to hold in such subgames that following S'_d is no worse than postponing infinitely:

$$\forall q \geq n, z(q) = \pi_d^{(x_q,p_q,t_q)}(S'_s, S'_d) \geq \pi_d^{(x_q,p_q,t_q)}(S'_s, S_d^{postpone}) \geq 0$$

Now we can use the fact $z(q) \geq 0$:

$$\begin{aligned} p_n &\geq \lim_{M\to\infty} p^{contr} - v_d(1) + v_d(x_n) \\ &\quad + \underbrace{\frac{z(n)}{f_d^p(t_{n+1})}}_{\geq 0} - \frac{z(M+1)}{f_d^p(t_{M+1})} + \sum_{q=n+1}^{M} \underbrace{[\frac{1}{f_d^p(t_{q+1})} - \frac{1}{f_d^p(t_q)}]}_{\geq 0} \underbrace{z(q)}_{\geq 0} \\ &\geq p^{contr} - v_d(1) + v_d(x_n) - \lim_{M\to\infty} \frac{z(M+1)}{f_d^p(t_{M+1})} \\ &= p^{contr} - v_d(1) + v_d(x_n) - \lim_{M\to\infty} \frac{z(M)}{f_d^p(t_M)} \end{aligned}$$

$$\begin{aligned}
&= p^{contr} - v_d(1) + v_d(x_n) \\
&\quad + \lim_{M\to\infty} \frac{\sum_{k=M}^{\infty} f_d^v(t_{k+1})[v_d(x_{k+1}) - v_d(x_k)] - f_d^p(t_{k+1})[p_{k+1} - p_k]}{f_s^p(t_M)} \\
&\geq p^{contr} - v_d(1) + v_d(x_n) + \lim_{M\to\infty} \frac{\sum_{k=M}^{\infty} p_k - p_{k+1}}{f_s^p(t_M)} \\
&= p^{contr} - v_d(1) + v_d(x_n) + \lim_{M\to\infty} \frac{\lim_{N\to\infty} p_M - p_{N+1}}{f_s^p(t_M)} \\
&= p^{contr} - v_d(1) + v_d(x_n) + \frac{p^{contr} - p^{contr}}{\lim_{M\to\infty} f_s^p(t_M)} \\
&= p^{contr} - v_d(1) + v_d(x_n) \\
&= p^{min}(x_n)
\end{aligned}$$

This contradicts the assumption $p^{min}(x_n) > p_n$. Thus no such equilibrium exists. □

Proof. (Theorem 5.10). Steps 1, 3, 5, 7, and 9 take $O(1)$ time. Steps 2 and 4 take $O(|C|)$ time. Steps 6 and 8 take $O(|C|^2)$ time. Therefore the entire algorithm always terminates in $O(|C|^2)$ time.

What remains to be proven is that the algorithm always finds a safe sequence if one exists. The check in step 3 does not violate this property, because if $p_{init}^{max} < 0$ or $p_{init}^{min} > 0$, the agents' strategies S_s and S_d specify that both agents will exit at point $(x_0, p_0) = (0, 0)$, and thus a safe exchange is impossible.

To show that the algorithm finds a sequence where $min(p^{max}(x), p^{contr}) \geq p^{min}(x')$ for all consecutive x and x' if such a sequence exists, it suffices to show that $p^{max}(x) \geq p^{min}(x')$ for all consecutive x and x'. This is because it has to hold that $p^{contr} \geq p^{min}(x')$ for all x' for a safe sequence to exist.

Let us call the sequence of goods that Algorithm 5.1 suggests s. In case the algorithm terminated saying "NO SOLUTION", s is only a partial sequence. Assume (for contradiction) that $\exists s'$ s.t. s' is a safe sequence (i.e. acceptable), but s is not. So s' fulfills the requirement that for all subsequent x and x', $p^{max}(x) \geq p^{min}(x')$, but s does not.

First, we prove that s' can be reordered (while maintaining acceptability) s.t. the chunks with $\Delta p_c^{max} - \Delta p_c^{min} \geq 0$ lie before those with $\Delta p_c^{max} - \Delta p_c^{min} < 0$. Take any pair of chunks c_i, c_j s.t. $\Delta p^{max}_{c_i} - \Delta p^{min}_{c_i} < 0$, $\Delta p^{max}_{c_j} - \Delta p^{min}_{c_j} \geq 0$, and c_i is assigned right before c_j in s'. Now we can swap c_i and c_j. This will not affect $p^{max}(x)$ or $p^{min}(x)$ before or after the pair. c_j can be moved before c_i, because c_i only made $p^{max}(x) - p^{min}(x)$ smaller. c_i can be moved after c_j, because c_j can only make $p^{max}(x) - p^{min}(x)$ larger (or same). We apply these swaps until the desired property holds. Let us call this new sequence s''.

Let us denote the sequence of chunks with $\Delta p^{max}_c - \Delta p^{min}_c \geq 0$ in s by s_{pos}, and the corresponding part of sequence s'' by s''_{pos}. We show that s''_{pos} can be converted into s_{pos} without losing acceptability. Let i be the position at which s_{pos} and s''_{pos} first differ. Let s_i be the item in s_{pos} at position i, and let s''_i be the item in s''_{pos} at position i. Let $k > i$ be the position where s_i was allocated in s''_{pos}. Now we can swap s''_i and s''_k in s''_{pos}. s''_k can be moved up to position i, because our algorithm allocated it there after having checked feasibility. s''_i can be moved down to position k, because the chunks in s''_{pos} between i and k could only have increased $p^{max}(x) - p^{min}(x)$. The presented swap can only increase $p^{max}(x) - p^{min}(x)$ for position between i and k, so no chunk that was feasible earlier can become infeasible. We can apply these swaps until $s''_{pos} = s_{pos}$.

Let us denote the sequence of chunks with $\Delta p^{max}_c - \Delta p^{min}_c < 0$ in s by s_{neg}, and the corresponding part of sequence s'' by s''_{neg}. Now we show that s''_{neg} can be converted into s_{neg} without losing acceptability. Let i be the last position at which s_{neg} and s''_{neg} differ. Let s_i be the item in s_{neg} at position i, and let s''_i be the item in s''_{neg} at position i. Let $k < i$ be the position where s_i was allocated in s''_{neg}. Now we can swap s''_i and s''_k in s''_{neg}. s''_k can be moved down to position i, because our algorithm allocated it there after having checked feasibility. s''_i can be moved up to position k, because the chunks in s''_{neg} between k and i could only have decreased $p^{max}(x) - p^{min}(x)$. The presented swap can only increase $p^{max}(x) - p^{min}(x)$ for position between k and i, so no chunk that was feasible earlier can become infeasible. We can apply these swaps until $s''_{neg} = s_{neg}$.

So, s is a safe sequence iff s'' is. Earlier we showed that s'' is a safe sequence iff s' is. In other words, s is a safe sequence iff s' is. This contradicts our assumption and completes the proof. $\square$

REFERENCES

[Abreu and Rubinstein, 1988] Dilip Abreu and Ariel Rubinstein. The structure of Nash equilibrium in repeated games with finite automata. *Econometrica*, 56(6):1259–1281, 1988.

[Agorics, Inc., 1996] Agorics, Inc. Going, going, gone! A survey of auction types. *http://www.agorics.com /agorics /auctions /bibliography.html*, 1996.

[Arrow, 1963] Kenneth Arrow. *Social choice and individual values.* New Haven: Cowles Foundation, 2nd edition, 1963. 1st edition 1951.

[Aumann, 1959] R Aumann. Acceptable points in general cooperative n-person games. volume IV of *Contributions to the Theory of Games.* Princeton University Press, 1959.

[Axelrod, 1984] Robert Axelrod. *The Evolution of Cooperation.* Basic Books, 1984.

[Bernheim and Whinston, 1987] B Douglas Bernheim and Michael D Whinston. Coalition-proof Nash equilibria: II applications. *Journal of Economic Theory*, 42(1):13–29, June 1987.

[Bernheim *et al.*, 1987] B Douglas Bernheim, Bezalel Peleg, and Michael D Whinston. Coalition-proof Nash equilibria: I concepts. *Journal of Economic Theory*, 42(1):1–12, June 1987.

[Binmore, 1994] Ken Binmore. *Game Theory and the Social Contract: Playing Fair*, volume 1. MIT Press, 1994.

[Boddy and Dean, 1989] Mark Boddy and Thomas Dean. Solving time-dependent planning problems. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 979–984, Detroit, MI, August 1989.

[Boddy and Dean, 1994] Mark Boddy and Thomas Dean. Deliberation scheduling for problem solving in time-constrained environments. *Artificial Intelligence*, 67:245–285, 1994.

[Bond and Gasser, 1988] Alan H. Bond and Les Gasser. *Readings in Distributed Artificial Intelligence.* Morgan Kaufmann Publishers, San Mateo, CA, 1988.

[Brainov, 1994] Sviatoslav Brainov. Deviation-proof plans in open multiagent environments. In *Proceedings of the 11th European Conference on Artificial Intelligence*, pages 274–278, August 1994.

[Brooks, 1986] Rodney Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23, 1986.

[Burke and Prosser, 1994] Peter Burke and Patrick Prosser. The distributed asynchronous scheduler. In Monte Zweben and Mark Fox, editors, *Intelligent Scheduling*, pages 309–339. Morgan Kaufmann, 1994.

[Calamari and Perillo, 1977] John D Calamari and Joseph M Perillo. *The Law of Contracts*. West Publishing Co., 2nd edition, 1977.

[Charnes and Kortanek, 1966] A Charnes and K O Kortanek. On balanced sets, cores, and linear programming. Technical Report 12, Cornell Univ., Dept. of Industrial Eng. and Operations Res., Ithaca, NY, 1966.

[Cheng and Wellman, 1995] John Q. Cheng and Michael P. Wellman. The walras algorithm: A convergent distributed implementation of general equilibrium outcomes. 1995. Submitted.

[Clarke, 1971] E H Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

[Cohen and Levesque, 1987] Philip R. Cohen and Hector J. Levesque. Intention = choice + commitment. In *Proceedings of the National Conference on Artificial Intelligence*, pages 410–415, July 1987.

[Cohen and Levesque, 1990] Philip R. Cohen and Hector J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(2-3):213–261, 1990.

[Conry *et al.*, 1991] Susan E Conry, K Kuwabara, Victor R Lesser, and R A Meyer. Multistage negotiation for distributed satisfaction. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6):1462–1477, 1991.

[D'Ambrosio, 1989] Bruce D'Ambrosio. Resource bounded-agents in an uncertain world. In *Proceedings of the Workshop on Real-Time Artificial Intelligence Problems*, IJCAI-89, Detroit, August 1989.

[Davis and Smith, 1981] Randall Davis and Reid G. Smith. Negotiation as a metaphor for distributed problem solving. AI Memo 624, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, May 1981. (Also published in *Readings in Distributed Artificial Intelligence*, Alan H. Bond and Les Gasser, editors, pages 333–356, Morgan Kaufmann, 1988.).

[Davis and Smith, 1983] Randall Davis and Reid G. Smith. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20:63–109, 1983.

[Dean and Boddy, 1988] Thomas Dean and Mark Boddy. An analysis of time-dependent planning. In *Proceedings of the National Conference on Artificial Intelligence*, pages 49–54, St. Paul, MN, August 1988.

[Decker and Lesser, 1993a] Keith Decker and Victor Lesser. A one-shot dynamic coordination algorithm for distributed sensor networks. In *Proc. Eleventh National Conference on Artificial Intelligence (AAAI-93)*, July 1993.

[Decker and Lesser, 1993b] Keith Decker and Victor Lesser. Quantitative modeling of complex computational task environment. In *Proc. Eleventh National Conference on Artificial Intelligence (AAAI-93)*, July 1993.

[Decker and Lesser, 1995] Keith Decker and Victor R Lesser. Designing a family of coordination algorithms. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 73–80, San Francisco, CA, June 1995.

[Decker *et al.*, 1989] Keith S. Decker, Edmund H. Durfee, and Victor R. Lesser. The evaluation of research in cooperative distributed problem solving. In Michael N. Huhns and Les Gasser, editors, *Distributed Artificial Intelligence*, volume 2 of *Research Notes in Artificial Intelligence*. Pitman, 1989.

[Drexler and Miller, 1988] K. Eric Drexler and Mark S. Miller. Incentive engineering for computational resource management. In B. A. Huberman, editor, *The Ecology of Computation*. North-Holland, 1988.

[Durfee and Lesser, 1989] Edmund H. Durfee and Victor R. Lesser. Negotiating task decomposition and allocation using partial global planning. In Les Gasser and Michael N. Huhns, editors, *Distributed Artificial Intelligence*, volume 2 of *Research Notes in Artificial Intelligence*, pages 229–243. Pitman, 1989.

[Durfee and Lesser, 1991] Edmund H. Durfee and Victor R. Lesser. Partial global planning : A coordination framework for distributed hypothesis formation. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(5):1167–1183, September 1991.

[Durfee *et al.*, 1989] E Durfee, V Lesser, and D Corkill. Cooperative distributed problem solving. In A Barr, P Cohen, and E Feigenbaum, editors, *The Handbook of Artificial Intelligence*, volume IV. Addison Wesley, 1989.

[Durfee *et al.*, 1993] Edmund H Durfee, J Lee, and Piotr J Gmytrasiewicz. Overeager reciprocal rationality and mixed strategy equilibria. In *AAAI-93*, pages 225–230, Washington DC, July 1993.

[Edelman, 1996] Joe Edelman. Web page mention of the use of Vickrey auction in AgentTCL. *http: //www.cs.dartmouth.edu/ ~jxe/ resume.html*, 1996.

[Ephrati and Rosenschein, 1991] Eithan Ephrati and Jeffrey S Rosenschein. The Clarke tax as a consensus mechanism among automated agents. In *Proceedings of the National Conference on Artificial Intelligence*, pages 173–178, Anaheim, CA, 1991.

[Ephrati and Rosenschein, 1993a] Eithan Ephrati and Jeffrey S Rosenschein. Multi-agent planning as a dynamic search for social consensus. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 423–429, Chambery, France, 1993.

[Ephrati and Rosenschein, 1993b] Eithan Ephrati and Jeffrey S Rosenschein. Planning to please: following another agent's intended plan. *Group Decision and Negotiation*, 1993. To appear.

[Ephrati, 1993] Eithan Ephrati. *Planning and consensus among autonomous agents.* PhD thesis, Hebrew University, 1993.

[Ephrati, 1994] Eithan Ephrati. A non-manipulable meeting scheduling system. In *Proc. 13th International Distributed Artificial Intelligence Workshop*, Lake Quinalt, Washington, July 1994. AAAI Press Technical Report WS-94-02.

[Fagin *et al.*, 1995] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning About Knowledge.* MIT Press, 1995.

[Ferguson *et al.*, 1988] D. Ferguson, Y. Yemini, and C. Nikolau. Microeconomic algorithms for load balancing in distributed computer systems. In *Proceedings of the IEEE International Conference on Distributed Computer Systems*, pages 491–499, 1988.

[Finin *et al.*, 1992] Tim Finin, Rich Fritzson, and Don McKay. A language and protocol to support intelligent agent interoperability. In *Proc. of the CE & CALS Washington '92 Conference*, June 1992.

[Fischer *et al.*, 1995] Klaus Fischer, Jörg Müller, Markus Pischel, and Darius Schier. A model for cooperative transportation scheduling. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 109–116, San Francisco, CA, June 1995.

[Friedman, 1990] James W. Friedman. *Game Theory with Applications to Economics.* Oxford University Press, 1990.

[Friend, 1973] K E Friend. *An information processing approach to small group interaction in a coalition formation game.* PhD thesis, Carnegie-Mellon University, 1973.

[Fudenberg and Tirole, 1991] Drew Fudenberg and Jean Tirole. *Game Theory.* MIT Press, 1991.

[Garvey and Lesser, 1993] Alan Garvey and Victor Lesser. Design-to-time real-time scheduling. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(6):1491–1502, 1993.

[Garvey and Lesser, 1994] Alan Garvey and Victor Lesser. A survey of research in deliberative real-time artificial intelligence. *Real-Time Systems*, 6:317–347, 1994.

[Gasser and Huhns, 1989] Les Gasser and Michael N. Huhns, editors. *Distributed Artificial Intelligence Volume II.* Morgan Kaufmann, 1989.

[General Magic, Inc., 1994] General Magic, Inc. Telescript technology: The foundation for the electronic marketplace, 1994. White paper.

[Genesereth *et al.*, 1984] M. R. Genesereth, M. L. Ginsberg, and J. S. Rosenschein. Cooperation without communications. Technical Report 84-36, Stanford Heuristic Programming Project, Computer Science Department, Stanford University, Stanford, CA 94305, 1984. (Also published in *Readings in Distributed Artificial Intelligence*, Alan H. Bond and Les Gasser, editors, pages 220–226, Morgan Kaufmann, 1988.).

[Gibbard, 1973] A Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–602, 1973.

[Gilboa and Samet, 1989] Itzhak Gilboa and Dov Samet. Bounded versus unbounded rationality: The tyranny of the weak. *Games and Economic Behavior*, pages 213–221, 1989.

[Glance and Hogg, 1995] Natalie S. Glance and Tad Hogg. Dilemmas in computational societies. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 117–124, San Francisco, CA, 1995.

[Gmytrasiewicz and Durfee, 1992] P. J. Gmytrasiewicz and E. H. Durfee. Logic of knowledge and belief for recursive modeling: Preliminary report. In *Proceedings of the National Conference on Artificial Intelligence*, pages 628–634, July 1992.

[Gmytrasiewicz and Durfee, 1995] P. J. Gmytrasiewicz and E. H. Durfee. A rigorous, operational formalization of recursive modeling. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 125–132, San Francisco, CA, June 1995.

[Good, 1971] Irving Good. Twenty-seven principles of rationality. In V Godambe and D Sprott, editors, *Foundations of Statistical Inference*. Toronto: Holt, Rinehart, Winston, 1971.

[Grosz and Kraus, 1993] Barbara Grosz and Sarit Kraus. Collaborative plans for group activities. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 367–373, Chambery, France, 1993.

[Groves and Ledyard, 1977] Theodore Groves and John Ledyard. Optimal allocation of public goods: a solution to the 'free rider' problem. *Econometrica*, 45:783–809, 1977.

[Gu and Ishida, 1995] Cheng Gu and Toru Ishida. A quantitative analysis of the contract net protocol. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, page 449, San Francisco, CA, June 1995. Poster.

[Hansen and Zilberstein, 1996] Eric A Hansen and Shlomo Zilberstein. Monitoring the progress of anytime problem-solving. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1229–1234, Portland, OR, August 1996.

[Hardin, 1968] G Hardin. The tragedy of the commons. *Science*, 162:1243–1248, 1968.

[Heiner, 1983] Ronald A Heiner. The origin of predictable behavior. *American Economic Review*, 73:560–595, September 1983.

[Heiner, 1985] Ronald A Heiner. The origin of predictable behavior: Further modeling and applications. *American Economic Review*, 75(2):391–396, 1985.

[Hogg, 1995] Tad Hogg. Social dilemmas in computational ecosystems. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 711–716, Montreal, Canada, 1995.

[Horvitz, 1987] E.J. Horvitz. Reasoning about beliefs and actions under computational resource constraints. In *Proceedings of Third Workshop on Uncertainty in Artificial Intelligence*, pages 429–444, Seattle, Washington, July 1987. American Association for Artificial Intelligence. Also in L. Kanal, T. Levitt, and J. Lemmer, ed., *Uncertainty in Artificial Intelligence 3,* Elsevier, 1989, pps. 301-324.

[Horvitz, 1988] E.J. Horvitz. Reasoning under varying and uncertain resource constraints. In *Proceedings of the National Conference on Artificial Intelligence*, pages 111–116, St. Paul, MN, August 1988. Morgan Kaufmann, San Mateo, CA.

[Howard, 1971] Nigel Howard. *Paradoxes of Rationality: Theory of Metagames and Political Behavior.* The MIT Press, 1971.

[Huberman and Clearwater, 1995] Bernardo Huberman and Scott H Clearwater. A multi-agent system for controlling building environments. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 171–176, San Francisco, CA, June 1995.

[Huberman and Hogg, 1988] Bernardo A. Huberman and Tad Hogg. The behavior of computational ecologies. In B. A. Huberman, editor, *The Ecology of Computation.* North-Holland, 1988.

[Huhns, 1987] Michael Huhns, editor. *Distributed Artificial Intelligence.* Morgan Kaufmann, 1987.

[Kahan and Rapoport, 1984] James P Kahan and Amnon Rapoport. *Theories of Coalition Formation.* Lawrence Erlbaum Associates Publishers, 1984.

[Kalai and Stanford, 1988] Ehud Kalai and William Stanford. Finite rationality and interpersonal complexity in repeated games. *Econometrica*, pages 397–410, 1988.

[Kalai, 1985] Ehud Kalai. Solutions to the bargaining problem. In Leonid Hurwicz, David Schmeidler, and Hugo Sonnenschein, editors, *Social goals and social organization: Essays in memory of Elisha Pazner*, chapter 3, pages 77–106. Cambridge University Press, 1985.

[Kalakota and Whinston, 1996] Ravi Kalakota and Andrew B. Whinston. *Frontiers of Electronic Commerce.* Addison-Wesley Publishing Company, Inc, 1996.

[Ketchpel, 1994] Steven Ketchpel. Forming coalitions in the face of uncertain rewards. In *Proceedings of the National Conference on Artificial Intelligence*, pages 414–419, Seattle, WA, July 1994.

[Klusch and Shechory, 1996] Matthias Klusch and Onn Shechory. Coalitions formation among rational information agents. In *MAAMAW*, Eindhoven, The Netherlands, January 1996. Submitted.

[Kraus *et al.*, 1992] Sarit Kraus, Jonathan Wilkenfeld, and Gilad Zlotkin. Multiagent negotiation under time constraints. Technical Report 2975, University of Maryland, College Park, Computer Science Department, 1992.

[Kraus, 1993] Sarit Kraus. Agents contracting tasks in non-collaborative environments. In *Proc. Eleventh National Conference on Artificial Intelligence (AAAI-93)*, pages 243–248, July 1993.

[Kreps, 1990] David M Kreps. *A Course in Microeconomic Theory*. Princeton University Press, 1990.

[Kristol *et al.*, 1994] David M Kristol, Steven H Low, and Nicholas F Maxemchuk. Anonymous internet mercantile protocol. 1994. Submitted.

[Kurose and Simha, 1989] James F. Kurose and Rahul Simha. A microeconomic approach to optimal resource allocation in distributed computer systems. *IEEE Transactions on Computers*, 38(5):705–717, 1989.

[Kuwabara and Ishida, 1992] Kazuhiro Kuwabara and Toru Ishida. A symbiotic approach to distributed resource allocation: Toward coordinated balancing. In *Proc. MAAMAW-92*, 1992.

[Kuwabara *et al.*,] Kazuhiro Kuwabara, Toru Ishida, and Yoshiyasu Nishibe. Market-based distributed resource allocation: Equilibratory approach.

[Lander and Lesser, 1993] Susan Lander and Victor Lesser. Understanding the role of negotiation in distributed search among heterogeneous agents. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 438–444, Chambery, France, 1993.

[Lax and Sebenius, 1981] D A Lax and J K Sebenius. Insecure contracts and resource development. *Public Policy*, 29(4):417–436, 1981.

[Lenting and Braspenning, 1994] Jacques Lenting and Peter Braspenning. An all-pay auction approach to reallocation. In *Proceedings of the 11th European Conference on Artificial Intelligence*, pages 259–263, August 1994.

[Lesser and Corkill, 1981] Victor R. Lesser and Daniel D. Corkill. Functionally accurate, cooperative distributed systems. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-11(1):81–96, November 1981.

[Lesser and Corkill, 1983] Victor R. Lesser and Daniel D. Corkill. The Distributed Vehicle Monitoring Testbed: A tool for investigating distributed problem solving networks. *AI Magazine*, 4(3):15–33, Fall 1983. (Also published in *Blackboard Systems*, Robert S. Engelmore and Anthony Morgan, editors, pages 353–386, Addison-Wesley, 1988 and in *Readings from AI Magazine: Volumes 1–5*, Robert Engelmore, editor, pages 69–85, AAAI, Menlo Park, California, 1988).

[Lesser and Erman, 1977] Victor R. Lesser and Lee D. Erman. A retrospective view of the Hearsay-II architecture. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 790–800, Cambridge, MA, August 1977.

[Lesser and Erman, 1980] Victor R. Lesser and Lee D. Erman. Distributed interpretation: A model and experiment. *IEEE Transactions on Computers*, C-29(12):1144–1163, December 1980. (Also published in *Readings in Distributed Artificial Intelligence*, Alan H. Bond and Les Gasser, editors, pages 120–139, Morgan Kaufmann, 1988.).

[Lesser *et al.*, 1988] Victor R. Lesser, Jasmina Pavlin, and Edmund H. Durfee. Approximate processing in real-time problem solving. *AI Magazine*, 9(1):49–61, Spring 1988.

[Lesser, 1991] Victor R Lesser. A retrospective view of FA/C distributed problem solving. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6):1347–1362, 1991.

[Lin and Kernighan, 1971] S Lin and B W Kernighan. An effective heuristic procedure for the traveling salesman problem. *Operations Research*, 21:498–516, 1971.

[Linnainmaa *et al.*, 1992] Seppo Linnainmaa, Olli Jokinen, Tuomas Sandholm, and Ari Vepsäläinen. Advanced computer supported vehicle routing for heavy transports. In *Finnish Artificial Intelligence Conference (STeP-92), New Directions in Artificial Intelligence*, volume 3, pages 163–172, Espoo, Finland, 1992.

[Low *et al.*, 1994a] Steven H Low, Nicholas F Maxemchuk, and Sanjoy Paul. Anonymous credit cards. 1994. Submitted.

[Low *et al.*, 1994b] Steven H Low, Nicholas F Maxemchuk, and Sanjoy Paul. Collusion in a multi-party communication protocol for anonymous credit cards. *IEEE/ACM Transactions on Networking*, 1994. Submitted.

[Luce and Raiffa, 1957] R. Duncan Luce and Howard Raiffa. *Games and Decisions*. John Wiley and Sons, New York, 1957. Dover republication 1989.

[Lundgren *et al.*, 1992] M G Lundgren, K Jörnsten, and P Värbrand. On the nucleolus of the basic vehicle routing game. Technical Report 1992-26, Linköping Univ., Dept. of Mathematics, Sweden, 1992.

[MacKie-Mason and Varian, 1993] Jeffrey K. MacKie-Mason and Hal R. Varian. Pricing the internet. In *Proceedings of the Public Access to the Internet Conference*. JFK School of Government, May 1993.

[MacKie-Mason and Varian, 1994] Jeffrey K. MacKie-Mason and Hal R. Varian. Some FAQs about usage-based pricing. *ftp: //alfred.sims.berkeley.edu /pub /Papers /useFAQs.html*, 1994.

[Malone *et al.*, 1983] T. W. Malone, R. E. Fikes, and M. T. Howard. Enterprise: A market-like task scheduler for distributed computing environments. Working Paper CISR WP 111 (Sloan WP 1537-84), Center for Information Systems Research, Massachusetts Institute of Technology, Cambridge, MA, 1983.

[Malone *et al.*, 1988] T. W. Malone, R. E. Fikes, K. R. Grant, and M. T. Howard. Enterprise: A market-like task scheduler for distributed computing environments. In B. A. Huberman, editor, *The Ecology of Computation*. North-Holland, 1988.

[Maskin, 1985] Eric S Maskin. The theory of implementation in Nash equilibrium: a survey. In Leonid Hurwicz, David Schmeidler, and Hugo Sonnenschein, editors, *Social goals and social organization: Essays in memory of Elisha Pazner*, chapter 6, pages 173–204. Cambridge University Press, 1985.

[McAfee and McMillan, 1996] R Preston McAfee and John McMillan. Analyzing the airwaves auction. *Journal of Economic Perspectives*, 10(1):159–175, 1996.

[Milgrom, 1985] Paul R Milgrom. The economics of competitive bidding: a selective survey. In Leonid Hurwicz, David Schmeidler, and Hugo Sonnenschein, editors, *Social goals and social organization: Essays in memory of Elisha Pazner*, chapter 9, pages 261–292. Cambridge University Press, 1985.

[Miller and Drexler, 1988a] Mark S. Miller and K. Eric Drexler. Comparative ecology: A computational perspective. In B. A. Huberman, editor, *The Ecology of Computation*. North-Holland, 1988.

[Miller and Drexler, 1988b] Mark S. Miller and K. Eric Drexler. Markets and computation: Agoric open systems. In B. A. Huberman, editor, *The Ecology of Computation*. North-Holland, 1988.

[Moehlman *et al.*, 1992] Theresa A Moehlman, Victor R Lesser, and Brandon L Buteau. Decentralized negotiation: An approach to the distributed planning problem. *Group Decision and Negotiation*, 2:161–191, 1992.

[Mor and Rosenschein, 1995] Yishay Mor and Jeffrey S. Rosenschein. Time and the prisoner's dilemma. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 276–282, San Francisco, CA, 1995.

[Morgenstern, 1990] Leora Morgenstern. A formal theory of multiple agent nonmonotonic reasoning. In *Proceedings of the National Conference on Artificial Intelligence*, pages 538–544, Boston, MA, 1990.

[Moses and Tennenholtz, 1993] Yoram Moses and Moshe Tennenholtz. Off-line reasoning for on-line efficiency. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 490–495, Chambery, France, 1993.

[Mullen and Wellman, 1995] Tracy Mullen and Michael P Wellman. A simple computational market for network information services. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 283–289, San Francisco, CA, June 1995.

[Muller and Satterwaithe, 1985] Eithan Muller and Mark A Satterwaithe. Strategy-proofness: the existence of dominant strategy mechanisms. In Leonid Hurwicz, David Schmeidler, and Hugo Sonnenschein, editors, *Social goals and social organization: Essays in memory of Elisha Pazner*, chapter 5, pages 131–172. Cambridge University Press, 1985.

[Myerson, 1985] Roger B Myerson. Bayesian equilibrium and incentive compatibility: an introduction. In Leonid Hurwicz, David Schmeidler, and Hugo Sonnenschein, editors, *Social goals and social organization: Essays in memory of Elisha Pazner*, chapter 8, pages 229–260. Cambridge University Press, 1985.

[Nash, 1950a] John Nash. The bargaining problem. *Econometrica*, 18:155–162, 1950.

[Nash, 1950b] John Nash. Equilibrium points in n-person games. *Proc. of the National Academy of Sciences*, 36:48–49, 1950.

[Neiman *et al.*, 1994] Daniel Neiman, David Hildum, Victor R Lesser, and Tuomas W Sandholm. Exploiting meta-level information in a distributed scheduling system. In *Proc. Twelfth National Conference on Artificial Intelligence (AAAI-94)*, August 1994.

[Neyman, 1985] A Neyman. Bounded complexity justifies cooperation in finitely repeated prisoner's dilemma. *Economic Letters*, pages 227–229, 1985.

[Office of Technology Assesment, 1994] Office of Technology Assesment. Electronic enterprises: Looking to the future, 1994.

[Ordeshook, 1986] Peter C Ordeshook. *Game theory and political theory.* Cambridge University Press, 1986.

[Osborne and Rubinstein, 1990] Martin J Osborne and Ariel Rubinstein. *Bargaining and Markets.* Academic Press, Inc., 1990.

[Osborne and Rubinstein, 1994] Martin J Osborne and Ariel Rubinstein. *A Course in Game Theory.* MIT Press, 1994.

[Papadimitriou and Yannakakis, 1991] C Papadimitriou and M Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991.

[Papadimitriou and Yannakakis, 1994] Christos H. Papadimitriou and Mihalis Yannakakis. On complexity as bounded rationality. In *STOC-94*, pages 726–733, 1994.

[Posner, 1977] Richard A Posner. *Economic Analysis of Law.* Little, Brown and Company, 2nd edition, 1977.

[Postlewaite, 1985] Andrew Postlewaite. Implementation via Nash equilibria in economic environments. In Leonid Hurwicz, David Schmeidler, and Hugo Sonnenschein, editors, *Social goals and social organization: Essays in memory of Elisha Pazner*, chapter 7, pages 205–228. Cambridge University Press, 1985.

[Raiffa, 1982] H. Raiffa. *The Art and Science of Negotiation.* Harvard Univ. Press, Cambridge, Mass., 1982.

[Rao and Georgeff, 1991] Anand Rao and Michael Georgeff. Asymmetry thesis and side-effect problems in linear-time and branching-time intention logics. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pages 498–504, Sydney, Australia, 1991.

[Rao and Georgeff, 1995] Anand Rao and Michael Georgeff. BDI agents: From theory to practise. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 312–319, San Francisco, CA, 1995.

[Rasmusen, 1989] Eric Rasmusen. *Games and Information.* Basil Blackwell, 1989.

[Rosenschein and Genesereth, 1985] Jeffrey S. Rosenschein and Michael R. Genesereth. Deals among rational agents. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 91–99, Los Angeles, CA, August 1985. (Also published in *Readings in Distributed Artificial Intelligence*, Alan H. Bond and Les Gasser, editors, pages 227–234, Morgan Kaufmann, 1988.).

[Rosenschein and Zlotkin, 1994] Jeffrey S Rosenschein and Gilad Zlotkin. *Rules of Encounter.* MIT Press, 1994.

[Rothkopf and Harstad, 1995] Michael H Rothkopf and Ronald M Harstad. Two models of bid-taker cheating in Vickrey auctions. *Journal of Business*, 68(2):257–267, 1995.

[Rothkopf *et al.*, 1990] Michael H Rothkopf, Thomas J Teisberg, and Edward P Kahn. Why are Vickrey auctions rare? *Journal of Political Economy*, 98(1):94–109, 1990.

[Rubinstein, 1986] Ariel Rubinstein. Finite automata play the repeated prisoner's dilemma. *Journal of Economic Theory*, 39:83–96, 1986.

[Russell and Wefald, 1991a] Stuart Russell and Eric Wefald. *Do the right thing: Studies in Limited Rationality.* The MIT Press, 1991.

[Russell and Wefald, 1991b] Stuart Russell and Eric Wefald. Principles of metareasoning. *Artificial Intelligence*, 49:361–395, 1991.

[Sandholm and Lesser, 1994] Tuomas W Sandholm and Victor R Lesser. Utility-based termination of anytime algorithms. In *ECAI Workshop on Decision Theory for DAI Applications*, pages 88–99, Amsterdam, The Netherlands, 1994. Extended version: Univ. of Mass. at Amherst, Comp. Sci. Tech. Report 94-54.

[Sandholm and Lesser, 1995a] Tuomas W Sandholm and Victor R Lesser. Coalition formation among bounded rational agents. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 662–669, Montreal, Canada, August 1995. Extended version appeared as University of Massachusetts at Amherst, Computer Science Department technical report 95-71.

[Sandholm and Lesser, 1995b] Tuomas W Sandholm and Victor R Lesser. Equilibrium analysis of the possibilities of unenforced exchange in multiagent systems. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 694–701, Montreal, Canada, August 1995.

[Sandholm and Lesser, 1995c] Tuomas W Sandholm and Victor R Lesser. Issues in automated negotiation and electronic commerce: Extending the contract net framework. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 328–335, San Francisco, CA, June 1995.

[Sandholm and Lesser, 1996] Tuomas W Sandholm and Victor R Lesser. Advantages of a leveled commitment contracting protocol. In *Proceedings of the National Conference on Artificial Intelligence*, pages 126–133, Portland, OR, August 1996. Extended version appeared as University of Massachusetts at Amherst, Computer Science Department technical report 95-72.

[Sandholm and Lesser, 1997] Tuomas W Sandholm and Victor R Lesser. Coalitions among computationally bounded agents. *Artificial Intelligence*, Special issue on Principles of Multiagent Systems, 1997. To appear.

[Sandholm, 1991] Tuomas W Sandholm. A strategy for decreasing the total transportation costs among area-distributed transportation centers. In *Nordic Operations Analysis in Cooperation (NOAS-91): OR in Business*, Turku School of Economics, Finland, 1991.

[Sandholm, 1992a] Tuomas W Sandholm. Automatic cooperation of area-distributed dispatch centers in vehicle routing. In *International Conference on Artificial Intelligence Applications in Transportation Engineering*, pages 449–467, San Buenaventura, California, 1992.

[Sandholm, 1992b] Tuomas W Sandholm. A bargaining network for intelligent agents. In *Finnish Artificial Intelligence Conference (STeP-92), New Directions in Artificial Intelligence*, volume 3, pages 173–181, Espoo, Finland, 1992.

[Sandholm, 1993] Tuomas W Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Proc. 11th National Conference on Artificial Intelligence (AAAI-93)*, pages 256–262, July 1993.

[Sandholm, 1996] Tuomas W Sandholm. Limitations of the Vickrey auction in computational multiagent systems. In *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-96)*, Keihanna Plaza, Kyoto, Japan, December 1996.

[Sathi and Fox, 1989] A Sathi and Mark Fox. Constraint-directed negotiation of resource reallocations. In Michael N. Huhns and Les Gasser, editors, *Distributed Artificial Intelligence*, volume 2 of *Research Notes in Artificial Intelligence*, chapter 8, pages 163–193. Pitman, 1989.

[Satterthwaite, 1975] M A Satterthwaite. Strategy-proofness and Arrow's conditions: existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, 1975.

[Schweers Cook and Levi, 1990] Karen Schweers Cook and Margaret Levi. *The Limits of Rationality*. University of Chicago Press, 1990.

[Selten, 1965] R Selten. Spieltheoretische behandlung eines oligopolmodells mit nachfrageträgheit. *Zeitschrift für die gesamte Staatswissenschaft*, 12:301–324, 1965.

[Sen, 1994] Sandip Sen. The role of commitment in cooperative negotiation. *International Journal on Intelligent Cooperative Information Systems*, 3(1):67–81, 1994.

[Sen, 1996] Sandip Sen. A formal study of distributed meeting scheduling. *Group Decision and Negotiation Support Systems*, 1996. To appear.

[Shapley, 1967] Lloyd S Shapley. On balanced sets and cores. *Naval Research Logistics Quarterly*, 14:453–460, 1967.

[Shechory and Kraus, 1995a] Onn Shechory and Sarit Kraus. Feasible formation of stable coalitions among autonomous agents in general environments. *Computational Intelligence Journal*, 1995. Submitted.

[Shechory and Kraus, 1995b] Onn Shechory and Sarit Kraus. Task allocation via coalition formation among autonomous agents. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 655–661, Montreal, Canada, August 1995.

[Shoham and Tennenholtz, 1993] Yoav Shoham and Moshe Tennenholtz. On social laws for artificial agent societies: Off-line design. *Artificial Intelligence*, 1993.

[Simon, 1982] Herbert A Simon. *Models of bounded rationality*, volume 2. MIT Press, 1982.

[Smart Market, 1996] Smart Market. Web page mention of the use of Vickrey auction in Smart Market. *http://mortadello.wu-wien.ac.at /~iwork /pricing /smart_market.html*, 1996.

[Smith and Davis, 1981] Reid G. Smith and Randall Davis. Frameworks for cooperation in distributed problem solving. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11(1):61–70, January 1981. (Also published in *Readings in Distributed Artificial Intelligence*, Alan H. Bond and Les Gasser, editors, pages 61–70, Morgan Kaufmann, 1988.).

[Smith, 1978] Reid Garfield Smith. *A Framework for Problem Solving in a Distributed Processing Environment*. PhD thesis, Stanford University, December 1978. A revised version was published by UMI Research Press.

[Smith, 1979] Reid G. Smith. A framework for distributed problem solving. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pages 836–841, Tiblisi, Georgia, USSR, August 1979.

[Smith, 1980] Reid G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, December 1980.

[Sun Microsystems, 1996] Sun Microsystems. Web page mention of the use of Vickrey auction in Webmart. *http://www.sun.com:80 /960201 /cover /webmart.html*, 1996.

[Sycara *et al.*, 1991] Katia Sycara, S Roth, Norman Sadeh, and Mark Fox. Distributed constrained heuristic search. *IEEE Transactions on Systems, Man, and Cybernetics*, Special issue on distributed artificial intelligence SMC-21(6):1446–1461, Nov/Dec 1991.

[Telser, 1980] L G Telser. A theory of self-enforcing agreements. *Journal of Business*, 53(1):27–44, 1980.

[Tennenholtz and Moses, 1989] Moshe Tennenholtz and Yoram Moses. On cooperation in a multi-entity model (preliminary report). In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 918–923, Detroit, MI, 1989.

[Turner and Eaton, 1994] Roy M Turner and Peggy S. Eaton. Handling unanticipated events during collaboration. In *Proceedings of the Cognitive Science Society*, Atlanta, Georgia, 1994.

[Turner, 1992] Roy M Turner. The tragedy of the commons and distributed AI systems. In *Proceedings of the 12th International Workshop on Distributed Artificial Intelligence*, pages 379–390, May 1992.

[van der Linden and Verbeek, 1985] Wim J van der Linden and Albert Verbeek. Coalition formation: A game-theoretic approach. In Henk A M Wilke, editor, *Coalition Formation*, volume 24 of *Advances in Psychology*. North Holland, 1985.

[Van Dyke Parunak, 1987] H Van Dyke Parunak. Manufacturing experience with the contract net. In Michael N. Huhns, editor, *Distributed Artificial Intelligence*, Research Notes in Artificial Intelligence, chapter 10, pages 285–310. Pitman, 1987.

[Varian, 1992] Hal R Varian. *Microeconomic analysis*. New York: W. W. Norton, 1992.

[Vickrey, 1961] W Vickrey. Counter speculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

[Vidal and Durfee, 1995] José M. Vidal and Edmund H. Durfee. Recursive agent modeling using limited rationality. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 376–383, San Francisco, CA, 1995.

[von Neumann and Morgenstein, 1947] J von Neumann and O Morgenstein. *Theory of games and economic behavior*. Princeton University Press, 1947.

[Waldspurger *et al.*, 1992] Carl A Waldspurger, Tad Hogg, Bernardo Huberman, Jeffrey O Kephart, and W Scott Stornetta. Spawn: A distributed computational economy. *IEEE Transactions on Software Engineering*, 18(2):103–117, 1992.

[Waters, 1987] C D Waters. A solution procedure for the vehicle-scheduling problem based on iterative route improvement. *Journal of the Operational Research Society*, 38(9):833–839, 1987.

[Wellman, 1992] Michael Wellman. A general- equilibrium approach to distributed transportation planning. In *Proc. 10th National Conference on Artificial Intelligence (AAAI-92)*, pages 282–289, San Jose, CA, July 1992.

[Wellman, 1994] Michael Wellman. A computational market model for distributed configuration design. In *Proc. 12th National Conference on Artificial Intelligence (AAAI-94)*, pages 401–407, Seattle, WA, July 1994.

[Zilberstein and Russell, 1991] Shlomo Zilberstein and Stuart Russell. Composing real-time systems. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pages 212–217, Sydney, Australia, 1991.

[Zilberstein and Russell, 1995] Shlomo Zilberstein and Stuart Russell. Optimal composition of real-time systems. *Artificial Intelligence*, December 1995.

[Zilberstein, 1993] Shlomo Zilberstein. *Operational rationality through compilation of anytime algorithms*. PhD thesis, University of California, Berkeley, 1993.

[Zlotkin and Rosenschein, 1993a] Gilad Zlotkin and Jeffrey S Rosenschein. A domain theory for task oriented negotiation. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 416–422, Chambery, France, August 1993.

[Zlotkin and Rosenschein, 1993b] Gilad Zlotkin and Jeffrey S Rosenschein. The extent of cooperation in state-oriented domains. *Computers and Artificial Intelligence*, 12(2):105–122, 1993.

[Zlotkin and Rosenschein, 1993c] Gilad Zlotkin and Jeffrey S Rosenschein. Negotiation with incomplete information about worth: Strict versus tolerant mechanisms. In *Proceedings of the First International Conference on Intelligent and Cooperative Information Systems*, pages 175–184, Rotterdam, The Netherlands, May 1993.

[Zlotkin and Rosenschein, 1994] Gilad Zlotkin and Jeffrey S Rosenschein. Coalition, cryptography and stability: Mechanisms for coalition formation in task oriented domains. In *Proceedings of the National Conference on Artificial Intelligence*, pages 432–437, Seattle, WA, July 1994.