KATHOLIEKE UNIVERSITEIT LEUVEN

WP EN2013-07

Enhanced priority list unit commitment method for power systems with a high share of renewables

E. Delarue, D. Cattrysse, W. D'haeseleer

TME WORKING PAPER - Energy and Environment
Last update: July 2013 2012

KULeuven Energy Institute
TME Branch

Enhanced priority list unit commitment method for power systems with a high share of renewables

E. Delarue[a], D. Cattrysse[b], W. D'haeseleer[a,]*

[a] *University of Leuven (KU Leuven) Energy Institute, TME branch (Applied Mechanics and Energy Conversion), Celestijnenlaan 300A box 2421 / B-3001 Leuven / Belgium.*

[b] *University of Leuven (KU Leuven) Centre for Industrial Management / Traffic and Infrastructure, Celestijnenlaan 300A box 2422 / B-3001 Leuven / Belgium.*

E-mail addresses: erik.delarue@mech.kuleuven.be ; dirk.cattrysse@cib.kuleuven.be ; william.dhaeseleer@mech.kuleuven.be

** Corresponding author: Tel.: +32 16 322511; fax: +32 16 322985.*

Abstract

Over the past decade, mixed integer linear programming (MILP) has become the preferential way to solve unit commitment problems for electricity generation. However, as this paper demonstrates, this method faces increasing computational difficulty when the residual demand (i.e., original demand minus variable renewables) in a power system sinks to relatively low values. Hence, a new unit commitment method is developed to specifically solve problems with low residual demand. The algorithm is set up based on an enhanced priority list of power plants (EPL). Plants are activated according to this list, while schedules are adapted to respect technical restrictions as minimum up and down times, and minimum operating points (particularly important in a low residual demand setting). The developed EPL algorithm is compared to a MILP set up, first on a benchmark system and second on a low residual demand case. Results are compared in terms of output and computational effort. Performance of the EPL algorithm is very satisfying (both in terms of optimality and calculation speed), thereby demonstrating its usefulness for real-life simulation and policy analysis, or, e.g., to be used in combination with MILP solvers to provide a starting solution.

Keywords: unit commitment; power generation scheduling; priority list; renewables; low residual demand

Nomenclature

Sets

I	Power plants (index i; number of elements *ni*)
J	Time periods (index j; number of elements *nj*)
L	Stepwise linear segments of output cost curve (index l; number of elements *nl*)
R	Ranked power plants (index r)
T	Time periods, up to period j (index t; number of elements *nt*)

Parameters

A	Fuels cost at minimum working point
a, b, c	Cost coefficients
cc	Cold startup cost
D	Demand
F	Slope of segment of linearized cost
FD	Weighting factor
FMU	Parameter used in up time correction
G	Generation level to be used in cost metric M
hc	Hot startup cost
Inist	Initial state
L	Lower bound of operating range
LD	Low residual demand

M Cost metric

MDT Minimum down time

MS Cost metric for startup cost

MUT Minimum up time

Pmax Maximum output

Pmin Minimum working point

R Reserves

SC Startup cost

T Upper limits of power segments (linearized cost function)

tcold Offline period determining startup (hot or cold)

U Upper bound of operating range

Variables

dt Down time

fc Fuel cost

g Generation

sc Startup cost

tc Total cost

ut Up time

z Commitment status

δ Generation in power segment

Abbreviations

EPL	Enhanced Priority List
ISO	Independent System Operator
MILP	Mixed Integer Linear Programming
PV	Photovoltaics
RES	Renewable Energy Sources
UC	Unit Commitment

1 Introduction

The demand and supply of electricity need to be in constant balance. As demand for electricity has a typical diurnal, weekly and seasonal pattern, power plants need to be carefully scheduled to meet this fluctuating demand. This scheduling optimization is known as the unit commitment (UC) problem and has been widely discussed in the literature. Traditionally, the UC problem was solved centrally, minimizing overall system cost. With the liberalization of electricity markets worldwide, the aim is to operate the electricity generation systems with higher (economic) efficiency. Focus has shifted to optimal economic performance and profit maximization. On the one hand, the UC problem can be considered from a system's perspective, i.e., the so-called security-constrained UC [1]. This type of UC is similar to the traditional UC and is what an Independent System Operator (ISO) currently deals with. Also towards policy making and planning, this UC is useful as a tool to perform market simulations and assess the impact of specific measures. On the other hand, from the viewpoint of a single market player, a price-based UC problem can be considered, optimizing output towards maximum profit, based on electricity price forecasts [2]. This paper will focus on the first kind of UC, i.e., security-constrained UC. For the sake of simplicity, in the remainder of this paper, we will refer to this just as UC.

A wide range of solution techniques for the UC problem have been proposed and developed over the years. Examples include priority listing (heuristics), Lagrangian relaxation, dynamic programming, genetic algorithms, etc., together with hybrid methods combining several of these. For an overview of methods, see, e.g., [3], and further in this paper. Over the past decade, especially mixed integer linear programming (MILP) has been put forward [4, 5].

Electric power systems worldwide are, however, changing. To cut greenhouse gas emissions and for reasons of security of supply (in terms of strategic primary energy security), a massive deployment of renewable energy sources (RES) like wind and solar photovoltaics (PV) is currently taking place or at least aimed for, in certain countries (e.g., Germany) or states (e.g., California). These renewable sources are characterized by a high degree of variability, i.e., they only produce electricity when the wind blows or when the sun shines. A further massive deployment of these intermittent renewables will render a net or residual load profile (as seen by the conventional dispatchable power plants) that is both lower and more volatile[1]. As will be demonstrated in this paper, computation times of MILP UC models increase heavily when (residual) demand levels are low compared to the overall system size. Hence, the aim of this paper is to set up an adequate UC optimization tool that is able to cope with variable and low net demand profiles in an efficient way.

Such model is relevant for several market parties, all with their specific objectives. It can for example serve as an algorithm in market simulation exercises for planning purposes, e.g., by market operators. It is further usable for energy and climate policy evaluation and assessment, focusing on RES integration. Also electricity generating companies can use this kind of model, e.g., to provide operational solutions on relatively large sale, or use the model in combination with other techniques such as MILP, to provide a starting solution.

Towards this end, a new enhanced priority list (EPL) based method is developed. Several priority list methods have been developed in the literature, e.g., [6-9]. These have a main focus on computational speed and on increasing the level of optimality. However, these models are typically not directly suited to be used in settings with low residual load (where minimum load problems occur, or downward reserves become relevant). Furthermore, as far as the impact of intermittent RES (e.g., wind) on generator scheduling algorithms is concerned, the focus in the literature has been mainly on dealing with the uncertainty (see, e.g., [10, 11]). The issue of increasing computational effort for systems with low residual demand has not really been addressed so far.

Hence, in this paper, a new UC method is developed specifically for low residual demand settings. As will be demonstrated, this new EPL based algorithm extends existing priority list methods such as, e.g., [9], not only regarding computational efficiency, but especially regarding feasibility for low residual load problems.

[1] Residual demand is the load to be covered by the centralized (i.e., dispatchable) system; it equals the total demand minus generation by e.g., wind and solar PV.

This paper proceeds as follows. The next section presents the basic formulation of the UC problem. The third section describes the algorithm of the newly developed EPL method. The fourth section provides the input data used in the simulations. Numerical results are then presented in the fifth section. In a first case, the MILP and EPL models are used on a benchmark system. In a second step, a specific low load case is set up and optimized (again both with MILP and EPL). The difficulties faced by MILP to solve such problems are demonstrated, while the adequate performance of the EPL method is shown. The final section concludes.

2 Problem formulation

A basic cost based unit commitment optimization problem is considered. This problem has been described widely in the literature. The description as presented below is partly based on [4].

The objective to be minimized is the total generation cost *tc*, which is equal to the sum of fuel costs *fc* and startup costs *sc* over all power plants *i* and all time periods *j* (in this case hourly time steps):

$$minimize \quad tc = \sum_{i,j} fc(i,j) + \sum_{i,j} sc(i,j) \tag{1}$$

The fuel cost of a power plant is typically a quadratic function of its output *g* and commitment status *z* (with *a*, *b* and *c* the cost coefficients, *I* the set of power plants and *J* the set of time periods) [12]:

$$fc(i,j) = a_i \cdot z(i,j) + b_i \cdot g(i,j) + c_i \cdot g(i,j)^2, \ \ \forall i \in I, \forall j \in J \tag{2}$$

This (convex) quadratic cost function can be linearized by a number of stepwise linear segments (set *L*, index *l*). Let *Pmax* and *Pmin* be the maximum and minimum power output (if online) respectively, *A* the fuel cost at minimum output, F_l the slope of the cost function of segment *l*, and T_l the upper bound power limit of each segment (note that in this case for the last segment *nl*: $T_{nl} = Pmax$). With δ the actual generated power in each segment *l*, the fuel cost and generation limits are set by the following equations:

$$fc(i,j) = A_i \cdot z(i,j) + \sum_l F_{i,l} \cdot \delta(i,j,l), \ \ \forall\, i \in I, \forall\, j \in J \tag{3}$$

$$g(i,j) = Pmin_i \cdot z(i,j) + \sum_l \delta(i,j,l), \ \ \forall\, i \in I, \forall\, j \in J \tag{4}$$

$$g(i,j) \leq Pmax_i \cdot z(i,j), \;\; \forall\, i \in I, \forall\, j \in J \tag{5}$$

$$\delta(i,j,l) \leq T_{i,l} - T_{i,l-1}, \;\; \forall\, i \in I, \forall\, j \in J, \forall\, l = 2 \ldots nl \tag{6}$$

$$\delta(i,j,l) \leq T_{i,l} - Pmin_i, \;\; \forall\, i \in I, \forall\, j \in J, l = 1 \tag{7}$$

$$\delta(i,j,l) \geq 0, \;\; \forall\, i \in I, \forall\, j \in J, \forall\, l \in L \tag{8}$$

Parameter A_i is the cost [\$/h] at minimum output, and determined as

$$A_i = a_i + b_i \cdot Pmin_i + c_i \cdot Pmin_i^2, \;\; \forall i \in I, \forall j \in J \tag{9}$$

The commitment status z is a binary variable:

$$z(i,j) \in \{0,1\}, \;\; \forall\, i \in I, \forall\, j \in J \tag{10}$$

The startup cost is a function of the time t the plant has been offline, previous to the startup. This is implemented as follows, with parameter SC_t presenting the startup cost if previously shut down for t hours (set T):

$$sc(i,j) \geq SC_{i,t} \cdot [z(i,j) - \textstyle\sum_{n=1}^{t} z(i,j-n)], \;\; \forall\, i \in I, \forall\, j \in J, \forall\, t \in T \tag{11}$$

$$sc(i,j) \geq 0, \;\; \forall\, i \in I, \forall\, j \in J \tag{12}$$

During all time periods j, the sum of the power generated g of all the power plants should be equal to the demand D_j:

$$\textstyle\sum_i g(i,j) = D_j, \;\; \forall j \in J \tag{13}$$

Furthermore, a certain amount of system reserves R_j need to be present in the system, both up- and downwards:

$$\textstyle\sum_i Pmax_i \cdot z(i,j) \geq D_j + R_j, \;\; \forall j \in J \tag{14}$$

$$\textstyle\sum_i Pmin_i \cdot z(i,j) \leq D_j - R_j, \;\; \forall j \in J \tag{15}$$

Finally, the minimum up and down times (*MUT* and *MDT*, respectively) are imposed as follows:

$$z(i,j) - z(i,j-1) - z(i,j+n) \leq 0, \quad \forall\, i \in I, \forall\, j \in J, \forall\, n = 1 \dots MUT_i - 1 \qquad (16)$$

$$z(i,j-1) - z(i,j) + z(i,j+n) \leq 1, \quad \forall\, i \in I, \forall\, j \in J, \forall\, n = 1 \dots MDT_i - 1 \qquad (17)$$

Note that corrections need to be made to these equations for the start (initial conditions) and the end of the considered time interval (set *J*). The UC problem consists of the objective (1) and the constraints (3)-(17).

3 Enhanced priority list (EPL) based algorithm

To solve the UC problem as outlined in the previous section, two methods are used. A MILP model is set up, basically by direct implementation of the equations from Section 2. Second, a new model is developed based on enhanced priority listing (EPL), specifically focusing on low demand issues. This section presents this newly developed algorithm. The algorithm comprises of different steps, which are sequentially passed through. These steps are presented in Figure 1 and are subsequently discussed below.

[Figure 1 about here]

3.1 Ranking of power plants

In the considered UC problem, fuel costs and startup costs are taken into account. The fuel cost is dependent on the power plant's output, and is described by a quadratic cost function. Hence, there is no single marginal or average operational cost for a power plant. As a criterion for the ranking of power plants, one could take the average cost of the power plant at its minimal working point, at its maximum power output, or indeed anywhere in between. According to the chosen metric, a different stacking order can be obtained. This is illustrated in Figure 2. The cost curves of two different power plants are presented (these correspond to plants 4 and 5, respectively, from the system as used further in the simulations). The segments for the stepwise linear approximation are also indicated (8 segments). One can see that if the average cost at minimum output is chosen as metric (slope of the steepest dashed and dotted lines), plant 5 is preferred over plant 4 (i.e., listed first). However, if one opts for the average cost at maximum output, the order reverses. To some extent, the choice of the metric should be based on the expected operating point of a power plant on the margin, i.e., where the order will have an important impact. In our model, the metric is initially set equal to the average cost in the middle of the operating range of the power plant (i.e., the slope from the origin to the fifth marker in the figure). This cost metric is denoted *M* and expressed as follows:

$$M_i = \frac{a_i + b_i \cdot G_i + c_i \cdot G_i^2}{G_i}, \quad \forall i \in I \tag{18}$$

with

$$G_i = \left(\frac{Pmax_i + Pmin_i}{2}\right), \quad \forall i \in I \tag{19}$$

[Figure 2 about here]

If startup costs are relatively high compared to fuel cost, the metric M_i could be complemented with a term accounting for startup costs. If turned on, a power plant needs to be online for at least the minimum up time MUT_i. Based on an average startup cost, and assuming a certain generation output G_i, the term accounting for startup costs in the metric could be (with *nt* the number of elements in set *T*):

$$MS_i = \frac{\left(\sum_{t=1}^{nt} SC_{i,t}\right)/nt}{MUT_i \cdot G_i}, \quad \forall i \in I \tag{20}$$

This term MS_i would have to be added to M_i as defined above in (18). Because of the modest startup costs in the considered data, this has not been applied in the present model. According to the chosen metric, power plants are now ranked (indexed *r*) with a basic sorting algorithm. This is the priority list.

3.2 Set up lower and upper bound

According to the ranking established in the previous step, a series of power intervals is determined, based on the power plant's minimum and maximum output. Let *r* be the index of the ranked power plants (the first one is the cheapest one, the last one the most expensive). If all plants up to the r^{th} plant would be activated in a certain hour, their operating range is $[L_r, U_r]$, with

$$L_r = \sum_{n=1}^{r} Pmin_n, \quad \forall r = 1 \dots ni \tag{21}$$

$$U_r = \sum_{n=1}^{r} Pmax_n, \quad \forall r = 1 \dots ni \tag{22}$$

L is the cumulative sum of the minimum power output (according to the stacked list), *U* presents the cumulative sum of the maximum power output and *ni* is the number of elements in set *I*. The intervals determined by these two vectors will be used in the following step of the algorithm, to determine appropriate power plant activation levels.

3.3 Activation of the power plants

For every hour, the interval $[L_r, U_r]$ with lowest index *r* is determined, which comprises both the demand minus the reserves (accounting for downward reserves requirement) and the sum of the demand and reserves (accounting for the upward reserves requirement) during that hour. I.e., the minimum $\underline{r}$ for which the following two equations hold is to be found:

$$L_r \leq D_j - R_j, \quad \forall j \in J \tag{23}$$

$$D_j + R_j \leq U_r, \quad \forall j \in J \tag{24}$$

The power plants up to this level $\underline{r}$ are activated. This way the minimum number of power plants that is required, is activated each hour.

Note that in sufficiently large systems, with a sufficiently high demand, such interval $[L_r, U_r]$ should normally be found. If demand levels drop, however, it could be that no feasible interval exists. Hence, to deal with low load systems, this algorithm is extended as follows. If no interval is found, the last interval *r* with U_r still lower than the demand and reserve is started from, with these *r* plants being activated. According to the stacking order, the algorithm now tries to identify the first next plant that can be activated, ensuring that if this plant is activated, both the demand minus reserves and the sum of demand and reserve are situated between the summed minimum and maximum output of all activated plants. Hence, this plant will not be the plant on position $r+1$, because initially no feasible interval was found. If no single plant fits this search requirement, this algorithm is iteratively repeated, each time identifying and activating a plant in a similar way.

3.4 Correction of minimum up and down time violations

The corrections for minimum up and down time are carried out in three different steps. These are subsequently discussed below. Before each step, the actual up times *ut(i,j)* and down times *dt(i,j)* of all power plants are calculated and updated[2].

Step 1. If a power plant is online for a number of periods lower than a certain factor *FMU* multiplied by the plant's minimum up time *MUT*, it is shut down[3].

Step 2. A correction for minimum down times is carried out. If a plant is offline for a number of time periods lower than the minimum down time, the plant is brought online in these periods if this is feasible regarding the minimum operating point. I.e., demand (minus reserve requirement) in these periods must be higher than the sum of the minimum operating points (lower bound) of the activated power plants (comprising the considered plant). If this is not the case (lower demand), the plant is not activated but shut down in additional hours (towards end), to respect the minimum down time *MDT*.

Step 3. A correction for minimum up times is carried out. If a plant is online for a number of time periods lower than the minimum up time, the plant is brought online in additional periods, if this is feasible both regarding the power system's cumulative minimum power (lower bound) and the new down times. In a first try, the activation of the power plant is extended in a symmetric way (i.e., both in hours before and after the originally activated periods). If this is not possible (due to violation of the restrictions above), a second try is undertaken to extend the activation towards the end, or, if still not feasible, towards the front. If none of these work out, the plant is shut down in the periods where the initial up time was too low.

[Figure 3 about here]

---

[2] A straightforward algorithm is implemented towards this end. This algorithm basically counts the number of periods that a plant is on- or offline. If plant *i* is online, the variable *ut(i,j)* is equal to the number of periods the plant is subsequently online and the variable *dt(i,j)* is zero. E.g., if a plant is online for 5 hours, the variables *ut(i,j)* during these periods all have a value of 5. If a plant *i* is offline during period *j*, the variable *dt(i,j)* presents the number of hours the plant is offline, while *ut(i,j)* is zero.

[3] *FMU* must lie between 0 and 1, and is determined based on an iterative optimization (by varying this parameter, and comparing the final objective value). This parameter cannot be too high (too many power plants will be shut down in this step) nor too low (it will have no effect). After an iterative optimization, this factor *FMU* is set at 0.3.

A schematic flow diagram of these different steps is presented in Figure 3. After going through all three steps, all up and down times of the plants are feasible and the cumulative minimum operating point (lower bound) is respected. It could, however, be that during certain hours not enough plants are committed to satisfy the demand and reserve requirement. The next step of the overall algorithm (explained in Section 3.5) will take care of this.

3.5 Activation of additional power plants if needed

After the previous step where plants have been potentially shut down, it could occur that during certain hours not enough power plants are online to meet the demand and reserve requirement. Therefore, in this step, additional power plants are brought online in these cases.

The hours with a shortage in activated capacity are identified. In these hours, the algorithm tries to commit additional power plants according to the priority list, ensuring feasibility regarding the minimum operating points and up and down times. If a power plant is brought online, it needs to be on for at least its *MUT*, while the down times that are also affected, need to stay feasible as well. Furthermore, during the additional activated periods, the cumulative minimum operating point (lower bound) needs to be respected. The algorithm basically subsequently (according to priority list) checks for all plants that are off whether they can be turned on given the restrictions above. If a plant is found, it is activated. The hours with a shortage in activated capacity are updated and the algorithm is repeated. This way, enough power is committed in each hour (e.g., it could be that in certain hour multiple power plants need to be activated).

If during a shortage hour no plant can be activated, the last activated plant (according to the priority list) is shut down, i.e., it is shut down for *MDT* periods[4], and if the affected up times are lower than the *MUT*, it is shut down in these periods as well. The search is then repeated for that hour, with the plant that has been shut down now not being allowed to turn on again.

3.6 Shutting down excess power

[4] If there is an adjacent down period, it is only turned off in this specific hour, as feasibility regarding *MDT* is then automatically ensured.

In this step, two algorithms are run to turn off power plants that are not required if this leads to a better solution. Such a shutdown can only take place if the new solution still respects the *MUT* and *MDT* constraints.

Shut down power plant over entire activated range

This algorithm runs over all the power plants, starting with the most expensive plant (last one), then moving up in the priority list up to the first plant. For every plant, it is checked whether there are time periods where the plant is scheduled on, and whether during these hours, the other activated power plants are able to meet the demand and reserve requirement:

$$\forall i \in I, \text{find } j \in J \text{ that satisfies:}$$

$$z(i,j) = 1 \;\; and \;\; \sum_{n=1}^{ni} Pmax_n \cdot z(n,j) - Pmax_i \geq D_j + R_j \tag{25}$$

If a set of subsequent time periods *j* is found, and if it is bounded by *z* equal to zero on both sides (right before and after the interval), the plant is shut down during these hours, if the overall cost of this new solution is lower than the original one (for cost determination, see next step). This way, no violations against the *MUT* and/or *MDT* constraint can be triggered. A plant can only be turned off during a previously entire set of activated periods, bounded by zeros.

Shut down power plant during specific selected hours

In contrast to the previous algorithm running over plants, this algorithm runs over the time periods. A backward loop (starting at the last period running towards the first one) is executed first, while the entire algorithm is performed again in a forward loop afterwards (starting at the first period running up to the last one). For every time period, the algorithm identifies power plants that meet the following conditions: the plant is on; the plant is off in the following period (backward loop) or off in the previous period (forward loop); the rest of the activated power plants is able to meet the demand and reserve constraint; the up time is strictly larger than the minimum up time. Mathematically, this results in:

$$\forall j \in J, \text{ find } i \in I \text{ that satisfies:}$$

$$z(i,j) = 1 \tag{26}$$

$$z(i,j+1) = 0 \text{ (backward loop) } \; or \; z(i,j-1) = 0 \text{ (forward loop)} \tag{27}$$

$$\sum_{n=1}^{ni} Pmax_n \cdot z(n,j) - Pmax_i \geq D_j + R_j \tag{28}$$

$$ut(i,j) > MUT_i \tag{29}$$

If a power plant i is found that satisfies these constraints, the fuel costs during the considered hour j are calculated for the case with this plant on and for the case where the plant is turned off (in this case a term accounting for a startup cost increase needs to be added if a cold start is triggered instead of hot start, because of this shut down). If the latter is cheaper, the plant is turned off during this specific hour. If it is not beneficial to turn the plant off, it stays on.

If during a specific hour multiple plants are found respecting the constraints listed above, the cost differences are calculated for a shutdown of all of these plants individually. The shutdown of the power plant that yields the highest cost reduction (if any) is retained, while the algorithm is executed again for this same hour, as it might be beneficial to turn off more than one plant.

This algorithm cannot trigger violations regarding minimum up and down times. The variable z is only adjusted from 1 to zero if it is adjacent to at least another zero (thereby only extending a previously feasible down time) and if the up time was strictly larger than the minimum one.

3.7 Dispatching of activated power plants and calculating the cost

In this final step the actual power generation level of each plant is determined, together with the fuel and startup costs. For every hour, the plants are dispatched as follows. First, the generation of all activated power plants ($z = 1$) is set at their minimum generation output *Pmin*. Second, the marginal costs of the linearized segments (see above) of all the activated power plants are put in a single list and ranked. Given convex cost functions, this ranking is used to subsequently fill up these segments, until overall generation is equal to the demand. Afterwards, the actual fuel cost is calculated with the original quadratic cost functions, and startup costs are determined (based on the down times).

4 Data description

The input data are presented in Table 1 [13]. A ten-unit generation system is considered, which will also serve as a basis to construct larger systems. These data have been used extensively in the literature to test a wide set of algorithms.

[Table 1 about here]

Inist presents the initial state of the power plant, previous to the time frame considered in the optimization. A positive number indicates an up time, while a negative number indicates a down time. The startup cost is modeled as a stepwise cost function with two steps, i.e., a cold (*cc*) and a hot start (*hc*). If a power plant is offline for more than the sum of the parameter *tcold* and the minimum down time (*MDT*), a cold startup is required, with a cost equal to *cc*. If the down time is shorter or equal to this sum, a hot start can take place (*hc*). This results in:

$$SC_{i,t} = hc_i, \ \ \forall\, i \in I, \forall\, t = 1 \dots tcold_i + MDT_i \tag{30}$$

$$SC_{i,t} = cc_i, \ \ \forall\, i \in I, \forall\, t > tcold_i + MDT_i \tag{31}$$

A 24-hour time frame is considered. The demand is presented in Table 2, , while the reserve requirement is equal to 10% of the hourly demand [13].

[Table 2 about here]

5 Numerical simulation

The EPL algorithm is implemented entirely in Matlab [14]. The MILP model (formulation partly based on the model described in [4]) is implemented in Matlab [14] and GAMS [15], and uses the Cplex 12.2 solver [16]. Simulations are run on an Intel® Core(TM) i7-2620M CPU @2.7GHz computer with 8 Gb of RAM. In a first step, a reference (benchmark) case is considered, with the data as presented in Section 4. In a second case, the reference case is adjusted towards a system facing low residual demand. The EPL and the MILP method are used on this setting and results are compared.

5.1 Benchmark case

The system as described in Section 4 is used, in a 10, 20, 40, 60, 80 and 100 unit- setting. Each system is composed of the appropriate number of copies of the original 10 unit system[5], with the demand profile scaled accordingly. This test system, originally set up in [13], has been used extensively in the literature, to test a wide range of algorithms, see, e.g., [4, 7, 8, 10, 13, 17-29]. In this benchmark case, only upward reserves are considered (consistent with the literature). As stated previously, the reserve requirement is set to 10% of the demand.

The results of the EPL method and the MILP model are presented in Table 3. The MILP model is run with an optimality gap[6] of zero (effectively proven optimality) for the systems of ten up to 40 units[7], and with an optimality gap of 0.5% for all systems (10 up to 100 units). The calculation times of both models are also presented in Table 3. It can be seen that the EPL has very low calculation times for all systems, while these of the MILP in 0.5% optimality setting stay within reasonable limits as well. If strict optimality is required (zero optimality gap), MILP faces higher calculation times, reaching the time limit of 3600 s with a system of 60 power plants.

[Table 3 about here]

As can be seen from Table 3, the EPL method turns out to be highly effective. It is fast while it provides solutions which are very close to optimal.

5.2 Low residual load case

Especially in the framework of high RES penetration, low residual demand (as seen by the dispatchable power plants) might occur more frequently. The EPL algorithm is specifically designed

---

[5] The original system is identically duplicated (with the same power plant characteristics).

[6] MILP solvers can terminate when a feasible solution is found which lies within a predefined range (i.e., the optimality gap) of a current best relaxed bound. This solution is then proven to be within this range close to optimal.

[7] For the 60 power plant system or higher, the optimal solution could not be found within the imposed time limit of 3600 sec.

to deal with low load and puts a high focus on technical feasibility in this regard (e.g., minimum operating constraints).

The demand level from the reference case is adapted to a low demand pattern, denoted *LD*. The demand is modified by subtracting a certain term. This term could represent a wind front passing by in combination with an amount of other intermittent RES like solar PV. This term is made very generic and based on a sine function, being zero at the beginning and end of the considered time interval, reaching a peak in the middle between.

$$LD_j = D_j - FD \cdot \underline{D} \cdot \sin\left(\frac{(j-1)\cdot\pi}{nj-1}\right), \quad \forall j \in J \tag{32}$$

FD is a weigh factor, $\underline{D}$ is the minimum value of the demand D_j, and *nj* is the number of time periods considered. Next to the 1 day profile, also a 5 day setting is applied. This is constructed as 5 subsequent copies of the 1 day profile. In this case, Eq. (32) is applied on the 5 days as a whole (i.e., *nj* in this case equal to 120). Compared to the benchmark case, downward reserves (Eq. (15)) are now also included (10% of demand).

FD ranges from zero (effectively resulting in the original demand pattern) up to 1.5 in the single day (24h) case and from zero up to 1 in the 5 day (120 h) case. As an illustration, the demand *LD* is presented in Figure 4 and Figure 5 below, for the 1 and 5 day case respectively. Demand is shown for different values of *FD*. The generic RES profile is also shown for *FD = 1*. As can be seen, the minimum demand is close to zero when *FD* is at its highest values.

[Figure 4 about here]

[Figure 5 about here]

Both the EPL algorithm and the MILP model (ran with an optimality gap of 0.5%) are now applied on these low load cases. The results are summarized in Table 4, presenting the MILP calculation times, and the relative differences between MILP and EPL, for the different power systems (10 to 100 units), for a demand with FD ranging from 0 (which is effectively the reference case as discussed in the previous section) up to 1.5, in steps of 0.5. Results for the one day demand and the 5 day demand case are presented.

One day demand pattern

First the one day demand pattern is considered. A first important thing to note from Table 4 is the steep increase of computation times of the MILP model towards lower load (higher *FD*). This increase is even higher than the increase of moving towards larger systems[8]. This clearly demonstrates the difficulty MILP models have in solving low load problems. In high or medium load situations, the large (mostly least-flexible) power plants are typically used in base-load mode, meaning they are committed continuously. The optimization of the commitment of more expensive power plants constitutes the actual computational effort. These are, however, typically the smaller and the more flexible plants, with lower (absolute) minimum operating points and minimum up and down times. At low demand, on the other hand, only few plants are required, moving the optimization towards these larger and less flexible cheaper plants, making it more difficult to determine feasible/optimal solutions.

The EPL algorithm is in this case run with 3 different values for the cost metric (G_i in M_i), i.e., at the beginning, in the middle and at the end of the fuel cost curve. The best solution of the three simulations is each time retained[9]. The calculation time of a single simulation stays overall well below 0.1 s, meaning that this way of optimization basically takes no more than 0.3 s . Hence, compared to MILP, the EPL method has drastically lower computation times (for the sake of brevity, these are not displayed in Table 4).

Table 4 also presents the relative differences in outcome (total cost) between the two algorithms. As can be seen from these results, the performance of the EPL slightly decreases towards lower load. The higher relative deviations at lower load are at least partly explained by the fact that the same absolute

---

[8] The computation time of MILP problems typically scales to some extent exponentially with problem size (number of binary variables).

[9] The reason for this way of optimization is to use/test different stacking orders. Dependent on the stacking order, the marginal plant(s) might be running more on minimum operating point ($G_i = Pmin_i$), in the middle of their operating range, or at full load. Hence, dependent on the load profile, it might be worth testing different metrics, if the fuel cost curves of the different power plants lie close together or intersect.

difference simply results in a higher relative difference if the system is smaller or if the load is lower (which have a lower absolute solution). Overall, results are close to optimal. Furthermore, the EPL method found a solution in all cases, while the MILP did not find a feasible solution in 2 cases within the provided time (3600 s).

When having a detailed look at the solutions provided by the two algorithms, part of the differences originate from the fact that the EPL algorithm uses a single ranking throughout the entire simulation horizon. During specific moments of variable and low load, however, it might be beneficial to commit more expensive plants which have a lower minimum up time (e.g., peak plants (such as plant 8), which can be quickly turned on and off, although with a higher variable cost). Nevertheless, the performance of the EPL is satisfactory.

Five day demand pattern

The results for the 5 day case are also presented in Table 4. In 3 cases, the MILP did not converge to a feasible solution, while the EPL found a feasible solution in all cases. Furthermore, in a high number of cases, the MILP was restricted by the computation time limit of 3600 seconds. The current best values provided in these cases are possibly still far away from optimal, as the EPL method in some cases provides values which are significantly lower (up to almost 18%). The EPL method again proves to be very accurate, with differences mostly below 1%. The EPL stays overall below 1 s (not presented in the table), while the MILP often encounters the 3600 s time limit.

[Table 4 about here]

6 Conclusion

UC models are required for optimizing the activation levels of power plants over time. A wide range of algorithms have been developed in the past to address this issue, with MILP currently being the preferential method. However, in various power systems across the globe, RES start to play an increasingly important role. In this regard, market models (relying on a UC algorithm) are required, effectively dealing with systems with low residual demand (e.g., for planning issues or policy evaluation). This paper has demonstrated that MILP is not well suited for this setting (i.e., to deal with low residual demand). Hence, a new UC method is developed specifically to deal with low load situations. The method is based on a priority list which is used in a heuristic algorithm to come to a feasible solution, which is then potentially improved in further steps. Throughout the algorithm, specific focus is on feasibility towards the power plants' minimum operating points and minimum up and down times (important for a low load setting).

The developed EPL model is first used on a benchmark case which has been widely used in the literature and compared to MILP. The EPL turns out to be both very accurate and fast.

In a second step, the developed EPL model is employed in a low demand setting (which is derived from the reference case), and compared to the MILP model in this same setting. The computational difficulties of MILP are demonstrated this way: MILP computation times increase heavily as the residual demand is being reduced, rendering this method (used as such) less suitable in this context. The EPL method on the other hand remains highly effective, both in outcome and especially in calculation time.

The EPL is built up as a 'once-through' algorithm (consecutive steps, no iteration), which makes it fast, and to some extent modular and adjustable for additional constraints. An electric network might be incorporated by using the developed EPL as optimizer within an overall iterative algorithm (adjusting generation in different nodes to respect network constraints). An extension towards further optimization could be to combine the EPL with MILP, where the EPL would first identify a feasible good solution, which can then serve in a second step as starting point for the MILP optimization (especially relevant in cases where the MILP did not converge to any feasible solution).

Acknowledgement

E. Delarue is a post-doctoral research fellow of the Research Foundation – Flanders (FWO). The authors thank the anonymous reviewers for their useful comments and suggestions.

References

[1] M. Shahidehpour, H. Yamin, Z. Li, Market operations in electric power systems: forecasting, scheduling, and risk management, Wiley-Interscience, New York, 2002.

[2] E. Delarue, P. Van Den Bosch, W. D'haeseleer, Effect of the accuracy of price forecasting on profit in a Price Based Unit Commitment, Electric Power Systems Research, 80 (2010) 1306-1313.

[3] N.P. Padhy, Unit commitment - A bibliographical survey, IEEE Transactions on Power Systems, 19 (2004) 1196-1205.

[4] M. Carrion, J.M. Arroyo, A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem, IEEE Transactions on Power Systems, 21 (2006) 1371-1378.

[5] B.F. Hobbs, M.H. Rothkopf, R.P. O'Neill, H.-p. Chao, The next generation of electric power unit commitment models, Kluwer Academic Publishers, Boston, 2001.

[6] V.N. Dieu, W. Ongsakul, Ramp rate constrained unit commitment by improved priority list and augmented Lagrange Hopfield network, Electric Power Systems Research, 78 (2008) 291-301.

[7] T. Senjyu, T. Miyagi, A.Y. Saber, N. Urasaki, T. Funabashi, Emerging solution of large-scale unit commitment problem by Stochastic Priority List, Electric Power Systems Research, 76 (2006) 283-292.

[8] T. Senjyu, K. Shimabukuro, K. Uezato, T. Funabashi, A fast technique for unit commitment problem by extended priority list, IEEE Transactions on Power Systems, 18 (2003) 882-888.

[9] K.R. Voorspools, W.D. D'haeseleer, Long-term Unit Commitment optimisation for large power systems: unit decommitment versus advanced priority listing, Applied Energy, 76 (2003) 157-167.

[10] H. Siahkali, M. Vakilian, Electricity generation scheduling with large-scale wind farms using particle swarm optimization, Electric Power Systems Research, 79 (2009) 826-836.

[11] H. Siahkali, M. Vakilian, Stochastic unit commitment of wind farms integrated in power system, Electric Power Systems Research, 80 (2010) 1006-1017.

[12] A. Wood, B. Wollenberg, Power generation, operation and control, 2nd ed., Wiley, New York, 1996.

[13] S.A. Kazarlis, A.G. Bakirtzis, V. Petridis, A genetic algorithm solution to the unit commitment problem, IEEE Transactions on Power Systems, 11 (1996) 83-92.

[14] The Mathworks, Matlab http://www.mathworks.com/products/matlab/, (2013).

[15] General Algebraic Modelling System, GAMS http://www.gams.com/, (2013).

[16] IBM, Cplex Solver http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/, (2013).

[17] M. Afkousi-Paqaleh, M. Rashidinejad, M. Pourakbari-Kasmaei, An implementation of harmony search algorithm to unit commitment problem, Electr Eng, 92 (2010) 215-225.

[18] K. Chandrasekaran, S. Hemamalini, S.P. Simon, N.P. Padhy, Thermal unit commitment using binary/real coded artificial bee colony algorithm, Electric Power Systems Research, 84 (2012) 109-119.

[19] C. Chuan-Ping, L. Chih-Wen, L. Chun-Chang, Unit commitment by Lagrangian relaxation and genetic algorithms, IEEE Transactions on Power Systems, 15 (2000) 707-714.

[20] I.G. Damousis, A.G. Bakirtzis, P.S. Dokopoulos, A solution to the unit-commitment problem using integer-coded genetic algorithm, IEEE Transactions on Power Systems, 19 (2004) 1165-1172.

[21] J. Ebrahimi, S.H. Hosseinian, G.B. Gharehpetian, Unit Commitment Problem Solution Using Shuffled Frog Leaping Algorithm, IEEE Transactions on Power Systems, 26 (2011) 573-581.

[22] M. Eslamian, S.H. Hosseinian, B. Vahidi, Bacterial Foraging-Based Solution to the Unit-Commitment Problem, IEEE Transactions on Power Systems, 24 (2009) 1478-1488.

[23] K.A. Juste, H. Kita, E. Tanaka, J. Hasegawa, An evolutionary programming solution to the unit commitment problem, IEEE Transactions on Power Systems, 14 (1999) 1452-1459.

[24] S.S. Kumar, V. Palanisamy, A dynamic programming based fast computation Hopfield neural network for unit commitment and economic dispatch, Electric Power Systems Research, 77 (2007) 917-925.

[25] W. Ongsakul, N. Petcharaks, Unit commitment by enhanced adaptive Lagrangian relaxation, IEEE Transactions on Power Systems, 19 (2004) 620-628.

[26] L. Sun, Y. Zhang, C. Jiang, A matrix real-coded genetic algorithm to the unit commitment problem, Electric Power Systems Research, 76 (2006) 716-728.

[27] A. Viana, J.P. de Sousa, M.A. Matos, Fast solutions for UC problems by a new metaheuristic approach, Electric Power Systems Research, 78 (2008) 1385-1395.

[28] T.A.A. Victoire, A.E. Jeyakumar, Unit commitment by a tabu-search-based hybrid-optimisation technique, Generation, Transmission and Distribution, IEE Proceedings-, 152 (2005) 563-574.

[29] J. Yun-Won, P. Jong-Bae, J. Se-Hwan, K.Y. Lee, A New Quantum-Inspired Binary PSO: Application to Unit Commitment Problems for Power Systems, IEEE Transactions on Power Systems, 25 (2010) 1486-1495.

Tables

Table 1. Power system characteristics.

	Pmax [MW]	Pmin [MW]	MUT [h]	MDT [h]	inist [h]	a [$/h]	b [$/MWh]	c [$/MWh²]	hc [$/h]	cc [$/h]	tcold [h]
plant 1	455	150	8	8	8	1000	16.19	0.00048	4500	9000	5
plant 2	455	150	8	8	8	970	17.26	0.00031	5000	10000	5
plant 3	130	20	5	5	-5	700	16.60	0.00200	550	1100	4
plant 4	130	20	5	5	-5	680	16.50	0.00211	560	1120	4
plant 5	162	25	6	6	-6	450	19.70	0.00398	900	1800	4
plant 6	80	20	3	3	-3	370	22.26	0.00712	170	340	2
plant 7	85	25	3	3	-3	480	27.74	0.00079	260	520	2
plant 8	55	10	1	1	-1	660	25.92	0.00413	30	60	0
plant 9	55	10	1	1	-1	665	27.27	0.00222	30	60	0
plant 10	55	10	1	1	-1	670	27.79	0.00173	30	60	0

Table 2. Hourly electricity demand.

hour	[h]	1	2	3	4	5	6	7	8	9	10	11	12
demand	[MW]	700	750	850	950	1000	1100	1150	1200	1300	1400	1450	1500
hour	[h]	13	14	15	16	17	18	19	20	21	22	23	24

demand	[MW]	1400	1300	1200	1050	1000	1100	1200	1400	1300	1100	900	800

Table 3. Objective and computation time of the developed EPL algorithm and the MILP model (with optimality gap equal to zero and to 0.5%).

# units	total cost [$]			computation time [s]		
	EPL	MILP opt gap = 0	MILP opt gap = 0.5%	EPL	MILP opt gap = 0	MILP opt gap = 0.5%
10	563,977	563,938	564,672	0.01	1.2	1.0
20	1,124,481	1,123,308	1,125,721	0.01	5.9	2.2
40	2,246,926	2,242,609	2,246,243	0.02	3139	3.6
60	3,366,240		3,367,262	0.03		5.1
80	4,489,342		4,488,560	0.04		7.0
100	5,609,109		5,609,210	0.06		7.6

Table 4. Relative difference [%] between EPL and MILP objective, together with computation time of MILP, for the 1 day (FD 0 – 1.5) and 5 day (FD 0 - 1) low demand simulations.

	MILP computation time [s]					Relative difference EPL - MILP			
	1 day demand pattern					1 day demand pattern			
# units	FD 0	FD 0.5	FD 1	FD 1.5		FD 0	FD 0.5	FD 1	FD 1.5
10	1.1	1.2	1.2	8.0		-0.12%	0.45%	0.15%	2.00%
20	2.0	11.5	9.9	3600.0*		-0.11%	0.69%	0.26%	1.70%*
40	3.5	21.7	781.1	3600.0*		0.03%	0.53%	0.43%	1.83%*
60	5.1	177.2	3549.1	-		-0.03%	0.44%	0.46%	-
80	7.0	75.7	1133.6	-		0.02%	0.52%	0.48%	-
100	7.7	101.3	230.0	3600.0*		0.00%	0.46%	0.45%	1.69%*
	5 day demand pattern					5 day demand pattern			
# units	FD 0	FD 0.5	FD 1			FD 0	FD 0.5	FD 1	
10	3.7	9.0	3600.0*			-0.10%	0.31%	1.85%*	
20	6.6	1944.7	-			0.01%	0.29%	-	
40	41.3	-	3600.0*			0.24%	-	0.58%*	
60	860.2	-	3600.0*			0.06%	-	0.75%*	
80	3273.7	3600.0*	3600.0*			0.22%	-17.91%*	0.25%*	
100	2616.5	3600.0*	3600.0*			0.32%	-17.96%*	0.60%*	

The cases marked by a * indicate that the MILP was bounded by the imposed computation time limit of 3600 seconds. In these cases, the current best solution is provided, but this solution is not guaranteed to lie within the optimality gap of 0.5%. In 5 cases no solution was found by the MIP solver within the provided time (3600). The EPL method found a solution in all cases. A positive relative difference indicates a better solution by MILP, a negative value indicates a better solution by EPL. Difference with the benchmark case (Section 5.1) can occur (1 day demand pattern, *FD = 0*) as downward reserves are now included, and the EPL is now run for three different values of G_i retaining only the best solution.

Figure captions and Figures

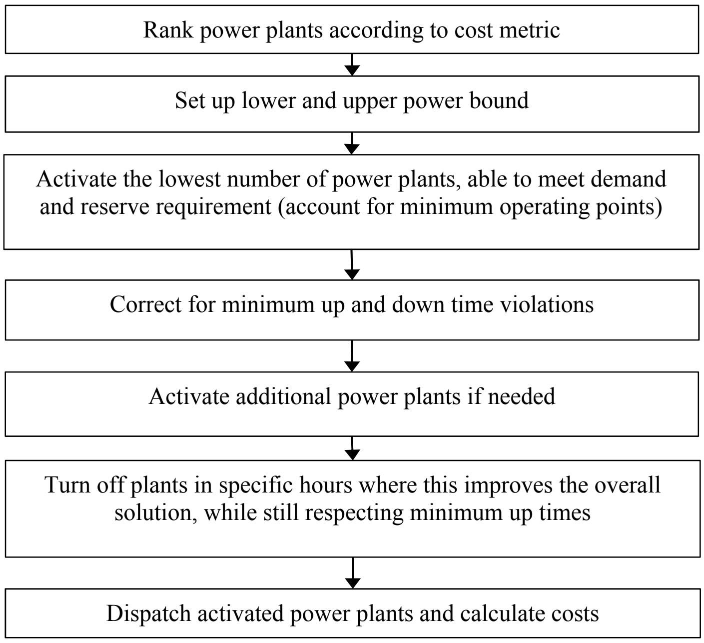

Figure 1. Flow chart of the newly developed EPL algorithm.

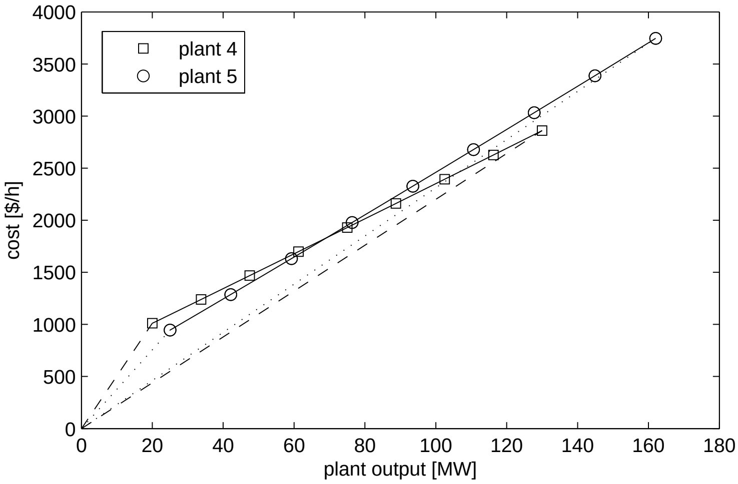

Figure 2. Power plant fuel cost over operating range, presented for two different power plants.

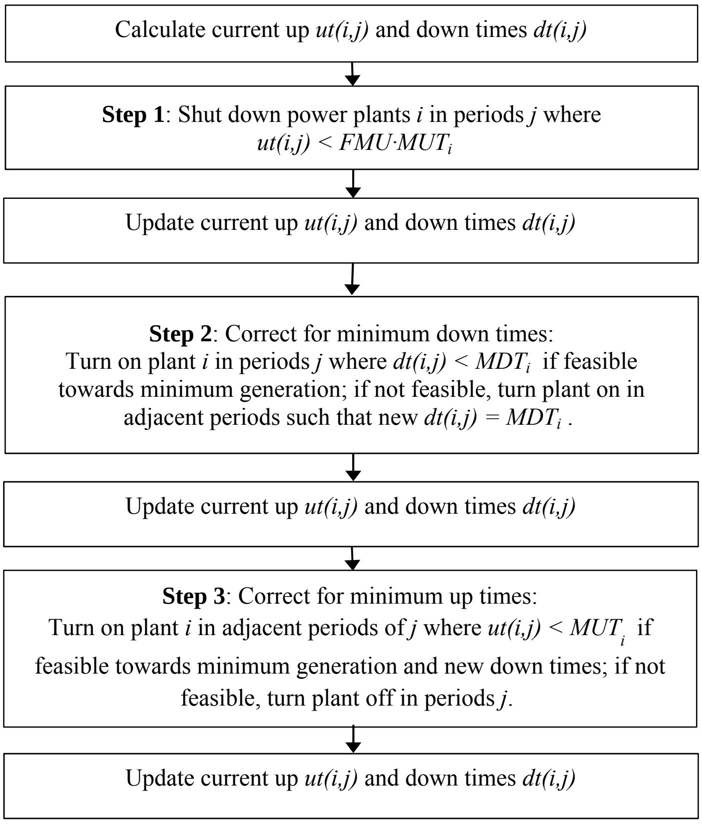

Figure 3. Flow chart of the correction in activation levels to respect minimum up and down times.

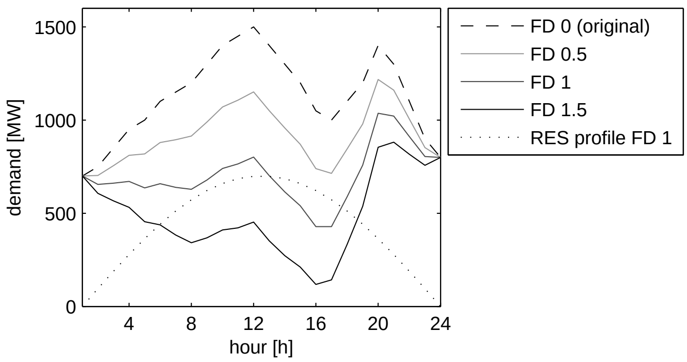

Figure 4. Different demand profiles and RES profile, in the 1 day setting.

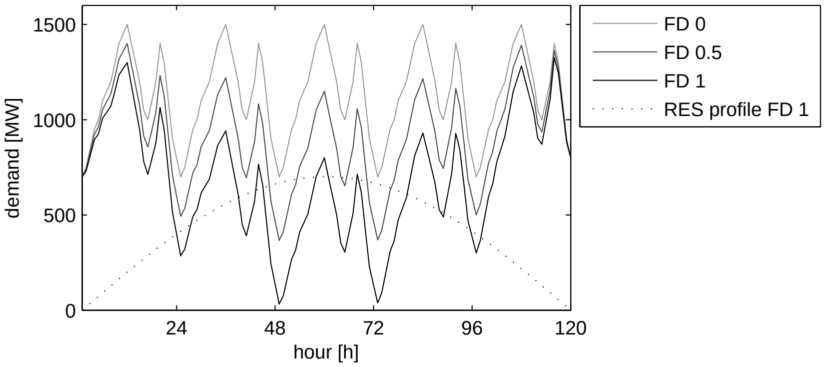

Figure 5. Different demand profiles and RES profile, in the 5 day setting.