The Pickup and Delivery Problem: a Many-objective Analysis

Abel García-Nájera and Antonio López-Jaimes

Universidad Autónoma Metropolitana, Unidad Cuajimalpa,
Departamento de Matemáticas Aplicadas y Sistemas, México, D.F.,
Mexico

{agarcian,alopez}@correo.cua.uam.mx

Abstract. The pickup and delivery problem (PDP) considers a set of transportation requests, which specify the quantity of product that has to be picked up from an origin and delivered to a destination. There exist a number of vehicles available to be used for completing these tasks. PDP consists of finding a collection of routes with minimum cost, such that all transportation request are serviced. Traditionally, cost has been associated with the number of routes and the total travel distance. However, in many applications, some other objectives emerge, for example, the minimization of travel time and the maximization of the collected profit. If we consider all these four objectives equally important, PDP can be tackled as a many-objective problem. In this paper we are interested in analyzing this many-objective problem in order to study some of its properties, specifically, (*i*) the change of difficulty when the number of objectives is increased, and (*ii*) the conflict degree between each pair of objectives. In order to analyze these topics, we compare the performance of a recently proposed multi-objective evolutionary algorithm against that of the well-known ϵ-MOEA, which has shown good results in many-objective problems.

Keywords: Many-objective optimization, pickup and delivery problem, evolutionary algorithm

1 Introduction

The pickup and delivery problem (PDP) is part of the class of problems known as vehicle routing problem (VRP) [7], which is well-known to be NP-hard [9]. The PDP considers transportation requests, which are defined between pairs of customers (origin and destination). The problem consists in designing a set of routes with minimum cost to service all transportation requests.

Cost is regularly associated with the number of routes and the travel distance, however, there are several other sources of cost [6]. One generalization of the PDP is the PDP with time windows and selective requests (PDPTWSR), which considers two additional sources of cost, namely the travel time and the uncollected profit. Moreover, if those four objectives are considered to be equally important, PDPTWSR can be tackled as a many-objective problem.

The PDPTWSR is defined as follows. There is a set $\mathcal{V} = \{0, 1, \ldots, N, N + 1, \ldots, 2N\}$ of $2N + 1$ vertices. Customers are represented by the vertices in the subset $\mathcal{V}' = \mathcal{V} \setminus \{0\}$. Each customer $i \in \mathcal{V}'$ is geographically located at coordinates (x_i, y_i), and has an associated time window $[b_i, e_i]$, during which it has to be serviced, and a service time s_i required to load or unload goods. Customer subset $\mathcal{V}_{\mathcal{O}} = \{1, \ldots, N\}$ corresponds to the pickup locations, while customer subset $\mathcal{V}_{\mathcal{D}} = \{N + 1, \ldots, 2N\}$ represents the delivery locations.

The set $\mathcal{TR} = \{1, \ldots, N\}$ represents N transportation requests. Each transportation request $i \in \mathcal{TR}$ specifies the size q_i of the load to be transported and the locations $j \in \mathcal{V}_{\mathcal{O}}$ and $k \in \mathcal{V}_{\mathcal{D}}$ where the load will be collected (origin) and delivered (destination), respectively. Finally, each transportation request $i \in \mathcal{TR}$ has an associated profit p_i, hence $P = \sum_{i \in \mathcal{TR}} p_i$ is the total possible profit.

The vertex 0 is located at (x_0, y_0) and has a time window $[0, e_0 \geq \max \{e_i : i \in \mathcal{V}'\}]$. This vertex is the base of a homogeneous fleet of vehicles which have capacity Q, greater or equal to the maximum size of the loads to be transported.

The travel between vertices i and j has associated costs, such as the distance d_{ij} (relating to fuel cost) and travel time t_{ij} (relating to driver salary). Transportation requests are optional to be serviced, this means that origin and destination customers associated to a request might not be visited. Therefore, if they are not visited, there is no profit collected from that request. For the benchmark problems to be considered later, unit velocity and direct travel are assumed, so the times and distances are both simply taken to be the Euclidean distances. Moreover, the profit p_i associated to transportation request $i \in \mathcal{TR}$ will be equaled to the size q_i of the transportation load. For real-world problems, however, the distances d_{ij} are unlikely to be Euclidean, the travel times t_{ij} are unlikely to be simply related to the distances, and profit does not necessarily involves only the load to be transported.

The aim of the problem is to find the set of a minimum number routes which minimize the total cost, the travel distance and the uncollected profit, such that: (i) each route starts and ends at the base, (ii) customers related to each transportation request are visited by only one vehicle or none, (iii) the vehicle load must not, at any time, be negative and must not exceed the vehicle capacity Q, (iv) for each request i, its corresponding pickup location p_i must be visited in the same route and before its corresponding delivery location d_i, and (v) for each request i, its corresponding delivery location d_i must be visited in the same route and after its corresponding pickup location p_i.

Having defined the PDPTWSR, we can define four key objectives this paper will concentrate on minimizing, namely the number of routes or vehicles (f_{R}), the total travel distance (f_{D}), the total travel time (f_{T}), and the uncollected profit (f_{P}), subject to the constraints explained above.

To the best of our knowledge, the problem under study has not been tackled before. However, the pickup and delivery problem with time windows (PDPTW) has been subject of plenty of investigation. Many approaches for solving the

PDPTW can be found in the literature and Parragh et al. [11] make an excellent survey of many of them. We will review some previous studies which have tested their approaches on the PDPTW benchmark sets of Li and Lim [10].

Li and Lim [10] introduced a metaheuristic based on a tabu-embedded simulated annealing algorithm, which restarts a search procedure from the current best solution after several non-improving search iterations. This restart strategy guides the local search in three neighborhoods defined to solve the general multiple-vehicle PDPTW. This is combined with a metaheuristic based on a K restarts annealing procedure with tabu-list to avoid cycling in the search process. In addition, authors generated several benchmark instances which are used in this study. Bent and Van Hentenryck [1] proposed a two-stage hybrid algorithm. The first stage uses a simple simulated annealing algorithm to decrease the number of routes, while the second stage uses a large neighborhood search to decrease the total travel distance. Ropke and Pisinger [12] presented an adaptive large neighborhood search, which is and extension to the large neighborhood search and the ruin-and-recreate heuristic. The proposed method is composed of a number of competing subheuristics that are used with a frequency corresponding to their historic performance. Hasle and Kloster [5] introduced SPIDER, which is a heuristic approach based on local search. This approach has three phases: construction of initial solutions, tour depletion, and iterative improvement. The construction phase considers extensions of classical constructions heuristics as well as other methods proposed by the authors. In the tour depletion phase, a greedy tour removal heuristic is invoked. A single tour is depleted, and insertion of the unassigned orders in the remaining tours is attempted. The new solution is accepted if all unassigned orders are successfully inserted in the remaining tours. Finally, the iterative improvement phase is based on variable neighborhood descent, using a selection of several intra-tour, inter-tour, and special operators.

In the above reviewed studies, the PDP has been solved considering the minimization of the number of routes first, and then, the total travel distance. More recently, García-Nájera and Gutiérrez-Andrade [3] proposed a multi-objective evolutionary approach for solving the PDPTW, minimizing the number of routes, the travel distance, and the travel time simultaneously. Their approach was able to find many best-known solutions to benchmark instances and outperformed a well-known multi-objective optimizer.

As far a we are concerned, the study of García-Nájera and Gutiérrez-Andrade [3] is the only regarding the solution of the PDPTW considering multiple objectives. The present study aims at remedy this situation, that is, the aim is to analyze the many-objective problem in the sense of how difficult the problem is when more objectives are considered.

The remainder of this paper is organized as follows. Section 2 introduces the main concepts of multi-objective optimization and explains the performance metrics that are used here to compare algorithms. The multi-objective optimizers used in this study are briefly described in Section 3. Then, Section 4 presents the analysis of the results from both algorithms. Finally, we present our conclusions in Section 5.

2 Multi-objective Optimization Problems

Any multi-objective optimization problem can be defined, without loss of generality, as the problem of minimizing $\mathbf{f}(\mathbf{x})$, subject to $g_i(\mathbf{x}) \leq 0$, $\forall\ i \in \{1, \ldots, p\}$, and $h_j(\mathbf{x}) = 0$, $\forall\ j \in \{1, \ldots, q\}$, where $\mathbf{x} \in \mathcal{X}$ is a potential solution to the problem, $\mathcal{X}$ is the domain of solutions, $\mathbf{f} : \mathcal{X} \to \mathbb{R}^m$ are the m objective functions, and the constraint functions $g_i, h_j : \mathcal{X} \to \mathbb{R}$ delimit the feasible search space.

We say that solution $\mathbf{x}$ *dominates* solution $\mathbf{y}$, written as $\mathbf{x} \prec \mathbf{y}$, if and only if $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$, $\forall\ i \in \{1, \ldots, m\}$, and $\exists\ j : f_j(\mathbf{x}) < f_j(\mathbf{y})$. Consequently, solution $\mathbf{x} \in \mathcal{S} \subseteq \mathcal{X}$ is *non-dominated* with respect to $\mathcal{S}$ if there is no solution $\mathbf{y} \in \mathcal{S}$ such that $\mathbf{y} \prec \mathbf{x}$. Solution $\mathbf{x} \in \mathcal{X}$ is said to be *Pareto optimal* if it is non-dominated with respect to $\mathcal{X}$, and the *Pareto optimal set* is defined as $\mathcal{P}_s = \{\mathbf{x} \in \mathcal{X} \mid \mathbf{x} \text{ is Pareto optimal}\}$. Finally, the *Pareto front* is defined as $\mathcal{P}_f = \{\mathbf{f}(\mathbf{x}) \in \mathbb{R}^m \mid \mathbf{x} \in \mathcal{P}_s\}$.

In contrast, with single-objective problems, where one can straightforwardly compare the best solutions from the various approaches studied, multi-objective problems have to compare whole sets of solutions. Many performance indicators have been proposed in the literature, being two of them the *hypervolume* and the *generational distance*, which are explained next.

2.1 Hypervolume Indicator

The hypervolume performance metric $\mathsf{H}(\mathcal{A}, \mathbf{z})$ concerns the size of the objective space defined by the approximation set $\mathcal{A}$, which is limited by setting a suitable reference point $\mathbf{z}$. Formally, for a two-dimensional objective space $\mathbf{f}(\mathbf{x}) = \big(f_1(\mathbf{x}), f_2(\mathbf{x})\big)$, each solution $\mathbf{x_i} \in \mathcal{A}$ delimits a rectangle defined by $\big(f_1(\mathbf{x_i}), f_2(\mathbf{x_i})\big)$ and the reference point $\mathbf{z} = (z_1, z_2)$, and the size of the union of all such rectangles is used as the measure. This concept can be extended to any number of dimensions m to give the general hypervolume metric [13].

2.2 Generational Distance Indicator

In order to evaluate the convergence of the algorithms to the Pareto front we adopted the generational distance indicator GD, which is defined by $\mathsf{GD}(\mathcal{A}) = 1/|\mathcal{A}|\big(\sum_{i=1}^{|\mathcal{A}|} d_i^2\big)^{1/2}$, where d_i is the Euclidean distance between each solution $\mathbf{x_i} \in \mathcal{A}$ and the nearest member of $\mathcal{P}_f$.

3 Many-objective Optimization of the PDPTWSR

In this section, the two multi-objective optimizers used in this study for solving the many-objective PDPTWSR are briefly described. The first is the well-known ϵ-MOEA [2], which has been proved to be successful in a number of applications. The second is the multi-objective evolutionary algorithm recently proposed by García-Nájera and Gutiérrez-Andrade [3], hereinafter GN-MOEA, which is able to find appropriate Pareto approximations to the related problem PDPTW.

3.1 ϵ-MOEA

Deb et al. [2] proposed ϵ-MOEA, based on the ϵ-dominance concept introduced in [8] that states: Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$. Then, $\mathbf{x}$ is said to ϵ-dominate $\mathbf{y}$ for some $\epsilon > 0$, denoted as $\mathbf{x} \succ_\epsilon \mathbf{y}$, if and only if $(1 + \epsilon) \cdot x_i \geq y_i, \forall\, i \in \{1, \ldots, m\}$.

In ϵ-MOEA, the search space is divided into a number of hyper-boxes and diversity is maintained by ensuring that a hyper-box can be occupied by only one solution. ϵ-MOEA randomly initializes a population. The non-dominated solutions of are copied to an archive population. Two solutions are chosen as parents: one is chosen from the population and one is chosen from the archive population. One offspring solution is created using these parents. If the offspring dominates one or more population members, then the offspring replaces one of them (chosen at random). On the other hand, if any population member dominates the offspring, it is not accepted. When both the above tests fail (that is, the offspring is non-dominated to the population members), the offspring replaces a randomly chosen population member, thereby ensuring that the EA population size remains unchanged. For the offspring to be included in the archive population, the offspring is compared with each member of the archive using ϵ-dominance criterion.

3.2 GN-MOEA

García-Nájera and Gutiérrez-Andrade [3] proposed a problem-specific multi-objective evolutionary algorithm (GN-MOEA) for minimizing three objectives of the PDPTW simultaneously, namely the number of routes, the travel distance, and the travel time. GN-MOEA uses an encoding of list of lists: a route is encoded as a list and a solution as a list of routes. They use the non-dominance sorting criterion [4] to assign fitness to individuals. Solution similarity is used as a diversity measure. This is simply computed as the ratio of the number of arcs that are common in two solutions to the total number of arcs traversed in both solutions. This measure is used in the mating selection process, since one parent is selected according to the diversity measure and the other is selected according the fitness. These parents are selected by using the tournament selection method. Crossover aims at combining routes from both parents, while mutation exchanges transportation requests between routes, and removes transportation requests from one route and inserts them into another.

4 Experimental Study

Our study has two main purposes. Firstly, to determine which of the two multi-objective optimizers described in the previous Section has a better performance on the problem at hand, and secondly, to perform a many-objective analysis of the PDPTWSR. To this end, we carried out two sets of experiments. The first set (RDT) considered the objectives number of routes (f_R), travel distance (f_D), and travel time (f_T), and the minimization of four combinations of these objetives, namely $f_{\mathrm{R}}f_{\mathrm{D}}$, $f_{\mathrm{R}}f_{\mathrm{T}}$, $f_{\mathrm{D}}f_{\mathrm{T}}$, and $f_{\mathrm{R}}f_{\mathrm{D}}f_{\mathrm{T}}$. The second set of experiments (RDTP), took into account the three previous objectives plus the uncollected profit (f_{P}), and the minimization of five combinations of these four objectives, namely $f_{\mathrm{R}}f_{\mathrm{P}}$, $f_{\mathrm{R}}f_{\mathrm{D}}f_{\mathrm{P}}$, $f_{\mathrm{R}}f_{\mathrm{T}}f_{\mathrm{P}}$, $f_{\mathrm{D}}f_{\mathrm{T}}f_{\mathrm{P}}$, and $f_{\mathrm{R}}f_{\mathrm{D}}f_{\mathrm{T}}f_{\mathrm{P}}$.

In order to have controlled experiments, we used the PDPTW benchmark sets of Li and Lim [10], which include 56 instances of size $N = 100$, and are divided into six categories: lc1, lc2, lr1, lr2, lrc1, and lrc2. Crossover and mutation operators used in GN-MOEA were set also in ϵ-MOEA for fair comparison. We ran both algorithms 30 times for each problem instance. The GN-MOEA parameters were set to the values reported in [3]: population size = 100, number of generations = 500, tournament size = 5, and mutation probability = 0.1. These values were also set in ϵ-MOEA, plus the number of hyper-boxes = 100.

4.1 Analysis of the Hypervolume Indicator

To compute the hypervolume indicator, we require an appropriate reference point. From the 60 Pareto approximations (30 from GN-MOEA and 30 from ϵ-MOEA) for each instance, we took the maximum value for each objective, and the reference point $\mathbf{z}$ was set 10% above each dimension's maximum value.

For each instance and repetition, we took the non-dominated set and computed the hypervolume covered by those solutions. Then, we applied a statistical t-test (two-sample, one-tailed, unequal variance) to the two vectors of 30 hypervolume values, from the GN-MOEA and ϵ-MOEA, respectively, to test the null hypothesis that data in the vectors are independent random samples from normal distributions with equal means. The summary of the t-test results are shown in Tables 1 and 2, for the sets of experiments RDT and RDTP, respectively. The first main column of these Tables show the instance category and the number of instances comprising that category. These Tables have one main column for each combination of objectives, and each main column has two subcolumns, corresponding to the number of instances in each instance category for which non-dominated solutions from GN-MOEA (GN) and from ϵ-MOEA (ϵ), respectively, covered a statistically larger hypervolume.

From Table 1 we can observe that, in the $f_{\mathrm{R}}f_{\mathrm{D}}$ case, both algorithms performed similarly, since for only three out of 56 instances there was a statistical difference in the size of the covered hypervolume. For the remaining three combinations of objectives, it is clear that GN-MOEA found non-dominated solutions that covered a significantly larger hypervolume for more instances than ϵ-MOEA.

On the other hand, from Table 2 we can see that the non-dominated solutions from ϵ-MOEA covered a significantly larger hypervolume for many instances in the cases $f_{\mathrm{R}}f_{\mathrm{P}}$, $f_{\mathrm{D}}f_{\mathrm{T}}f_{\mathrm{P}}$, and $f_{\mathrm{R}}f_{\mathrm{D}}f_{\mathrm{T}}f_{\mathrm{P}}$, while the Pareto approximations from GN-MOEA delimited a significantly larger hypervolume in more instances for the cases $f_{\mathrm{R}}f_{\mathrm{D}}f_{\mathrm{P}}$ and $f_{\mathrm{R}}f_{\mathrm{T}}f_{\mathrm{P}}$.

Overall we can conjecture that GN-MOEA has a superior performance over ϵ-MOEA when objective $f_{\mathrm{R}}f_{\mathrm{D}}f_{\mathrm{T}}$ are considered, however, when objective f_{P} is included, ϵ-MOEA outperforms GN-MOEA in many cases.

Table 1. Number of instances for which non-dominated solutions from each optimizer cover a significantly larger hypervolume for the set of experiments RDT.

Inst.		$f_{\mathrm{R}}f_{\mathrm{D}}$		$f_{\mathrm{R}}f_{\mathrm{T}}$		$f_{\mathrm{D}}f_{\mathrm{T}}$		$f_{\mathrm{R}}f_{\mathrm{D}}f_{\mathrm{T}}$	
Cat.	No.	GN	ϵ	GN	ϵ	GN	ϵ	GN	ϵ
lc1	9	1	0	3	1	1	0	2	0
lc2	8	0	0	3	0	2	0	2	0
lr1	12	0	1	5	0	10	0	5	0
lr2	11	0	0	6	0	10	0	5	1
lrc1	8	0	1	1	0	4	0	3	0
lrc2	8	0	0	6	0	6	0	2	0
Total	56	1	2	24	1	33	0	19	1

Table 2. Number of instances for which non-dominated solutions from each optimizer cover a significantly larger hypervolume for the set of experiments RDTP.

Inst.		$f_{\mathrm{R}}f_{\mathrm{P}}$		$f_{\mathrm{R}}f_{\mathrm{D}}f_{\mathrm{P}}$		$f_{\mathrm{R}}f_{\mathrm{T}}f_{\mathrm{P}}$		$f_{\mathrm{D}}f_{\mathrm{T}}f_{\mathrm{P}}$		$f_{\mathrm{R}}f_{\mathrm{D}}f_{\mathrm{T}}f_{\mathrm{P}}$	
Cat.	No.	GN	ϵ	GN	ϵ	GN	ϵ	GN	ϵ	GN	ϵ
lc1	9	0	3	3	2	9	0	3	3	0	9
lc2	8	0	5	5	0	8	0	1	4	0	5
lr1	12	0	3	0	7	0	9	0	12	0	12
lr2	11	0	6	5	0	10	0	0	11	0	9
lrc1	8	1	5	2	3	0	5	0	7	0	8
lrc2	8	0	5	3	0	6	0	0	8	0	7
Total	56	1	27	18	12	33	14	4	45	0	50

4.2 Analysis of the Generational Distance Indicator

Since the optimal Pareto front is not known for the benchmark instances, for computing GD we used, as a reference set, the non-dominated solutions resulting from the union of all the approximation sets to $\mathcal{P}_f$ obtained by each algorithm at the end of every run. In order to analyze the difficulty of PDPTWSR when objectives are added we used the comparison of both algorithms to get useful insights about its difficulty. More specifically, if the difference between the performance of the optimizers vary with the number of objectives, that would indicate that PDPTWSR's difficulty vary with the number of objectives.

After computing GD for each approximation set obtained by both algorithms we carried out the Wilcoxon rank-sum test (two-sample, one-tailed) to determine which algorithm yielded the smaller generational distance. We tested the alternative hypothesis that the mean of GD generated by optimizer A is less than that of B. In this test we employed a significance level of 5%. In Table 3 we present a summary of the statistical tests showing the number of instances in which each MOEA achieved a significantly better GD value. For the set of experiments RDTP in Table 3 we can observe that GN-MOEA outperformed ϵ-MOEA in most of the instances. Interestingly, the result for $f_R f_P$ does not agree with that of the hypervolume. By analyzing some approximation sets we realized that in some instances, although GN-MOEA generated solutions very

Table 3. Number of instances for which non-dominated solutions from each optimizer has a significantly better GD for the set of experiments RDTP.

Inst.		$f_{\mathrm{R}}f_{\mathrm{P}}$		$f_{\mathrm{R}}f_{\mathrm{D}}f_{\mathrm{P}}$		$f_{\mathrm{R}}f_{\mathrm{T}}f_{\mathrm{P}}$		$f_{\mathrm{D}}f_{\mathrm{T}}f_{\mathrm{P}}$		$f_{\mathrm{R}}f_{\mathrm{D}}f_{\mathrm{T}}f_{\mathrm{P}}$	
Cat.	No.	GN	ϵ	GN	ϵ	GN	ϵ	GN	ϵ	GN	ϵ
lc1	9	9	0	9	0	9	0	0	9	0	9
lc2	8	7	0	8	0	8	0	0	2	0	5
lr1	12	12	0	3	3	4	0	0	12	0	12
lr2	11	11	0	9	0	11	0	0	9	0	9
lrc1	8	8	0	7	0	6	0	0	7	0	7
lrc2	8	8	0	8	0	8	0	0	7	0	7
Total	56	55	0	44	3	46	0	0	46	0	49

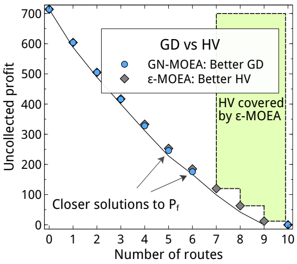

Fig. 1. Approximation sets where GN-MOEA achieves a good GD and ϵ-MOEA a good HV.

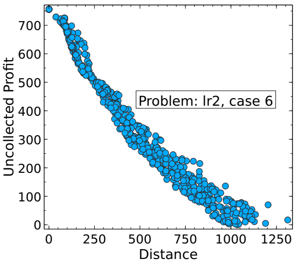

Fig. 2. Conflict between objectives distance and uncollected profit.

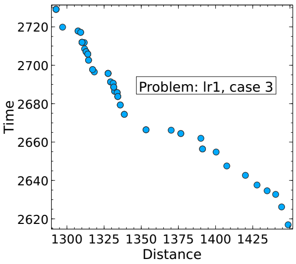

Fig. 3. Conflict between objectives distance and time.

near to the reference set, it did not cover some solutions in the middle of it. In order to illustrate this situation, we present Figure 1, in which solutions with 7, 8 and 9 routes were not found by GN-MOEA. In this approximation sets, GN-MOEA achieved a better GD (4.7 vs 5.15). However, with the same outputs, ϵ-MOEA overcame GN-MOEA in terms of hypervolume. The explanation is that the hypervolume contribution of those middle solutions was more important than the hypervolume advantage produced by solutions close to the reference set. Regarding the difficulty of PDPTWSR as the number of objective increases we have the following observations. In some comparative studies, ϵ-MOEA has shown a good performance in many-objective optimization problems. That is, its convergence ability degrades very slowly when more objectives are added. Therefore, we expect ϵ-MOEA to maintain a similar performance for every number of objectives. The results show that, regarding both hypervolume and generational distance, GN-MOEA outperformed ϵ-MOEA in most of the problem instances with 2 and 3 objectives. In contrast, ϵ-MOEA obtained better indicator values in $f_D f_T f_P$ and $f_R f_D f_T f_P$. We argue that GN-MOEA's convergence ability was affected by the additional objective. Therefore, this could indicate that the difficulty of PDPTWSR increases as more objectives are added.

4.3 Conflict among Objectives of PDPTWSR

One can expect a natural conflict between uncollected profit (f_P) and either time (f_T) or distance (f_D). However, at first sight, there is no conflict between distance and time. In order to estimate the conflict among all pair of objectives we employed the Spearman's rank correlation coefficient (which value is in $[-1, 1]$). That is, a coefficient value close to -1 means that one objective grows while the other decreases. For this purpose, we test the alternative hypothesis (significance value of 0.05) that the Spearman's rank coefficient is negative for a pair of objective values. The results show that, according to correlation, there is conflict between time (f_T) and distance (f_D) for many of the instances: 0, 3, 9, 11, 7, 8 for instance categories lc1, lc2, lr1, lr2, lrc1, lrc2, respectively. In the other hand, there is no conflict at all between f_R and either f_T or f_D, and there is conflict between f_P and f_R, f_D, f_T in all instances. In order to illustrate this situation we present Figures 2 and 3 in which the Pareto front of the best approximation of $\mathcal{P}_f$ is plotted.

5 Conclusions

In this paper we have analyzed some properties of the many-objective PDP, namely the change of difficulty when the number of objectives is increased, the conflict degree between each pair of objectives, and whether the difficulty of particular objectives decreases due to the change of the fitness landscape. To this end, we employed two multi-objective optimizers: the well-known ϵ-MOEA, which has been successful in a number of applications, and the recently proposed GN-MOEA, which showed improved performance over a popular optimizer on some benchmark instances of the PDP with time windows. We ran two sets of experiments: the first to optimize different combinations of the objectives number of routes, travel distance and travel time, and the second to optimize different combinations of the previous objectives plus the uncollected profit.

Our analysis is threefold. First, we computed the hypervolume covered by the non-dominated solutions found by both algorithms. For the first set of experiments, we found that GN-MOEA has a better performance over ϵ-MOEA in three out of four combinations of objectives, and, for the remaining case, there is no difference between both algorithms. For the second set of experiments, ϵ-MOEA outperformed GN-MOEA in three out of five combinations of objectives, and GN-MOEA has a better performance in the remaining two cases.

Secondly, we computed the generational distance the non-dominated solutions found by both algorithms. In this case, GN-MOEA solutions have a shorter generational distance than those from ϵ-MOEA for all four combinations of objectives in the first set of experiments and for three combinations of objectives in the second set, and solutions from ϵ-MOEA have a shorter generational distance for the remaining two combinations of objectives in the second set. These results are consistent with those from the hypervolume, except for the combination $f_{\mathrm{R}}f_{\mathrm{P}}$, for which ϵ-MOEA found solutions with a larger hypervolume and GN-MOEA solutions have a shorter generational distance. After analyzing these results, we can argue that this situation is due to that, although GN-MOEA generated solutions very near to the reference set, it did not cover some solutions in the middle of it.

Finally, we analyzed the conflict between objectives. We found that there is no conflict at all between f_{R} and either f_{D} or f_{T}, however objectives f_{D} and f_{T} are in conflict in many instances, and objective f_{P} is in conflict with the other three objectives in all the 56 instances.

After these interesting results, we believe that we can continue with our research by investigating why ϵ-MOEA is not able to find better solutions than those from GN-MOEA for the first set of experiments, that is, what are the properties of GN-MOEA that make it a better solver when only objectives f_{R}, f_{D} and f_{T} are considered. To further analyze the many-objective performance of both algorithms, we are planning to include at least two additional objectives.

References

1. Bent, R., Van Hentenryck, P.: A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. In: Principles and Practice of Constraint Programming. pp. 123–137 (2003)
2. Deb, K., Mohan, M., Mishra, S.: Towards a quick computation of well-spread pareto-optimal solutions. In: Proceedings of EMO 2003. pp. 222–236. Springer (2003)
3. Garcia-Najera, A., Gutierrez-Andrade, M.A.: An evolutionary approach to the multi-objective pickup and delivery problem with time windows. In: 2013 IEEE Congress on Evolutionary Computation. pp. 997–1004. IEEE (2013)
4. Goldberg, D.E.: Genetic algorithms in search, optimization and machine learning. Addison-Wesley (1989)
5. Hasle, G., Kloster, O.: Industrial vehicle routing. Geometric modelling, numerical simulation, and optimization pp. 397–435 (2007)
6. Jozefowiez, N., Semet, F., Talbi, E.G.: Multi-objective vehicle routing problems. Eur. J. Oper. Res. 189(2), 293–309 (2008)
7. Laporte, G.: Fifty years of vehicle routing. Transport. Sci. 43(4), 408–416 (2009)
8. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining convergence and diversity in evolutionary multiobjective optimization. Evolut. Comput. 10(3), 263–282 (2002)
9. Lenstra, J.K., Rinnooy-Kan, A.H.G.: Complexity of vehicle routing and scheduling problems. Networks 11(2), 221–227 (1981)
10. Li, H., Lim, A.: A metaheuristic for the pickup and delivery problem with time windows. In: 13th International Conference on Tools and Artificial Intelligence. vol. 1, pp. 160–167. IEEE Computer Society (2001)
11. Parragh, S.N., Doerner, K.F., Hartl, R.F.: A survey on pickup and delivery problems. J. Betriebswirtschaft 58(2), 81–117 (2008)
12. Ropke, S., Pisinger, D.: An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transport. Sci. 40(4), 455–472 (2006)
13. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms – A comparative case study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.P. (eds.) PPSN V. LNCS, vol. 1498, pp. 292–304. Springer (1998)