Proceedings of the Twenty-Third International Conference on Automated Planning and Scheduling

# Combining a Temporal Planner with an External Solver for the Power Balancing Problem in an Electricity Network

**Chiara Piacentini**
King's College London
*name.surname@kcl.ac.uk*

**Varvara Alimisis**
Durham Energy Institute
*varvara.alimisi@durham.ac.uk*

**Maria Fox, Derek Long**
King's College London
*name.surname@kcl.ac.uk*

### Abstract

The electricity network balancing problem consists of ensuring that the electricity demands of the consumers are met by the committed supply. Constraints are imposed on the different elements of the network, so that damage to the equipment is prevented when transformers are stepped up or down, or generation is increased. We consider this problem within *zones*, which are subnetworks constructed using carefully chosen decomposition principles. The automation of decision making in electricity networks is a step forward in their management which is necessary for coping with the increase in power system complexity that we expect in the near term. In this paper we explore the deployment of planning techniques to solve the zone-balancing problem. Embedding electricity networks in a domain description presents new challenges for planning. The key point is that the propagation of information requires complex updates to the state when an action is applied. We have developed a method in which the computation of the critical numeric quantities is performed calling an external power flow equation solver, demonstrating a clean interface between the planner and this domain-specific computation. This solver allows us to move the power flow computations outside of the planning process and update the values efficiently. We also examine a second important feature of this problem, which is the interaction between exogenous events and constraints over the entire plan trajectory within a zone.

## 1 Introduction

In this paper we explore an application of planning technology to the problem of balancing generated power and demand in an electrical power system. As we move towards the decarbonised energy future of 2050, the increasing complexity and partial observability of the network must be managed. To cope with these problems, decisions are to be made locally rather than by a single centralised authority. Each zone must self-manage all network operational decisions in real time, in order to meet the goals presented by the various stakeholders. Future energy networks must therefore be organised to provide increased flexibility and controllability through the provision of appropriate real time decision-making techniques. Within this context, planning is one of the most suitable computer science techniques to achieve automation within zones of an electricity network.

In modelling the autonomous management of an electricity network we have faced new issues for planning. The domain is characterised by small local events (such as stepping up a transformer) that have global effects and non-linear numeric effects. Furthermore, all planning must take place against the background demand curve, which must be represented as a sequence of timed initial fluents. The relationship between the demand curve and the plan requires the introduction of new techniques to make the heuristic aware of the effects of the timed initial fluents. This requires a significant change to the way that a relaxation-based heuristic is computed.

In the electricity network there are different elements such as busbars, lines, generators, transformers and loads and several numeric quantities that must be reasoned about. The most important numeric quantities are voltages and phase angles on the busbars. The behaviour of these quantities are affected by every single element of the network. This implies that every time we change a parameter of the network, the numeric effects propagate all over the system. The planning task is to serve a given demand, maintaining the balance between the power generated and the power consumed and satisfying other constraints on the voltage of the busbars and the power flowing into the lines. The key difference between this and the benchmark domain considered so far in planning, is that the calculation of the numeric quantities requires the solution of a set of non-linear equations (*power flow*). This calculation needs to be done every time a change in the system occurs and it affects every element of the system.

In order to deal with these new features we have integrated an existing planner (POPF) with an external module that solves power flow problems. The goal of the planning problem is to reach the end of the 24 hour period over which the zone is to be balanced. The external module receives parametrised calls from the planner and passes back the power flow solutions. We have extended the POPF heuristic to take account of the timed initial fluents representing the demand. Without this change, the heuristic values are always zero, because the demand is ignored and since the initial state respects all constraints, no actions need to be taken to achieve the goal state. An alternative approach used to tackle problems with electricity networks is presented in the paper (Coffrin et al. 2012). In this case the issue of non-linearity of the equations is overcome by introducing a linear-programming approximation of the AC power flow equations, using a piecewise-linear approximation of the cosine. This approach is very specific to the electricity network, while our solution is general and extends planners to the management of a new class of interesting numeric problems, since many real world domains feature background numeric events.

In this paper we first describe the planning problem, then we motivate the choice of the external solver and its implementation. We also discuss some domain-independent modification that we introduced in order to acquire enough expressiveness power. In the following part we describe the PDDL model used, the new heuristic and the results of our implementation in terms of scalability of the size of the network and the number of control points.

## 2 Fundamentals of the power network balancing problem

In this section we describe the general problem that planning attempts to solve.

As mentioned before, our research is motivated by the need for autonomy in the management of the electricity network. In order to deliver electricity from suppliers to consumers different components are interconnected into a network. From the production process to the consumption phase different components are distinct:

- Power generation: various units produce power from combustible fuels (coal, natural gas, biomass) or non-combustible fuels (wind, solar, nuclear, hydro power);
- Transmission network: this is used for the bulk transfer of power over long distances and at high voltages between main load centres;
- Distribution network: from the transmission network the power is stepped down in voltage from a transmission level voltage to a distribution level voltage;
- Demand: electricity is requested by industrial, commercial and domestic system users/customers.

Since no large-scale energy storage devices are available, the power produced by the power stations must be equal to the demand (plus the losses during the transmission and distribution process) at any time. The behaviour of demand is assumed to be roughly predictable throughout the day. In Figure 1 the typical trend of daily load profiles in UK are shown.

Transmission and distribution networks are composed of different elements:

- *Busbars*: they are the nodes of the network and they are characterised by the voltage $\tilde{V}$. Since the electricity networks are AC circuits, the voltage is a complex quantity and it can be expressed with its magnitude $V$ and its phase angle $\theta$.

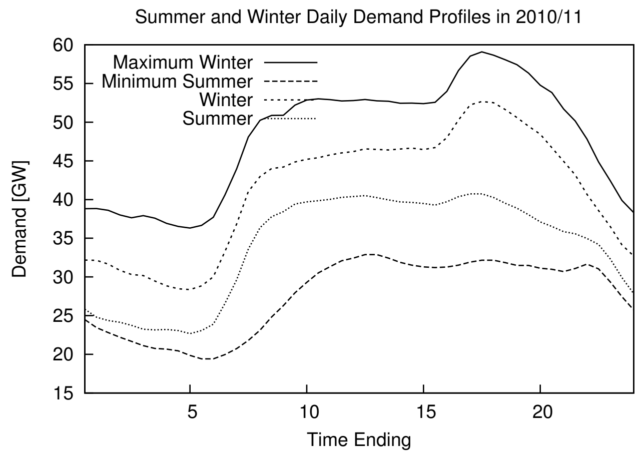


Figure 1: UK loads profile for a typical summer and winter days and minimum and maximum loads for years 2010/2011 (Copyright, National Grid PLC 2010).

- *Lines*: they are electrical conductors carrying alternate current. The physical quantities that must be specified for a line are the resistance, the reactance and the susceptance. These quantities determine the relationship between the voltage and the current flowing into the lines and they can be summarised in the quantity called admittance $Y$.
- *Generators*: they are the units that produce electrical power.
- *Loads*: they are the output at the terminals of the network and they refer to the power consumed.
- *Transformers*: they are devices that convert high voltage power to low-voltage power and vice versa. The regulation of the voltage is realised by a connection point to a coil (*taps*) that can be stepped up and down.

During the power production chain, the transmission and distribution networks must satisfy certain constraints in order to prevent faults and damages to the equipment. In particular two main constraints must be met:

- Voltage constraints: the voltage of each busbar of the network must lie within an upper and a lower limit.
- Thermal constraints: the power flowing into the wires of the circuit must not exceed a given threshold.

In order to provide enough power to satisfy demand at a given time, without exceeding the limits, and within the acceptable voltage margins, there are different controllable elements that can be deployed: the power can be generated from different generators, different branches of the network can be disconnected or connected, transformers can vary the tap settings, thereby increasing or decreasing the voltage of the connected branch, and so on. Each of these actions modifies the *power flow* in the network, thus, every time a change occurs, the voltage on all of the busbars and the power flowing into the wires is affected.

The *power flow* analysis (Glover and Sarma 2001) is the calculation of the voltage magnitude and phase angle at each node (*busbar*) of an electrical network. For a system with N nodes we can write the nodal equations

$$\tilde{Y}\tilde{V} = \tilde{I} \tag{1}$$

where $\tilde{V} \in \mathbb{C}^N$ is the vector of nodal voltages, $\tilde{I} \in \mathbb{C}^N$ is the current injections, and $\tilde{Y} \in \mathbb{C}^N \times \mathbb{C}^N$ is the admittance matrix. Because the admittance matrix is singular, a reference node (*slack busbar*) is defined, its voltage magnitude is set to be 1 pu and its phase angle is zero. Knowing the power injections (generators and loads), for any node $i$, it is possible to write

$$P_i + jQ_i = \tilde{V}_i \left[ \sum_{n=1}^{N} \tilde{Y}_{in} \tilde{V}_n \right]^* \tag{2}$$

where $P \in \mathbb{R}^N$ is the real power and $Q \in \mathbb{R}^N$ is the reactive power. As a by-product of this calculation, real and reactive power flows in equipment and losses can be computed.

### 2.1 The Planning Task

This work applies planning to address the problem of power balancing in zones of control and operation of an electrical power system. We deliberately focus on a *zone* as the complexity of the entire control task contradicts the requirement for real-time decision-making. We can consider planning to be applied for a network zone that would be a physically connected collection of power elements strongly self-contained in terms of actions directed within its territory and, by extension, loosely affected by actions outside its boundaries. For a planner to be provided with appropriately good boundaries, the zone definition partitions a power network by minimising physical power flow dependencies and their possible deviations (Alimisi et al. 2013). The zone definition provides the planner with zones of equal importance from a control point of view, then the planner can plan power balancing operations within each of them.

Although the planner does not have to deal with the complexity of the whole network, it must handle a large number of non linear equations. The power flow problem is non linear due to the constraints placed on power system operation, such as maintaining a bus voltage at 1 volt per unit (pu). The functions $P_i$ and $Q_i$ are non linear functions of the state variables phase angle $\theta_j$ and voltage magnitude $V_j$ (with $j = 1, ..., N$). For a network with $R$ injections this corresponds to a set of $2(N-1)-(R-1)$ non linear equations. From Eq. (2) we can see that if we change an injection variable, the effects propagate over all nodes of the network. Addressing this feature by integrating an external solver with a planner that is equipped with a suitably modified heuristic, is presented and analysed in the next sections and is the main contribution of this paper.

## 3 External Solver

In this section we describe the approach used to tackle the planning problem. First we explain the motivations for the use of the external solver, underlining the features of our domain that make the external solver necessary. In the second part we describe the implementation of the external solver in the planner chosen.

### 3.1 Motivations

In order to deal with the problems of local actions having global effects, and non linear function effects, we have to modify some assumptions that are typical of Planning. Typically, when we model an action we have to specify every possible effect that applying the action will bring about. Instead, we can recognise two possible types of effect: the *direct*, and the *indirect* effects. The direct effects are the ones that the planner can bring about by the application of actions, while the indirect effects are those brought about by the environment in which the action is applied, as a *consequence* of the application of the action. Modelling these indirect effects is similar to the introduction of *events* in PDDL+ (Fox and Long 2006).

We clarify the role of the *direct* and *indirect* effects with an example. In a recent paper (Löhr et al. 2012) on hybrid planning, an example with a *Maze Ball* was presented. In this domain there is a platform that can be inclined through fixed angles. A ball rolls around on it in a way that is determined by the inclinations of the platform. The planning task is to move the ball to a destination position by performing multiple inclinations of the table. The movements of the table are the direct effects of the inclinations chosen by the planner, while the movements of the ball are secondary effects. In the method described in the paper, the position of the ball is calculated after every inclination action and is treated as part of the direct effect of the action. Instead, with our approach, we distinguish the *direct* effect of inclining the table and the consequence that it causes on the movement of the ball. In the same way, in the electricity domain we can separate the direct effects, such as changing the tap-settings of transformers, from the indirect ones, such as the resulting change of voltage. A crucial aspect of the *indirect* effects is that they cannot usually be expressed with arithmetic formulas. Indeed, they might be complex to compute and it might not be possible to express the necessary functions in PDDL. In the electricity domain the relationship between direct and indirect effects is not trivial and the indirect effects cannot be expressed in standard PDDL because they require the use of numeric axioms and these are not provided.

In the following, we will explore some exemplar domains that can be viewed from this perspective. As mentioned before the *maze ball* domain can be seen as an example of one with *indirect effects*. In this case the only indirect effect caused by the inclination of the platform is the variation of the position, velocity and acceleration of the ball. Since the number of *indirect effects* is limited, they can be merged with the *direct* ones. The link between them is solved in a pre-processing stage, so that they can be written as a look up table in the domain. On the contrary, our domain has a large number of objects and a similar approach is not suitable because the dimension of the look-up table blows-up exponentially with the size of the network.

Another example is the PSR domain (Thiebaux and

Cordier 2001). The network considered in this domain is very similar to ours, although it does not consider numeric functions, but only propositions. In this case it is still possible to use standard PDDL because the link between the *direct* and *indirect* effects is made by the use of derived predicates (Thiebaux, Hoffmann, and Nebel 2005). As an example of a domain with increased complexity we can consider a DC circuit, in which we want some controllability on the numeric functions. In this case the link could be made by means of introducing a new PDDL field called a *derived fluent*. Derived fluents are similar to derived predicates, but they update the values of numeric quantities linked to the *indirect* effects.

However, our domain contains an additional difficulty, because the numeric quantities do not depend on each other by simple arithmetic expressions. In order to handle this difficulty we modify an existing planner, introducing a special solver that can recognise the functions described above, and evaluate them during the calculation of the state, according to the specific model of the problem. The idea of using an external solver is a specialisation of the Planning Modulo Theories framework (Gregory et al. 2012), because it exploits the same idea of having a dedicated sub-solver connected to a core-planner by means of special communication constraints.

## 3.2 Implementation

In principle an external solver could be implemented in any planner, but we choose to work with POPF (Coles et al. 2010): the load profiles are naturally expressed as functions of time so we need to use a temporal planner. Among all the temporal planners we decide to use POPF because it is the only one capable of dealing with durative actions with continuous numeric effects, negative timed initial literals and timed initial fluents. These are required to model functions whose trend is predictable. Although the state of the art of POPF has these desirable characteristics, we needed to add more functionality in order to deal with exogenous events and constraints over the entire plan trajectory. More details are presented in a following section.

**POPF Planner.** Before discussing the implementation of the external solver we first give some background about the POPF planner. POPF is a forward-chaining planner based on a partial-order obtained by delaying the commitment to ordering decisions, time-stamps and the values of numeric parameters. A state in a general temporal planning problem can be characterised by a tuple $< F, V, Q, P, C >$, where:

- $F$ is the set of atomic propositions that hold in the state;
- $V$ is the vector of values of the task numeric variables;
- $Q$ is a list of actions whose execution has begun but not yet finished;
- $P$ is the plan to reach the current state;
- $C$ is a list of temporal constraints over the steps in $P$.

When an action is added to the plan, the state is updated. In order to extend the forward search to support partial-order planning in POPF, the state representation is extended

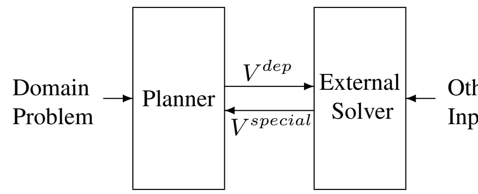


Figure 2: Planner and external solver implementation

adding new elements to the tuple in order to take into account when a proposition or a numeric variable is first recorded or when it is required to hold.

The heuristic evaluation is realised with a modified version of the Temporal Relaxed Planning Graph (TRPG).

**External Solver Implementation.** The architecture of the external solver is shown in Figure 2.

The planner takes as input the domain and the problem PDDL files. In order to deal with the local actions with global effects and non linear functions, we divide into different categories the vectors of numeric variables in the state:

- $V^{special}$ is a vector of variables that are global numeric effects that cannot be expressed with linear functions. This vector of variables records the *indirect effects* described previously;
- $V^{dep}$ is a vector of variables that influence $V^{special}$;
- $V^{indep}$ is a vector of variables that do not influence $V^{special}$.

When the planner updates a state, $V^{dep}$ and $V^{indep}$ are recorded. If there is a change in one of the $V^{dep}$ variables, then the $V^{special}$ values are calculated calling the external solver, using as input all the updated values of the $V^{dep}$ variables and other optional information of the problem. At the outset we used IPSA (TNEI Services Ltd 2012), a power engineering licensed software specialised for power flow calculations, to calculate these values, but due to the computational cost of communicating with IPSA, we decided to encode directly the power equations and solve them using the Newton-Raphson algorithm (Ypma 1995), an iterative numeric algorithm that allows us to find successively better approximations to the zeros of a function. In order to execute the calculation, the external solver needs as input the configuration of the network (the links between the different elements) and information about the resistance and impedance of wires. An advantage to using an external solver, instead of trying to model everything in PDDL, is that we do not need to express variables in the domain that are not relevant to the search (such as impedance), but are necessary to perform the power flow calculations.

For the heuristic evaluation, the relaxation of numeric state variables used in POPF is based on the same principle introduced in metric-ff (Hoffmann 2003) of ignoring delete of linear numeric functions. As the $V^{special}$ variables cannot be expressed as linear numeric functions, in the heuristic evaluation we use an approximation of the special functions

```
(:durative-action constraint-check
   :parameters ()
   :duration (<= ?duration 1000)
   :condition (and
      (at start (S))
      (at end (F))
      (over all (forall (?g - slack-gen)
         (and (>= (ipsa-slack-p ?g)
             (p-minimum-gen ?g))
         (<= (ipsa-slack-p ?g)
             (p-maximum-gen ?g))
         (>= (ipsa-slack-q ?g)
             (q-minimum-gen ?g))
         (<= (ipsa-slack-q ?g)
             (q-maximum-gen ?g)))))
      (over all (forall (?li - line) (and
         (<= (ipsa-line-power ?li) 25)
         (>= (ipsa-line-power ?li) 0))))
      (over all (forall (?b - Bus) (and
         (<= (ipsa-voltage ?b) 1.06)
         (>= (ipsa-voltage ?b) 0.94)))
   :effect (and (at end (is-end))))
```

Figure 3: `constraint-check` action in the temporal model.

that indicate whether an action increases, decreases or is irrelevant to $V^{special}$. The amount of increase or decrease of the $V^{special}$ is determined in a pre-processing stage and it is explicitly written in the PDDL model as effects of the actions on the $V^{special}$ variables. The heuristic evaluation is further modified in order to improve the management of global constraints in the presence of exogenous events. More details will be explained in Section 5.

# 4 Domain Description

In this section we describe the temporal model in PDDL used to encode the problem defined previously.

## 4.1 The Domain

In our domain the $V^{special}$ variables are identified with the prefix `ipsa`. They are the voltages at each of the busbars, the power flowing into the lines and the power (real and reactive) generated by the generator connected to the slack busbar. For this generator the power is calculated by the power flow, while for the others it is decided by the actions.

The other variables, such as the tap setting of transformers, the power from other generators and the power that can be shed are put in the $V^{dep}$ subset.

In order to model the numeric constraints that must be satisfied during the entire duration of the plan, such as the thermal limit on the lines, the voltage boundaries on the busbars, and the maximum power that a generator can produce, we introduce a durative action called `constraint-check` (Figure 3). This action is forced to start at the beginning of the plan, requiring as an `at start` precondition a proposition that is deleted by a negative timed initial literal after few time steps. The action has also, as an `at end` condition, a proposition that is asserted by a timed initial literal

```
(:durative-action step-down-tap
   :parameters (?t - tap)
   :duration (= ?duration 3)
   :condition (and
      (at start (is-available ?t)))
   :effect (and
   (at end (decrease (tap-level ?t) 1))
   (at start (not (is-available ?t)))
      (forall (?b - bus) (and
         (at end (increase
         (ipsa-voltage ?b)
         (step-tap-min ?t ?b)))
         (at end (decrease (ipsa-voltage ?b)
         (step-tap-min ?t ?b)))))
   (at end (is-available ?t))))
```

Figure 4: `step-down-tap` action in the temporal model.

after the last change in load profiles. These `at start` and `at end` conditions ensure that the constraint check action envelopes the entire plan so that all other activities are concurrent with it. Then the numeric constraints that must always be observed can modelled as `over all` conditions for this action. The trick of using envelope actions in this way provides a simple method for enforcing trajectory constraints in a plan.

The other actions are expressed as durative actions, depending on their effects. In Figure 4 we can see an example of a durative action. It models a tap change of a transformer. Once the tap changing is performed, the next one is constrained on when to be done, thus the transformer is expected to not be available for other operation on the tap for a specified time interval. This mechanism is modelled requiring as a precondition the proposition `(is-available ?t)`. This condition is deleted by a start effect and restored by an end effect. The tap position is a $V^{dep}$ variable, while the voltage and power flowing into lines are $V^{special}$. In the PDDL model the effects on these variables are not all explicitly written, but only the effects relevant to the heuristic appear. In this case the main effect of changing a tap position is to change the voltage on the closest busbars.

Other possible actions are the increasing or decreasing of the power of generators, the connection or disconnection of branches of the network with switches and the shedding of some loads. Using a temporal model allows us to express some temporal constraints, such as guaranteeing an energy request before a time point, or exceeding some numeric limits for some amount of time.

## 4.2 The Problem

The instances of the problem are the predictable demand and the configuration of the network. To model the fluctuating demand in the initial state a series of timed initial fluents are listed, as shown in Figure 5.

The goal of the plan is to guarantee the satisfaction of the demand and the constraints on busbars and lines over the whole time interval considered. This is modelled by requiring as the goal state to have reached the end of the

```
(at 0 (= (p-level load1) 7.357))
(at 0 (= (q-level load1) 0.541))
(at 0 (= (p-level load2) 6.217))
(at 0 (= (q-level load2) 1.741))
(at 0 (= (p-minimum-shedding load1) 4.426))
(at 5 (= (p-level load1) 8.357))
(at 5 (= (q-level load1) 1.541))
(at 5 (= (p-level load2) 2.217))
(at 5 (= (q-level load2) 0.741))
(at 5 (= (p-minimum-shedding load1) 4.426))
```

Figure 5: A fragment of the set of timed initial fluents.

`constraint-check` action.

## 5 Exogenous event

In this section we examine the role of numeric exogenous events (Edelkamp and Hoffmann 2004) in relation to global constraints, and how we modify the heuristics in order to have a more informative evaluation.

First we can consider the possible effects that a timed initial literal can have:

1. Delay resource: TILs could add precondition of actions;
2. Remove resource: TILs could prevent actions.
3. Violate active condition: TILs can violate a global condition. These can be caused by direct or indirect effects.

The first case is trivial and it is handled by building a TRPG that adds the effect when the TIL enables it. The second case represents a deadline, so it should affect when actions are reachable and hence when conditions are reachable too. The last case is the most interesting one: if an active action has an over all condition ($c$) and we have a TIL ($t$) and we have $t \to \neg c$, then the violation is inevitable. Instead, if we have a proposition $s$ such that $t \wedge s \to \neg c$, then we must enforce $t \wedge c \to \neg s$, so that $t \to \neg s$ has to be propagated.

As we have described in the previous section we need timed initial fluents to express the behaviours of the load profiles. If the new set of parameters introduces an indirect effect on the voltage of a busbar that exceeds the allowed limits, we need to apply an action (for example: step up the tap of a transformer) before the TIL, that brings the voltage of the busbars into the acceptable limits when the TIL is applied.

During the heuristic evaluation we need to take into account this effect in order to have a more informative heuristic. The construction of the TRPG is done in the forward direction as in the original POPF planner. During the extraction of the relaxed plan we need to take into account all the active constraints introduced by the open actions. Starting from the goal layer containing the goal, we proceed backwards, applying the actions needed for achieving the goal. If a TIL violates an active constraint, then an extra action taken from a previous layer is added in the relaxed plan, modifying the heuristic value. If the TIL that violates the constraint is the next action that should be applied, then the new action added becomes a helpful action. Among the possible actions

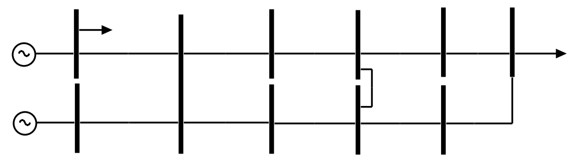

Figure 6: Example of circuit.

```
0.001: (constraint-check) [25.000]
6.501: (shed-load load1) [17.000]
9.001: (shed-load load1) [14.501]
9.002: (shed-load load1) [14.501]
10.001: (shed-load load1) [13.503]
10.002: (shed-load load1) [13.503]
```

Figure 7: Example plan.

that can be applied to restore the constraint, the action that is chosen as helpful is the one that needs to be applied next.

## 6 Evaluation

In this section we discuss results of the application of our planner with our model. First we examine the scalability of the planner in terms of the size of the network, while in the second part we evaluate domains with an increasing number of control points, showing that is possible to move towards bigger networks.

### 6.1 Scalability of the Size of the Network

We apply the external solver and the model previously described for circuits, with an increasing number of busbars and lines. An example of a network is shown in Figure 6. It is composed of two generators: one connected to the slack bus and an external generator (so power can be imported or exported, according to its availability). There are two loads, of which one can be shed, and different lines and busbars subjected respectively to a thermal and a voltage constraint. The slack generator must satisfy the requirement of a minimum and maximum (real and reactive) power production. We want to produce a plan for one day, knowing the load profile every half an hour. We run the model with different load profiles, taken from the data set of National Grid for several winter days of 2010 (National Grid PLC 2012). The total demand is scaled by a constant factor in order to be consistent with the demand of two loads. The total demand is divided into two load profiles following the typical behaviour of domestic and industrial profiles, according to the data of UKGDS. In this kind of domain if no decision is made, there is a violation on the limit of power produced by the slack bus, so the planner can decide whether to import power from an external generator or shed a load. An example resulting plan is shown in Figure 7.

Figure 8 shows the results of the experiments that have different daily load profiles every 5 days from 5 November to 30 December 2010. In the left plot the number of states

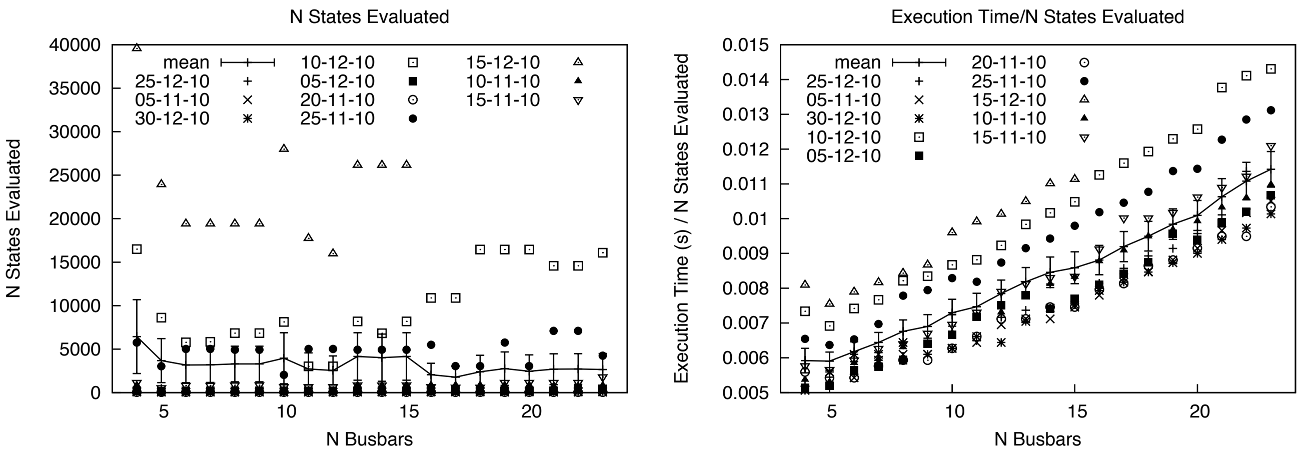


Figure 8: Results showing scaling with the number of busbars. The different points represents the different load profiles.

evaluated in function of the number of busbars (the complexity of the network) is shown and it indicates that the time spent just on the search is constant. In the right plot we show the execution time over the number of states evaluated, that can be seen as a measure of the time spent for the evaluation of a single state, in function of the number of busbars. As we can see from the plot, execution time increases with more complex networks, but this is not alarming because the increase is linear and the slope is small.

### 6.2 Scalability of the Control Points

The second evaluation that we perform is in terms of scalability of the number of control points present on the circuit. For this test we take the circuit shown in Figure 9 (Currie et al. 2007). It is a model taken from the Centre for Distributed Generation and Sustainable Electrical Energy of a 33 kV rural network fed from a 132 kV supply point. We simplified the original network excluding a sub-sea cable and putting a single transformer when there are two connecting the same busbars. This network consists of 61 busbars, 64 lines, 18 loads and 21 transformers, that are our control points.

Depending on the values of the power consumed by the load, voltage excursions may arise, but they can be controlled by setting a different value of the tap ratio on the appropriate transformer.

For this domain we generated different problems changing the load profiles and the number of transformers present in the circuit. We start with 3 transformers, increasing by one until the configuration in Figure 9. We run the planner for a maximum of 30 minutes.

An example plan is shown in Figure 10, while in Figure 11 we can see how the plan scales with the number of control points. In the left plot the number of states evaluated in function of the number of transformers is shown, while the right plot represents the variation of the execution times over the number of states evaluated in function of the number of transformers in the circuit. Also in this case we can see that there is a linear increase of the execution time depending on the calculation of the power flow, but the number of states evaluated is constant with respect to the number of transformers.

```
0.001: (constraint-check) [25.500]
8.501: (step-down-tap tap1) [3.000]
16.001: (step-down-tap tap1) [3.000]
16.002: (step-down-tap tap14) [3.000]
17.501: (step-down-tap tap18) [3.000]
```

Figure 10: Example of plan.

## 7 Future Work

Reasoning about network balancing takes place on a scale of seconds and minutes, because the physical operations of the network are mechanical and therefore time-consuming. While for problems with a granularity of milliseconds reactive control is most suitable, the network balancing time scale suggests that planning is an appropriate approach. However, more work can be done in order to improve the performance of the planner. The effects of the actions on the indirect effects are considered in the heuristic evaluation as approximated constants and they are hand written in the PDDL model. Instead, there are cases in which it would be more appropriate to consider a more realistic approximation of the indirect effects, linearizing the power flow equations or using more accurate approximations (Coffrin et al. 2012).

In the experiments performed so far we mainly considered the operations of shedding loads, tap changing and generating power from units. In the model the line-switching operation is also available, but this action presents different features with respect to the first considered: connecting or disconnecting two different branches of the networks results in a change in the topology of the network and an appropriate heuristic needs to be considered.

In the future we will also include the costs of actions (both component wear and tear and market tariffs are being modelled by other partners in our project) and we will try to

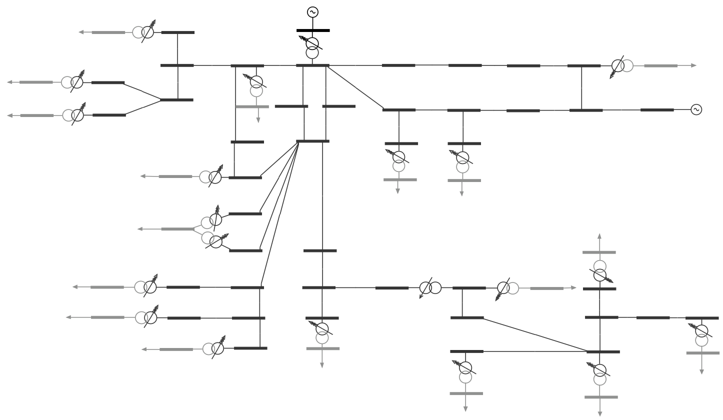

Figure 9: The small 33 kV rural network.

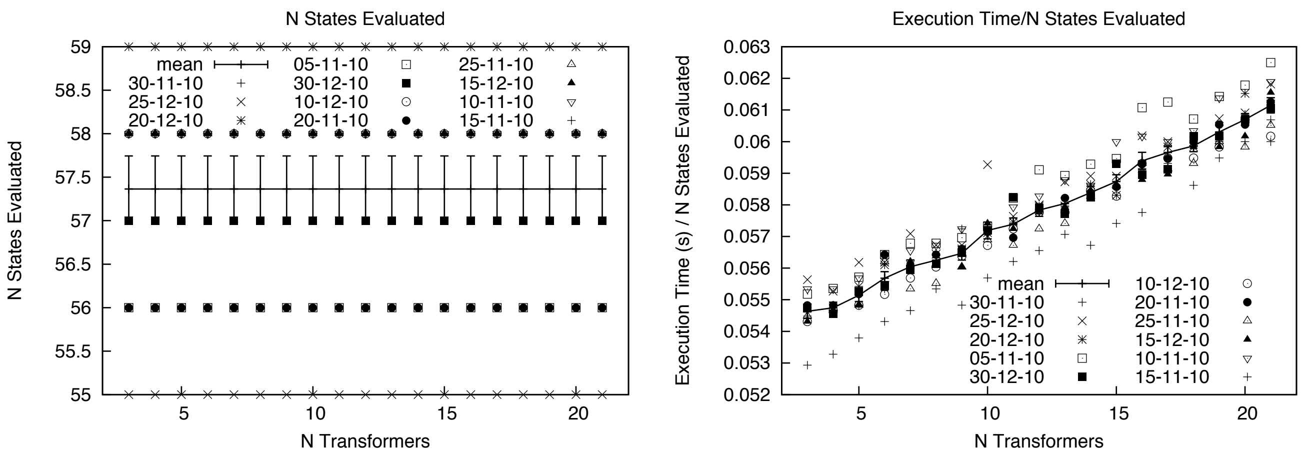


Figure 11: Results showing scaling with the number of decision points (transformers). The different points represents the different load profiles.

find plans that minimise these costs. This kind of domain, like the one presented in the paper (Tierney et al. 2012) has the peculiarity that the cost depends also on the duration of the action, but we have in addition the interaction with the indirect effects. In addition, we will consider replanning to handle uncertainty in the demand and power commitments. These curves are assumed at the outset of planning, but in fact they are subject to some variation so the planner must be responsive to the breakdown of these assumptions when plan steps are executed.

## 8 Conclusion

In this paper we have presented an application of planning techniques to the management of an electricity network. The task of the planner is to provide power to serve a predicted demand, respecting some constraints on elements of the network. An important difficulty of this problem is that effects propagate all over the network and they cannot be expressed in simple linear functions. We showed that planners can handle these effects using a specific solver that communicates with the planner, passing back the results of particular power flow computations. With this approach it is possible to use planning techniques to solve network balancing problems on electricity networks with increasing numbers of elements and control points.

In the paper we have also analysed an important domain-independent feature, that was necessary to include to solve our domain. In the presence of exogenous events and trajectory constraints we can infer some information that should be considered in the heuristic evaluation. This is an important feature that is not specific to this domain but arises in many domains in which there is interesting numeric background behaviour. We showed that the modified heuristic allows the planner to scale with the number of control points in the network.

## 9 Acknowledgments

This work is supported by Engineering and Physical Sciences Research Council (EPSRC) as part of the project Autonomic Power System.

## References


Alimisi, V.; Piacentini, C.; King, J. E.; and Taylor, P. C. 2013. Operation and control zones for future complex power systems. In *5th annual IEEE Green Tech Conference*.

Coffrin, C.; Hentenryck, P. V.; Bent, R.; Coffrin, C.; Member, S.; Hentenryck, P. V.; and Bent, R. 2012. Approximating line losses and apparent power in ac power flow linearizations. In *Proceedings of the 2012 IEEE Power and Energy Society General Meetings (PES 2012)*.

Coles, A. J.; Coles, A. I.; Fox, M.; and Long, D. 2010. Forward-chaining partial-order planning. In *Proceedings of the Twentieth International Conference on Automated Planning and Scheduling (ICAPS 10)*. AAAI Press.

Currie, R.; Ault, G.; Foote, C.; and McDonald, J. 2007. Active power-flow management utilising operating margins for the increased connection of distributed generation. *Generation, Transmission Distribution, IET* 1(1):197 –202.

Edelkamp, S., and Hoffmann, J. 2004. Pddl2.2: The language for the classical part of the 4th international planning competition. In *Technical report IPC*.

Fox, M., and Long, D. 2006. Modelling mixed discrete-continuous domains for planning. *Journal of Articial Intelligence Research* 27:235–297.

Glover, J. D., and Sarma, M. S. 2001. *Power System Analysis and Design*. CL-Engineering, 3 edition.

Gregory, P.; Long, D.; Fox, M.; and Beck, J. C. 2012. Planning modulo theories: Extending the planning paradigm. 65–73.

Hoffmann, J. 2003. The metric-ff planning system: Translating ”ignoring delete lists” to numerical state variables. *Journal of Artificial Intelligence Research. Special Issue on the 3rd International Planning Competition* 20.

Löhr, J.; Eyerich, P.; Keller, T.; and Nebel, B. 2012. A planning based framework for controlling hybrid systems. *Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling (ICAPS 12)* 164–171.

National Grid PLC. 2012. National grid uk. http://www.nationalgrid.com/uk/Electricity/Data/Demand +Data/.

Thiebaux, S., and Cordier, M. 2001. Supply restoration in power distribution systems - a benchmark for planning under uncertainty. *6th European Conference on Planning (ECP 01)* 85–95.

Thiebaux, S.; Hoffmann, J.; and Nebel, B. 2005. In defense of pddl axioms.

Tierney, K.; Coles, A. J.; Coles, A. I.; Kroer, C.; Britt, A.; and Jensen, R. M. 2012. Automated planning for liner shipping fleet repositioning. In *Twenty Second International Conference on Automated Planning and Scheduling (ICAPS 12)*. AAAI Press.

TNEI Services Ltd. 2012. Ipsa power. http://http://www.ipsa-power.com/.

Ypma, T. J. 1995. Historical development of the newton-raphson method. *SIAM Rev.* 37(4):531–551.