Energy-aware multi-mode resource constrained project scheduling under time dependent electricity costs and user compromised consumption

F. Torrent-Fontbona[a,*], B. López[a]

[a] *eXiT research group, Institute of Informatics and Applications, University of Girona, Campus Montilivi, Building P4, 17071 Girona*

Abstract

Future smart grid involves the monitoring and control of the energy consumption profile of each consumer with demand-side strategies aiming to incentivise changes in consumers' profiles like time-dependent energy prices and compromised load patterns. However, demand-side management strategies need consumers capable to respond to the incentives. As a consequence, the project scheduling problem, which consists of a set of activities that has to be scheduled subject to precedence and resource constraints, need to be reviewed to consider the new challenges posed by smart grids. In this paper we model the multi-mode project scheduling problem under time-dependent energy prices and compromised load patterns to support decision making with energy-aware issues. We carried out experimentation based on real-based simulated scenarios to show the consequences of the proposed model.

Keywords: Project scheduling, resource allocation, energy optimisation, time-dependent prices, load profile, problem model

*Corresponding author

Email addresses: `ferran.torrent@udg.edu` (F. Torrent-Fontbona), `beatriz.lopez@udg.edu` (B. López)

URL: `http://eia.udg.es/~ftorrent` (F. Torrent-Fontbona), `http://eia.udg.es/~blopez` (B. López)

1. Introduction

The future smart grid provides a new scenario in which demand for electricity could be made more adaptive to supply conditions, in what is known as demand-side management. Demand-side management involves utility strategies that influence the end use of energy according to the desired changes in the pattern and magnitude of an energy load, known as load shape [12]. One of the strategies for demand-side handling is to promote time-dependent rates, thus energy demand peaks can be softened as a consequence of users shifting their consumption to the energy cheapest hours.

However, it is not only a case of shifting energy consumption but, due to the sustainable issues, it is important to reduce the amount of energy consumed, in what is called energy efficiency. In this regard, the ISO50001:2011 standard considers the definition of energy plans, so that companies can compromise to move from a current energy load shape to a lower one. The fulfilment of this standard regarding the consequent contribution of the companies to the reduction of emissions, will be as important as the ISO:9000 has been for quality.

These energy-related aspects would, therefore, affect the scheduling of resources in companies. On the one hand, the energy consumption of the resources in either their production or service activities should follow the load shape agreed. On the other hand, when there is some margin for scheduling the use of one resource inside a time window, companies would be more interested in using the resource on the cheapest energy hours. Nowadays scheduling tools are mainly based on the makespan and costs. But, in the following years it would become crucial to incorporate energy issues in business process management to make them adaptive to the changes the smart grids would bring about [5].

In this regard, we propose to reformulate the multiple Mode Resource Constrained Project Scheduling Problem (MRCPSP) [16] to include time-dependent energy prices in combination with compromised load patterns. MRCPSP consists of assigning resources to a set of tasks if such resources have the appropriate skill, and scheduling the execution of the tasks in order to optimise a particular objective (often the makespan and the production costs). Tasks have precedence relationships and different execution modes (different features, like execution time) defined by the resources used to perform the task. We extend MRCPSP to consider a compromised load pattern as well as time-dependent energy prices whilst minimising the makespan, the production costs and the energy consumption, which we call the energy-aware MRCPSP (e-MRCPSP).

This paper is organised as follows: in Section 2 we review some related work. Next, in Section 3, we present the problem and provide its model. In Section 4 we analyse the presented problem solving it using two approaches: an exact method and a meta-heuristic. Finally we end the paper in Section 5 with the conclusions and proposals for some future work.

2. Related work

Project scheduling [7; 18] has attracted ever growing attention in recent years. The problem has many extensions [16; 8] but, to the best of our knowledge, it has not been solved considering variable energy prices and compromised load patterns. Nevertheless, the starting point of this paper is the MRCPSP which have a large number of related works in the literature.

Additionally, there are different extensions or types of RCPSP in the literature besides the typical MRCPSP like those tackled by [30; 31; 4]. For example, [14; 20] consider uncertain costs and durations of the activities. [8] presents a formulation for a MRCPSP considering time-varying resources to model periods of time that resources (i.e. workers) are not available (i.e. holidays and weekends) and the consequences of stopping an activity. [24] consider a non-constant use of a resource throughout the execution of an activity. This paper presents another extension of the MRCPSP, which is not opposed to these previous works. The presented e-MRCPSP do not consider different modes of using a resource (like [8; 24]) but includes the consequences of using resources which cost is not constant.

The use of each type of MRCPSP depends on the use case and the associated complexity to provide a solution to an instance of the problem. The literature presents many approaches for solving MRCPSP, specially meta-heuristic methods capable to find good solutions yet not the optimal. Following this line, [1; 2; 3; 15; 25] present Genetic Algorithms (GAs) to solve the MRCPSP and make an analysis of the most suitable values of the parameters of the GAs. [19] present a particle swarm optimisation based approach. [6] present a simulated annealing algorithm with a particular cooling mechanism that improves its convergence. Furthermore, [6] compare their method with other meta-heuristic algorithms of the literature gibing a classification according their performance. However, all these methods have to be adapted (or cannot be used) to solve the e-MRCPSP. As a consequence, this paper present to basic approaches to solve the given problem, yet it is not the main contribution of the paper. The approaches are presented in order to complement the presented formulation.

Besides the great research related to project scheduling problems, public institutions are making great efforts to design and develop the future smart grid [11; 10]. Part of this effort is being translated in developing household management systems that deal with time-dependent prices [27; 23], studying consumers' behaviour when faced with these variable prices [13; 21; 23] and studying and designing new negotiation systems between electricity companies (distributors), producers and consumers [29]. However, few efforts have been put into how to adapt and solve activity scheduling problems in the context of the future smart grid.

3. The energy-aware multi mode scheduling problem

In this paper we tackle the MRCPSP under time-dependent energy prices and compromised load shapes, henceforth called e-MRCPSP. The MRCPSP is made up of a set of activities A_i, $i \in \{1, \ldots, N\}$ linked by a classical end-to-start precedence relationship, which means that an activity cannot be started before all its predecessor tasks are finished.

Following the notation of [4], each resource R_m, $m \in \{1, \ldots, M\}$, masters one or more skills among all the skills S_k, $k \in \{1, \ldots, K\}$, existing in the project. Each activity has to be performed by one resource that masters the skill(s) it requires and the resource has to be available during the execution of such activity. Resources define the mode used to carry out a particular activity, which means that the resource used to perform an activity determines the processing time $p_{i,m}$ and the resource cost $c_{i,m}$. A resource cannot be assigned to more than one activity at a time; to this end, the binary variable $z_{i,m}$ is defined and it indicates that resource m is assigned to activity i if and only if $z_{i,m} = 1$, otherwise $z_{i,m} = 0$. Variable $s_i \in [0, T_{max}]$ indicates the scheduled starting time of activity i (T_{max} is the maximum time horizon considered). In this way, an assignment of values to those variables $\mathcal{S} = \{s_0, \ldots, s_N\}$ and $\mathcal{Z} = \{z_{1,1}, \ldots, z_{i,m}, \ldots, z_{N,M}\}$ define the scheduled starting time and mode of the activities of the project. For the sake of simplicity, a schedule can be denoted as $(\mathcal{S}, \mathcal{Z})$. The MRCPSP consists of finding the values of those variables subject to activity precedence constraints, and optimising an objective function.

Regarding the objective function most works in the literature that solve the MRCPSP minimise the makespan of the schedule. In the e-MRCPSP, we want to minimise three objectives: the makespan, the energy consumption and the economical cost of the schedule. To this end, the e-MRCPSP considers also the energy consumption per unit of time $e_{i,m}$, the company energy load shape Σ, and the day-ahead tariffs Γ.

For a given schedule $(\mathcal{S}, \mathcal{Z})$, we define the energy consumption, $\rho_t(\mathcal{S}, \mathcal{Z})$, at time t as follows:

$$\rho_t(\mathcal{S}, \mathcal{Z}) = \sum_{\forall_i | s_i \leq t < s_i + \sum_{k=1}^{M} z_{i,k} p_{i,k}} \sum_{m=1}^{M} z_{i,m} e_{i,m} \tag{1}$$

which is the sum of the energy consumptions (second summation of the equation (1)) of all active tasks at time t (first summation of equation (1)). Thus, ρ_t $\forall t$ is the load profile of the schedule. This load profile has to fit a compromised load shape Σ which is loosely defined within a set of boundaries as follows:

$$\Sigma = \left\langle \underline{P_t}, \overline{P_t}, \underline{\rho_t}, \overline{\rho_t} \right\rangle_{\forall t} \tag{2}$$

where $\underline{P_t}$ and $\overline{P_t}$ are the minimum and maximum allowed energy consumption at time t (usually due to physical constraints) and $\underline{\rho_t}$ and $\overline{\rho_t}$ are the lower and upper bounds of the compromised energy load. An organisation with a compromised energy profile Σ must keep its energy consumption ρ_t within the interval $\left[\underline{P_t}, \overline{P_t}\right]$, but also it is expected to keep ρ_t in the interval $\left[\underline{\rho_t}, \overline{\rho_t}\right]$ because consuming energy out of this interval would involve some economic consequences like augmented prices or fines. We define the economic agreement (energy tariff) an organisation is subject to as:

$$\Gamma = \left\langle \pi_t, \underline{\pi_t}, \overline{\pi_t}, \underline{f_t}, \overline{f_t} \right\rangle_{\forall t} \tag{3}$$

where π_t is the time-dependent price of the energy when $\rho_t \in \left[\underline{\rho_t}, \overline{\rho_t}\right]$, $\underline{\pi_t}$ is the price when $\rho_t < \underline{\rho_t}$ and $\overline{\pi_t}$ the price when $\rho_t > \overline{\rho_t}$; $\underline{f_t}$ and $\overline{f_t}$ are fines applied when $\rho_t < \underline{\rho_t}$ and $\rho_t > \overline{\rho_t}$ respectively. Note that $\underline{f_t}$ and $\overline{f_t}$ are not per-energy-unit prices. Figure 1 shows a graphical representation of Γ and Σ.

Therefore, the three objectives of the e-MRCPSP are defined as follows:

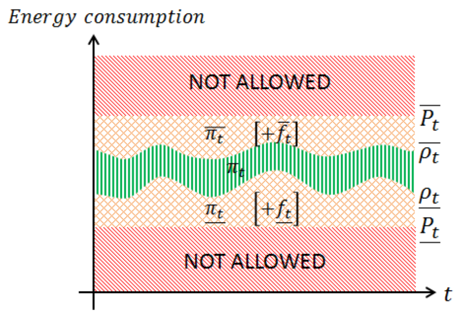

Figure 1: Load shape example. The green zone (middle zone) defines the gap $[\underline{\rho_t}, \overline{\rho_t}]$ where the energy consumption should be. Outside this gap, augmented prices, $\underline{\pi_t}$ or $\overline{\pi_t}$ and fines, $\underline{f_t}$ or $\overline{f_t}$, will be imposed by the electricity company. Red zone is the not allowed consumption due to, for example, physical features of the line.

- Makespan: the processing time of the whole project. It is defined as the maximum difference between the start of a task and the end of another.

$$C_T(\mathcal{S}, \mathcal{Z}) = \max_{i,j} \left(s_i + \sum_{m=1}^{M} z_{i,m} p_{i,m} - s_j \right), \forall i,j \; 1 \leq i,j \leq N \tag{4}$$

- Energy consumption: sum of all the energy needed to execute tasks in the scheduled modes

$$C_E(\mathcal{S}, \mathcal{Z}) = \sum_{t=1}^{T_{max}} \rho_t(\mathcal{S}, \mathcal{Z}) \tag{5}$$

- Economic cost: cost of the resources used to perform the activities plus the cost of the energy consumed.

$$C_M(\mathcal{S}, \mathcal{Z}) = \sum_{i=1}^{N} \sum_{m=1}^{M} z_{i,m} c_{i,m} + \sum_{t=1}^{T_{max}} \Phi_t(\rho_t(\mathcal{S}, \mathcal{Z}), \Sigma, \Gamma) \tag{6}$$

where

$$\Phi_t\left(\rho_t, \Sigma, \Gamma\right) = \begin{cases} \rho_t \pi_t + \left(\underline{\rho_t} - \rho_t\right) \underline{\pi_t} + \underline{f_t} & \rho_t < \underline{\rho_t} \\ \rho_t \pi_t & \underline{\rho_t} \leq \rho_t \leq \overline{\rho_t} \\ \rho_t \pi_t + \left(\rho_t - \overline{\rho_t}\right) \overline{\pi_t} + \overline{f_t} & \rho_t > \overline{\rho_t} \end{cases} \tag{7}$$

Given the three objectives to minimise, the e-MRCPSP consists of finding the scheduling (the sets $\mathcal{S}$ and $\mathcal{Z}$) that minimises a weighted sum of the makespan, the economical cost and the energy consumption. In this paper we propose the weighted sum as aggregation function, thus, the minimisation problem can be expressed as follows:

$$\min_{\mathcal{S},\mathcal{Z}} \left\{\Psi\left(\mathcal{S}, \mathcal{Z}\right)\right\} \tag{8}$$

where

$$\Psi\left(\mathcal{S}, \mathcal{Z}\right) = w_1 C_T\left(\mathcal{S}, \mathcal{Z}\right) + w_2 C_E\left(\mathcal{S}, \mathcal{Z}\right) + w_3 C_M\left(\mathcal{S}, \mathcal{Z}\right) \tag{9}$$

with $\sum_k w_k = 1$.

The complexity of the e-MRCPSP is higher than the MRCPSP, as there are two components of the minimization problem that depend on when the resources are assigned, $C_E\left(\mathcal{S}, \mathcal{Z}\right)$ and $C_M\left(\mathcal{S}, \mathcal{Z}\right)$. Complexity is experimentally analysed in Section 4.

3.1. Illustrative example

Consider that there is a project to schedule that consists of a set of activities $\{A_1, A_2, A_3\}$. All tasks need to be performed by a resource with the same skill S_1. The maximum time horizon considered is $T_{max} = 5$. Consider a discretionary interval of 1 unit.

Also consider that we have a set of resources $\{R_1, R_2\}$ where both have the skill S_1. Resources' energy consumptions are $e_{1,1} = 1$, $e_{1,2} = 5$, $e_{2,1} = 2$, $e_{2,2} = 2$, $e_{3,1} = 4$, $e_{3,2} = 2$ (kWh); the durations are $d_{1,1} = 3$, $d_{1,2} = 1$, $d_{2,1} = 2$, $d_{2,2} = 2$, $d_{3,1} = 2$, $d_{3,2} = 1$ (hours). We focus on the time-dependent prices and its related cost, $\sum_{t=1}^{T_{max}} \Phi\left(\rho_t(\mathcal{S}, \mathcal{Z}), \Sigma, \Gamma\right)$, that is, the second component of C_M. Thus, we consider the resources costs $c_{i,m}$ are negligible ($\sum_{i=1}^{N} \sum_{m=1}^{M} z_{i,m} c_{i,m} \approx 0$), and $w_1 = w_2 = 0$, and $w_3 = 1$ of $\Psi\left(\mathcal{S}, \mathcal{Z}\right)$.

Finally, consider the three different scheduling scenarios with different load shapes and electric tariffs:

Case (a):

$$\Gamma = \left\langle \begin{bmatrix}1\\2\\3\\2\\1\end{bmatrix}, \begin{bmatrix}2\\4\\6\\4\\2\end{bmatrix}, \begin{bmatrix}2\\4\\6\\4\\2\end{bmatrix}, \begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix}, \begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix} \right\rangle$$

$$\Sigma = \left\langle \begin{bmatrix}0\\0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}10\\10\\10\\10\\10\end{bmatrix}, \begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix}, \begin{bmatrix}4\\4\\4\\4\\4\end{bmatrix} \right\rangle$$

For $t = 0$, the allowed energy defined in case (a) is defined by $\underline{P_0} = 0$ and $\overline{P_0} = 10$, while the compromised energy consumption is within $\underline{\rho_0} = 1$ and $\overline{\rho_0} = 4$. That happens for all t. Regarding energy tariffs, for $t = 0$, $\pi_0 = 1$, $\underline{\pi_0} = 2$, $\overline{\pi_0} = 2$, $\underline{f_0} = 1$, and $\overline{f_0} = 1$; while for $t = 1$, $\pi_1 = 2$, $\underline{\pi_0} = 4$, $\overline{\pi_0} = 4$, $\underline{f_0} = 1$, and $\overline{f_0} = 1$.

Case (b): Same Γ as (a), but with a broader load shape Σ, as follows

$$\Sigma = \left\langle \begin{bmatrix}0\\0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}10\\10\\10\\10\\10\end{bmatrix}, \begin{bmatrix}0\\0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}5\\5\\5\\5\\5\end{bmatrix} \right\rangle$$

$\underline{\rho_t}$ has been lowered from 1 to 0, and $\overline{\rho_t}$ has been increased from 4 to 5.

Case (c): Same Σ as (a), but with a different time-dependent tariff Γ, as follows

$$\Gamma = \left\langle \begin{bmatrix}3\\2\\1\\2\\3\end{bmatrix}, \begin{bmatrix}6\\4\\2\\4\\6\end{bmatrix}, \begin{bmatrix}6\\4\\2\\4\\6\end{bmatrix}, \begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix}, \begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix} \right\rangle$$

In this case, the prices behaves in an opposite way than (a): when prices in (a) decrease, in (c) increase; and vice-versa.

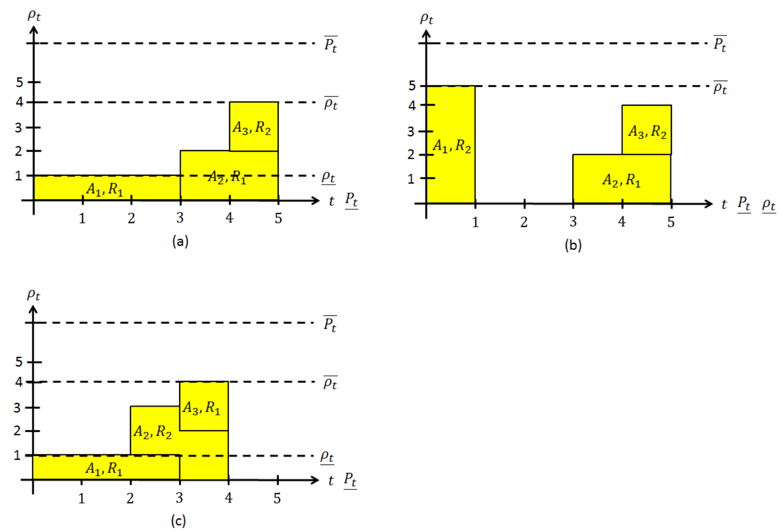

Figure 2: Examples of scheduling

The resulting optimal schedules of each case are illustrated in Figure 2. It shows how optimal schedules try to keep the energy profile in the bounds defined by Σ while trying to perform most of the tasks in the cheapest time slots. Solutions for cases (a) and (b) fulfil the compromised load shape ($\rho_t \in \left[\underline{\rho_t}, \overline{\rho_t}\right]$); The optimal solution found in case (c) does not fulfil the load shape (there is no load for $t = 5$, so the minimum required load $\underline{\rho_5}$ is not reached) because, from the economic point of view, it is cheaper breaking it down than moving one activity to the slot 4-5. Thus, we have to work out if it is worthwhile to break down some soft-constraints and to face the involved penalty.

4. Empirical study

Once we have defined the e-MRCPSP, in this section we analyse the consequences on the scheduled projects by following the proposed model. To this end, we implement two different methods to solve the modelled problem: Branch and Bound (B&B) and GA. We have chosen B&B because it is an exact optimisation method (it guarantees finding out the optimal solution to a problem if enough time is given) and it can handle any objective function or constraint. Nevertheless, when the size of the problem increases, state-of-the art solutions are based on meta-heuristics like GA (see discussion in

Section 2). Moreover, the complexity of the e-MRCPSP posed in this paper is greater than a common MRCPSP due to the time-dependent prices and the load shape. In this sense, the use of meta-heuristic algorithms like GA is justified in. Of course, other off-the-shelf methods can be also used, but the goal of this paper is not to provide a review or the best algorithmic approach but to validate the e-MRCPSP model presented.

4.1. Branch and bound

B&B is a complete optimisation method. For seeking the optimal schedule it first builds a tree-shaped graph where each depth level corresponds to an activity, each node corresponds to an activity performed by a particular resource at a particular start time (each node corresponds to a $z_{i,m}$ and s_i) and each branch corresponds to a particular schedule. Since all nodes in the same depth level correspond to the same activity, nodes that share a level are not connected between them. Furthermore, the depth levels of the tree are ordered into a decreasing order of their number of nodes, meaning that activities with less available modes and starting times make up the top levels and the activities with more possible modes and starting times make up the bottom levels. This top-down ordering ensures that the tree has the minimum number of nodes [9; 26].

Due to the size of the tree, the B&B algorithm explores it in a depth-first-search way, enabling it to keep in memory only the best branch found and the current one. When the algorithm is exploring a branch and at some point it finds out that the branch does not fit the constraints (i.e. a resource is unavailable, or the $\rho_t > \overline{P_t}$, etc.) it prunes such a branch and backtracks to the first node with an unchecked path to explore. Note that the constraint $\rho_t > \underline{P_t}$ should not be checked before reaching the leaf node because all activities are energy consuming. When the algorithm reaches a leaf node it evaluates the branch and saves it if the branch is better than the best one found. Then it backtracks to the first node with another path. Checking the constraints at each point avoids continually exploring infeasible branches. Algorithm 1 shows the exploration function of the proposed B&B. It is a recursive function that expands a branch at a time and saves it in *bestBranch* if it is the best branch found. For simplicity it refers to the task mode as a task to be executed by a particular resource at a particular start time.

Algorithm 1 BB_expand(bag)

Input: bag of activity modes (bag) grouped by tasks and ordered into a decreasing order

$b \leftarrow choose\,(bag\,[0])$ *'take a task mode from the corresponding task-level'*
$branch.add\,(b)$
$fits \leftarrow checkConstraints\,(branch)$ *'returns true if it fits.'*
$Aux \leftarrow remove\_current\_task\_bids(bag)$ *'removes all bids related to the task done by b.'*
if $fits$ **and not** leaf node **then**
    BB_expand(Aux)
else
    $value \leftarrow evaluate\,(branch)$ *'returns schedule's value and* ∞ *if it does not fit the constraints.'*
    if $value < bestValue$ **then**
        $bestBranch \leftarrow branch$
    end if
end if
$branch.remove\,(b)$ *'remove b from the current explored branch.'*

4.2. Genetic algorithms

GAs, [17; 28; 22], exploit the ability of the evolution operators to improve the quality of a population of solutions, generation after generation, in order to find the optimum solution to a given problem. GAs are widely used to solve hard optimisation problems because they are very effective tools for performing a global search and their use does not involve many mathematical assumptions (GAs can handle any objective function and constraint and there is no need of convexity in the objective function).

The GA proposed in this paper to solve the e-MRCPSP uses chromosomes which are strings of length M where each slot corresponds to an activity and each slot has the information regarding the scheduled mode (resource assigned to carry out the activity and the scheduled start time) used to perform the corresponding activity. Therefore, each chromosome represents a candidate solution $(\mathcal{S}, \mathcal{Z})$ to the problem. The GA starts computing an initial population (new random chromosomes) of size *popSize*. Then it determines the fitness values of the individuals of the initial population. The fitness function used is

$$f\,(\mathcal{S}, \mathcal{Z}) = \frac{1}{w_1 C_T\,(\mathcal{S}, \mathcal{Z}) + w_2 C_E\,(\mathcal{S}, \mathcal{Z}) + w_3 C_M\,(\mathcal{S}, \mathcal{Z})} \tag{10}$$

meaning that the higher the fitness the better. Note that we do not expect zero values in the denominator. Once the initial population is made up, the GA carries out reproduction and elitism, generation after generation, to make the population evolve and to find better solutions.

Reproduction consists of 3 main steps:

- Selection of parents. At each generation the GA selects N_c couples of parents to breed N_c couples of children. The selection of each couple is done using the 3 tournament selection rule that consists of selecting randomly 3 random chromosomes and choosing the best as the first parent of the couple. The process is repeated to select the second parent. This method tends to keep more diversity than the roulette wheel selection [17].

- Crossover. After selecting the parents, each couple of parents breeds a couple of children exchanging their *genetic information* using the 2 point crossover [17]. Thus, each child has 2 strings of information from one of the parents separated by a string of information from the other parent.

- Mutation. After each new child chromosome is created it mutates by randomly changing the execution mode of some of the activities. Particularly, it changes the execution mode by another randomly selected one with a probability of 0.01.

After new chromosomes are created and added to the population, GA uses elitism to remove the worst members of the population and maintain the population size. Algorithm 2 summarises the procedure of the explained GA. The termination criterion is based on the number of generations because we prioritised to control the search time instead of the quality of the solutions in the experimentation. However, a termination criterion based on how the best solution has improved in the last generations can be easily implemented as well as a mix of different termination criteria (number of generations, improvement in last generations, etc.).

Algorithm 2 Genetic Algorithm

Input: $N_g = 1000$, $popSize = 300$
1: $population \leftarrow initialize\_population\,(popSize)$
2: Compute the fitness of each chromosome using $f\,(\mathcal{S}, \mathcal{Z})$
3: **for** $generation = 0$ **to** N_g **do**
4: Selection: selects $\frac{popSize}{2}$ couples of parents using 3 tournament selection
5: Crossover: creates each couple of parents creates a couple of children using 2 point crossover
6: Mutation
7: Compute the fitness of each new chromosome using $f\,(\mathcal{S}, \mathcal{Z})$
8: Add new chromosomes to *population*
9: Elitism: remove the worst chromosomes from *population* keeping only the best *popSize* members
10: **end for**

4.3. Experimental setup

We conducted experimentation over simulations based on real projects[1] that a company has to schedule and perform using their own resources (7 different resources). Each resource masters a set of skills that allow it to perform activities and has a processing time, cost and energy consumption that depend on the activity to execute.

Three different scenarios are considered:

Scenario 0: Comparison of e-MRCPSP with MRCPSP. We solved a set of 80 different projects (of sizes from 4 to 9 activities) taking into account energy consumption and variable energy prices (e-MRCPSP) and considering only the makespan as in a typical MRCPSP. Results are provided in terms of makespan, energy consumption, and economic cost (according to Equations (4) to (6)), in order to be able to compare the outcomes. Moreover, the computational time is also provided.

Scenario 1: Analysis and comparison of the performance of B&B and GA. We solved the MRCPSP for different projects using B&B and GA. The sizes of the projects vary from 4 to 9 activities and we scheduled 20 projects for each size. The results are provided according to the objective function (Equation (9)) and the computational time.

Scenario 2: Analysis of the performance of GA with greater projects (with more activities). We used it to schedule different projects with sizes of 15, 20, 25, 30, 35 and 40 activities. There are up to 10 projects of size 15 to 25; while up to 5 projects of size 30 to 40. The results are measured in terms of the objective function (Equation (9)).

The experiments have been carried out in a PC with an Intel®Core™i5 @ 2.80GHz CPU, 8.00GB of RAM and Windows 7 64 bits.

4.4. Results: MRCPSP versus e-MRCPSP

Figure 3 shows the statistical information (minima, maxima and percentiles 25, 50 and 75) of the relative differences of the cost, makespan and energy consumption between MRCPSP and e-MRCPSP (scenario 0). It clearly shows that, when we consider energy consumption and the price of

[1]Experimentation data available at `http://eia.udg.es/~apla/fac_data/`

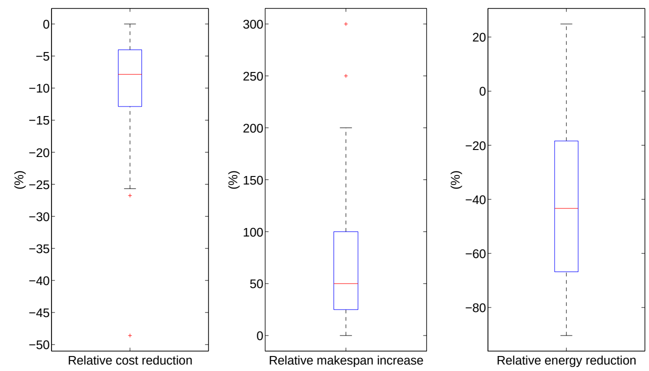

Figure 3: Relative difference in cost, makespan and energy consumption of the optimal schedules of different projects of different sizes when energy consumptions and energy prices are considered (e-MRCPSP) respect when only the makespan is considered as in a typical MRCPSP.

the energy in the problem solving process, the cost and the needed energy of the final schedule is reduced (40% in average) in exchange for increasing the makespan. On the other hand, cost is also reduced (10%). That is expected to happen in any multi-objective optimisation problem when all the objectives are aggregated in a function (see Equation (9)). However we want to highlight the importance of taking into account energy consumptions and energy prices in project scheduling problems with energy consuming activities such as the projects solved in this paper.

Another implication of taking into account variable energy prices, energy consumptions and compromised load shapes, is the complexity of the problem to solve: variable prices increase the complexity of the e-MRCPSP in respect of a typical MRCPSP. Figure 4 shows the average time elapsed by the B&B algorithm presented in this paper to solve 120 projects of sizes from 4 to 9 activities (20 projects per size). For scheduling the projects, as in a typical MRCPSP, the B&B algorithm minimised the makespan without considering energy prices and only considering energy consumption to keep the energy profile $\rho_t \in \left[\underline{P_t}, \overline{P_t}\right]$. Results show that the algorithm needs about 2 to 3 more orders of magnitude of time to solve the e-MRCPSP than to solve MRCPSP.

Due to the complexity of the e-MRCPSP, an optimal solution cannot be found within reasonable time using complete methods like B&B when the size of the project increases. A proof of it is the mean elapsed time by B&B

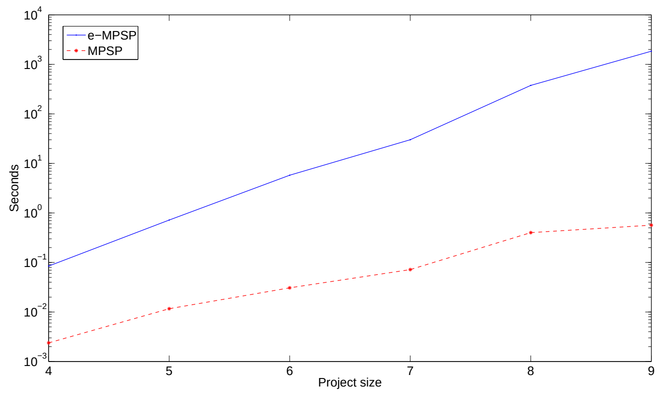

Figure 4: Mean elapsed time by B&B algorithm presented in this paper to solve e-MRCPSP and MRCPSP of different sizes.

showed in Figure 5 which exponentially rises with the number of activities and that, for a project with 9 activities, B&B needs an average of 10^3 seconds to schedule it. In this sense, the use of meta-heuristic algorithms like GA is justified. Nevertheless, GA does not guarantee the optimal solution. Figure 5 shows the relative error of the solutions found by GA. It shows that the more activities the project has, the greater the error of the solution.

4.5. Scalability and optimality analysis

Since we do not know the optimal schedules of the projects, we have solved each project 20 times with GA and we present the statistics of the solutions found for each project (average cost, standard deviation of the cost, minimum cost, maximum cost and percentage of times that GA achieved the minimum cost) on Tables 1 to 6. When dealing with projects with a particular complexity (Tables 1 to 3), GA converges in some occasions to the same minimum (see for example, projects #01 and #08 of Table 1, projects #14 and #18 of Table 1, and projects #23, #24, #25, #28, and #29 of Table 3). Therefore, we can consider that in such situations, it is very likely that the minimum found is the optimal one. In general, the standard deviation obtained in all the solutions found by the GA is around a 2%. However, when the complexity of the projects increases (Tables 5 to 6), the solutions in each GA run diverge, and then we are probably obtaining an approximate solution somehow far from the optimal.

In summary, results on these tables show that although GA is not able

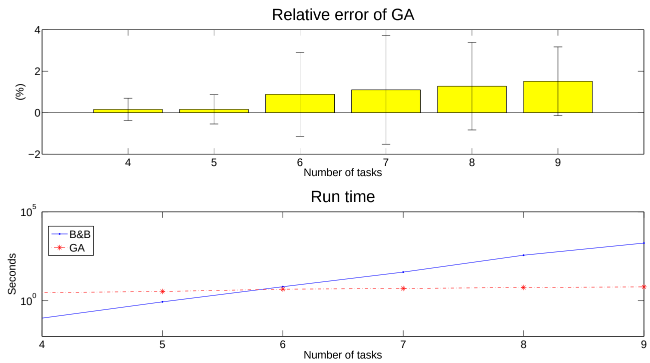

Figure 5: Scheduling results from different projects grouped by the number of activities. On the top: relative error (mean and standard deviation) of the solutions found by GA respect the optimums (find by B&B). On the bottom: elapsed time by B&B and GA.

Project ID	Mean	St. dev.	Min	Max	Percentage of minima found (%)
#01	64.28	0.00	64.28	64.28	100.00
#02	58.46	0.55	58.21	60.04	80.00
#03	63.10	0.67	62.81	65.61	80.00
#04	59.74	0.21	59.53	60.12	40.00
#05	58.08	2.72	57.35	69.74	90.00
#06	41.20	0.21	41.08	41.67	75.00
#07	63.83	0.10	63.76	63.99	65.00
#08	66.99	0.00	66.99	66.99	100.00
#09	75.83	0.49	75.67	77.84	85.00
#10	54.12	0.21	54.01	54.65	75.00

Table 1: Scheduling results using GA with projects with 15 activities. GA has been run 20 times per project.

Project ID	Mean	St. dev.	Min	Max	Percentage of minima found (%)
#11	80.61	2.03	79.25	87.29	35.00
#12	89.62	0.75	89.26	92.67	35.00
#13	74.83	0.80	74.22	77.41	30.00
#14	40.21	0.00	40.21	40.21	100.00
#15	92.20	0.28	91.89	93.37	5.00
#16	74.15	0.47	73.59	74.59	40.00
#17	81.44	0.31	81.34	82.37	90.00
#18	75.12	0.00	75.12	75.12	100.00
#19	86.80	0.00	86.80	86.80	50.00
#20	89.08	1.32	88.39	93.59	65.00

Table 2: Scheduling results using GA with projects with 20 activities. GA has been run 20 times per project.

Project ID	Mean	St. dev.	Min	Max	Percentage of minima found (%)
#21	53.13	0.20	53.02	53.55	80.00
#22	105.03	2.12	103.16	108.88	40.00
#23	87.68	0.00	87.68	87.68	100.00
#24	111.05	0.00	111.05	111.05	100.00
#25	104.57	0.00	104.57	104.57	100.00
#26	86.06	0.29	85.93	86.74	60.00
#27	95.87	0.46	95.37	96.28	45.00
#28	107.10	0.00	107.10	107.10	100.00
#29	90.34	0.00	90.34	90.34	100.00

Table 3: Scheduling results using GA with projects with 25 activities. GA has been run 20 times per project.

Project ID	Mean	St. dev.	Min	Max	Percentage of minima found (%)
#30	127.20	0.19	126.94	127.35	35.00
#31	122.11	2.09	120.88	130.20	50.00
#32	120.94	0.82	119.99	123.29	35.00
#33	113.30	0.35	113.10	114.54	40.00
#34	122.65	1.97	121.77	131.01	45.00

Table 4: Scheduling results using GA with projects with 30 activities. GA has been run 20 times per project.

Project ID	Mean	St. dev.	Min	Max	Percentage of minima found (%)
#35	150.23	2.10	147.83	156.69	30.00
#36	150.11	4.42	147.49	159.69	65.00
#37	133.87	0.92	133.66	137.86	95.00
#38	149.12	2.23	147.78	154.38	30.00
#39	169.24	6.74	164.52	191.87	30.00

Table 5: Scheduling results using GA with projects with 35 activities. GA has been run 20 times per project.

Project ID	Mean	St. dev.	Min	Max	Percentage of minima found (%)
#40	175.58	0.92	175.06	178.30	25.00
#41	197.95	4.29	196.25	215.51	75.00
#42	171.48	0.79	169.60	171.92	15.00
#43	155.25	2.63	154.20	166.41	20.00
#44	172.25	7.72	167.30	204.20	30.00

Table 6: Scheduling results using GA with projects with 40 activities. GA has been run 20 times per project.

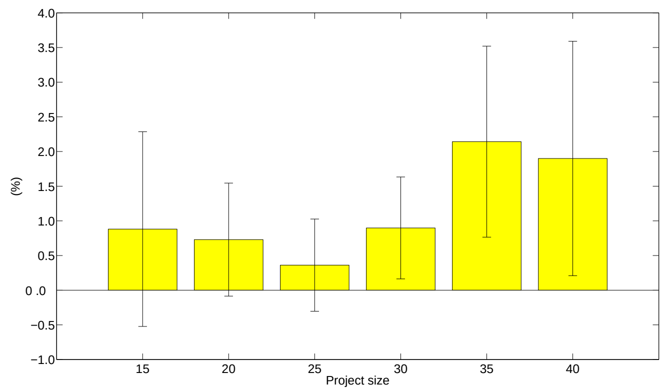

Figure 6: Relative distance to the minima of the solutions of large/complex projects shown in Tables 1-6.

to guarantee the optimal schedule, it converges around a particular value at each project. Furthermore, we define a relative error measure dividing the distance between the solutions found and the minima by the $\Psi(\mathcal{S}, \mathcal{Z})$ value of the minima. Figure 6 shows this relative error (in percentage) according to the different project sizes. We can state that the relative error of the solutions found is very small (around 1%, as Figure 6 shows) and thus, the presented GA achieves very good results when it deals with the e-MRCPSP. In this sense, we can state that despite the greater difficulty of solving the e-MRCPSP good solutions can be found with traditional meta-heuristics, what should encourage researchers and engineers to take account of energy issues in scheduling problems.

5. Conclusions and future work

Smart grid involves new hourly-based day-ahead energy tariffs that condition resource energy costs. While resources involved in project scheduling are mainly optimised based on their economic and makespan costs, nowadays, they should also be scheduled taking into account the energy they consume and the price of such energy while meeting the constraints imposed by compromised load profiles. To that end, we present and model, in this work, the multi mode project scheduling problem under time-dependent energy prices and consumption compromises, e-MRCPSP.

The formalisation of the problem is presented for the first time in this paper. Resources include their energy costs in energy units per time. Energy tariffs are time-dependent. Load shapes are defined so that there is a compromise to keep energy consumption inside a particular pattern. The e-MRCPSP is more complex than multi mode project scheduling problems due to the fact that resource economic cost depends on when activities are carried out.

To analyse the problem we solved it using two different methods: a branch and bound algorithm and a genetic algorithm. The branch and bound is useful to schedule projects with few activities, however an optimal solution cannot be found within reasonable time when the complexity of the problem increases. At that point, the use of a genetic algorithm is justified in order to find good schedules (even if they are not the optimal) into a feasible amount of time.

Experimentation corroborates that the complexity of the e-MRCPSP is greater than MRCPSP. But the solutions provided by e-MRCPSP shows how energy can be saved in exchange of makespan, as could be expected. The final cost of the schedule can be tuned through the weights of the aggregation function. Future work should consider other approaches to deal with the multiple objectives than an aggregation function. However, the new problem presented in this paper, e-MRCPSP, enables engineers to take into account energy issues in scheduling problems.

Other research lines to be carried out is to search other solutions approaches to e-MRCPSP. Some examples are the *activity list representation* of the chromosomes used by [1] or the cooling down mechanism used in the simulated annealing algorithm proposed by [6]. On the other hand, it would be interesting to merge the e-MRCPSP presented in this paper with other MRCPSP such as those presented in [14; 20], which consider uncertain costs and durations, and [8; 24], which consider different modes of using a resource (i.e. time-varying availability).

Acknowledgements

This research project has been partially funded through BR-UdG Scholarship of the University of Girona granted to Ferran Torrent-Fontbona. Work developed with the support of the research group SITES awarded with distinction by the Generalitat de Catalunya (SGR 2014-2016) and the MESC project funded by the Spanish MINECO (Ref. DPI2013-47450-C2-1-R).

References

[1] Alcaraz, J., & Maroto, C. (2001). A robust genetic algorithm for resource allocation in project scheduling. *Annals of Operations Research*, *102*, 83–109.

[2] Alcaraz, J., Maroto, C., & Ruiz, R. (2003). Solving the Multi-Mode Resource-Constrained Project Scheduling Problem with genetic algorithms. *Journal of the Operational Research Society*, *54*, 614–626.

[3] Barrios, A., Ballestín, F., & Valls, V. (2011). A double genetic algorithm for the MRCPSP/max. *Computers & Operations Research*, *38*, 33–43.

[4] Bellenguez-Morineau, O., & Néron, E. (2007). A branch-and-bound method for solving multi-skill project scheduling problem. *RAIRO-Operations Research*, *41*, 155–170.

[5] Bose, I., & Pal, R. (2012). Do green supply chain management initiatives impact stock prices of firms? *Decision Support Systems*, *52*, 624–634.

[6] Bouleimen, K., & Lecocq, H. (2003). A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. *European Journal of Operational Research*, *149*, 268–281.

[7] Brucker, P., Drexl, A., Möhring, R., Neumann, K., & Pesch, E. (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, *112*, 3–41.

[8] Cheng, J., Fowler, J., Kempf, K., & Mason, S. (2015). Multi-mode resource-constrained project scheduling problems with non-preemptive activity splitting. *Computers & Operations Research*, *53*, 275–287.

[9] Escudero, L. F., Landete, M., & Marín, A. (2009). A branch-and-cut algorithm for the Winner Determination Problem. *Decision Support Systems*, *46*, 649–659.

[10] European Co-Directorate for Research Co-operation Energy (2007). European technology platform smartgrids - strategic research agenda for Europe's electricity networks of the future.

[11] European Co-Directorate General for Research Sustainable Energy Systems (2006). European smartgrids technology platform - vision and strategy for Europe's electricity networks of the future.

[12] Gellings, C. W. (2009). *The smart grid: enabling energy efficiency and demand response*. The Fairmont Press, Inc.

[13] Gottwalt, S., Ketter, W., Block, C., Collins, J., & Weinhardt, C. (2011). Demand side management - A simulation of household behavior under variable prices. *Energy Policy*, *39*, 8163–8174.

[14] Gutjahr, W. J. (2015). Bi-Objective Multi-Mode Project Scheduling Under Risk Aversion. *European Journal of Operational Research*, *246*, 421–434.

[15] Hartmann, S. (2001). Project scheduling with multiple modes: a genetic algorithm. *Annals of Operations Research*, *102*, 111–135.

[16] Hartmann, S., & Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, *207*, 1–14.

[17] Haupt, R. L., & Haupt, S. E. (2004). *Practical genetic algorithms*. Wiley-Interscience.

[18] Herroelen, W., Demeulemeester, E., & Reyck, B. (1999). A classification scheme for project scheduling. In J. Wglarz (Ed.), *Project Scheduling* (pp. 1–26). Boston, MA: Springer US volume 14 of *International Series in Operations Research & Management Science*.

[19] Jarboui, B., Damak, N., Siarry, P., & Rebai, A. (2008). A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems. *Applied Mathematics and Computation*, *195*, 299–308.

[20] Ji, X., & Yao, K. (2014). Uncertain project scheduling problem with resource constraints. *Journal of Intelligent Manufacturing*, .

[21] Jia, W., Kang, C., & Chen, Q. (2012). Analysis on demand-side interactive response capability for power system dispatch in a smart grid framework. *Electric Power Systems Research*, *90*, 11–17.

[22] Mitchell, M. (1999). An introduction to genetic algorithms. *Cambridge, Massachusetts London, England, Fifth printing*, .

[23] Mohsenian-Rad, A.-H., Wong, V. W. S., Jatskevich, J., Schober, R., & Leon-Garcia, A. (2010). Autonomous Demand-Side Management Based on Game-Theoretic Energy Consumption Scheduling for the Future Smart Grid. *IEEE Transactions on Smart Grid*, *1*, 320–331.

[24] Naber, A., & Kolisch, R. (2014). MIP models for resource-constrained project scheduling with flexible resource profiles. *European Journal of Operational Research*, *239*, 335–348.

[25] Ozdamar, L. (1999). A genetic algorithm approach to a general category project scheduling problem. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, *29*, 44–59.

[26] Sandholm, T., Suri, S., Gilpin, A., & Levine, D. (2005). CABOB: A fast optimal algorithm for winner determination in combinatorial auctions. *Management Science*, *51*, 374–390.

[27] Setlhaolo, D., & Xia, X. (2015). Optimal scheduling of household appliances with a battery storage system and coordination. *Energy and Buildings*, *94*, 61–70.

[28] Torrent-Fontbona, F. (2012). *Decision support methods for global optimization*. Master's thesis University of Girona.

[29] Vale, Z. A., Morais, H., & Khodr, H. (2010). Intelligent multi-player smart grid management considering distributed energy resources and demand response. In *Power and Energy Society General Meeting, 2010 IEEE* (pp. 1–7).

[30] Xiao, J., Ao, X.-T., & Tang, Y. (2013). Solving software project scheduling problems with ant colony optimization. *Computers & Operations Research*, *40*, 33–46.

[31] Ziarati, K., Akbari, R., & Zeighami, V. (2011). On the performance of bee algorithms for resource-constrained project scheduling problem. *Applied Soft Computing*, *11*, 3720–3733.