Evolutionary Scheduling of Flexible Offers for Balancing Electricity Supply and Demand

Tea Tušar
Department of Intelligent Systems
Jožef Stefan Institute
Ljubljana, Slovenia
Email: tea.tusar@ijs.si

Erik Dovgan
Department of Intelligent Systems
Jožef Stefan Institute
Ljubljana, Slovenia
Email: erik.dovgan@ijs.si

Bogdan Filipič
Department of Intelligent Systems
Jožef Stefan Institute
Ljubljana, Slovenia
Email: bogdan.filipic@ijs.si

Abstract—To address the needs of rapidly changing energy markets, an energy data management system capable of supporting higher utilization of renewable energy sources is being developed. The system receives flexible offers from producers and consumers of energy, aggregates them on a regional level and schedules the aggregated flexible offers to balance forecast energy supply and demand.

This paper focuses on formulating and solving the optimization problem of scheduling aggregated flexible offers within such a system. Three metaheuristic scheduling algorithms (a randomized greedy search, an evolutionary algorithm and a hybrid between the two) tailored to this problem are introduced and their performance is assessed on a benchmark test problem and two realistic problems. The best results are achieved by the evolutionary algorithms, which can efficiently handle thousands of aggregated flex-offers.

I. Introduction

Rapidly changing electrical energy markets, which are faced with deregulation, increased smart metering and requirements for higher utilization of renewable energy sources, seek new solutions to support their flexibility, ensure reliable supply, and balance the costs and benefits of the involved parties. A system to serve the needs of a deregulated electricity market and enable the integration of a higher rate of energy from distributed and renewable sources is being developed in the European Seventh Framework Programme project MIRABEL (Micro-Request-Based Aggregation, Forecasting and Scheduling of Energy Demand, Supply and Distribution) [1]. The project proposes a conceptual and infrastructural approach to supply and demand side management where electricity producers and consumers issue flexible offers (termed *flex-offers*), indicating flexibilities in start time and energy amount. These flex-offers are then processed by the MIRABEL system to balance electricity supply and demand.

As electricity market regulations vary across the countries, the Harmonized Electricity Market Role Model [2] was chosen as a common platform to build upon in MIRABEL. This model defines the roles of producers and consumers, sometimes denoted by a common term *prosumers*. A collection of metering points for imbalance settlement is called the *balance group* and the role providing balance responsibility and financial security for a balance group is the *balance responsible party* (BRP).

Balance group is the basic domain where the MIRABEL system will be applied. To assist the BRP in equalizing the inflows and outflows of electricity at the balance group endpoints the system provides:

- handling of the novel concept of flex-offers for electricity production and consumption,
- forecasting of electricity supply and demand,
- aggregation of flex-offers on a regional level, scheduling of electricity production and consumption based on aggregated flex-offers, and disaggregation of the scheduled flex-offers for the purpose of their contracting,
- a distributed, decentralized and scalable computer infrastructure to handle the data load from the prosumers.

The overall MIRABEL architecture and functionalities are described in [3]. This paper focuses solely on presenting and solving the problem of scheduling aggregated flex-offers and investigating the implications of the performance of scheduling algorithms for aggregation.

The paper is further organized as follows. First, the scheduling problem is formally defined refining its initial version from [4]. Then, three scheduling algorithms (the randomized greedy search, an evolutionary algorithm and a hybrid between the two) able to solve this problem are presented. Next, the algorithms are experimentally evaluated on three problems, where attention is paid to their optimality on small problems as well as their efficiency on large problems. Finally, the paper concludes with a summary of the presented work and future research directions.

II. Related Work

Scheduling in the energy sector is used for assigning energy units to producers, consumers and transportation lines. This problem is called the *unit commitment problem*. Here, a schedule defines when each unit is started, stopped, and how much energy it generates in order to minimize the cost while still satisfying the constraints. The techniques used for solving this problem can be classified into deterministic techniques, meta-heuristic techniques, and hybrid approaches combining deterministic and meta-heuristic techniques [5].

Examples of deterministic techniques are the extensive enumeration, priority list, dynamic programming, linear programming, Lagrangian relaxation, and branch-and-bound. Extensive enumeration [6], [7] checks all possible schedules and stores the result in a table. Since the whole search space is checked,

optimality of the result is assured. However, such a method can only be used on small search spaces. Priority list [8], [9] commits the units according to previously given priorities, such as production costs, production capacity, etc. Priorities are then used to create a schedule. Dynamic programming [10], [11] operates with stages where each stage consists of states of energy units. It starts at the end of the observed time interval and defines the optimal state transition for each state in each stage until the initial stage is reached thus finding the optimal schedule for the observed time interval. An initial implementation of linear programming was used for solving only the problem of scheduling thermal units [12]. Several other linear programming approaches have been developed since, for example, a mixed-integer programming technique [13], and combinations with other approaches [14], [15]. The main idea of the Lagrangian relaxation [16], [17] is the usage of a dual function. This function is a cost function that combines power balance and security constraints. After the dual cost function is defined, the reformulated problem allows decomposing the original problem into smaller sub-problems dealing with the operation of single energy units. The branch-and-bound approach [18], [19] first finds a lower bound of schedules and afterwards the near-optimal schedule by taking into account the lower bound. In addition, an upper bound can be also used. If one or more bounds are used, the search space is significantly reduced. The search is performed with the priority list approach. Examples of meta-heuristic techniques used for scheduling in energy sector are artificial neural networks [20]–[22], simulated annealing [6], [23], evolutionary algorithms [24]–[26], and tabu search [27].

Although these methods have previously been used for scheduling units with non-flexible offers, they are general enough to be able to handle units with flexible offers, too. Such an example is presented in this paper, where evolutionary algorithms are used to schedule flexible offers.

III. Problem Description

The scheduling problem addressed in this work is described by first introducing the concepts used and then presenting the formal problem definition.

A. Concepts

1) Time Intervals: In the MIRABEL scheduling problem, time is discretized into *time step intervals* (usually 15 minutes long) and every time related concept is defined as a multiple of time step interval. This holds also for the *scheduling interval*—the interval for which scheduling needs to be performed.

2) Mismatch and Imbalance Prices: For each time step interval in the scheduling interval, a *mismatch* amount is given. Mismatch represents the difference between all produced and consumed energy that is forecast for the corresponding time step interval. Mismatch is positive when forecasts imply more produced than consumed energy, and negative when the consumed energy is forecast to exceed the produced energy. Mismatch is merely a prediction of imbalances that are about to happen in reality when the time in question has passed and

the producers and consumers of energy will adhere (or not) to the forecast behavior. The BRP has to pay penalties for any such imbalance. Their price is called the *imbalance price* and is usually different for positive and negative imbalances.

3) Flex-Offers: A *flex-offer* represents an offer of a consumer to buy energy from the BRP or an offer of a producer to sell energy to the BRP. Each flex-offer is defined with:
  – start time flexibility,
  – energy intervals, where each interval is defined with its duration, price, and energy flexibility, and
  – a total energy constraint.

The start time flexibility denotes on which time step intervals the execution of the flex-offer can start. For example, the production flex-offer from Fig. 1 can start on four time step intervals, while the consumption flex-offer can start from the second to the eleventh time step interval. The flex-offer's energy intervals are shown as boxes in Fig. 1 (the production flex-offer has four, while the consumption flex-offer has two). Each energy interval has the duration expressed in multiples of time step intervals, a price per energy amount and energy flexibility, i.e., the minimum and maximum energy that can be assigned to the energy interval (indicated with arrows in Fig. 1). The produced energy is regarded as positive, and the consumed energy as negative.
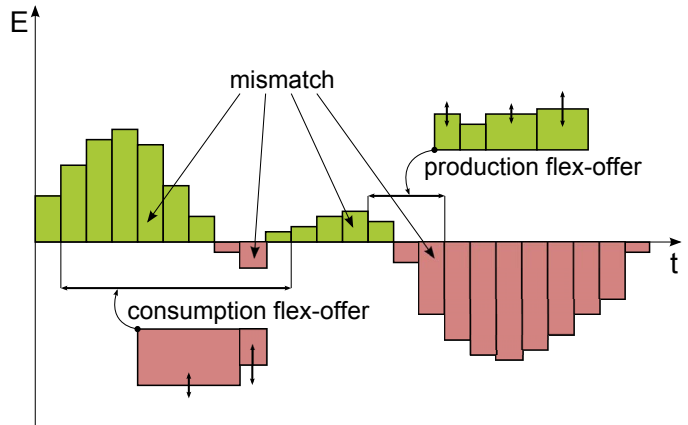

Fig. 1. An example of mismatch and two flex-offers (one production and one consumption).

4) Market Energy and Prices: The mismatch that remains after all flex-offers have been scheduled can sometimes be sold (positive mismatch) or bought (negative mismatch) on the *energy market* at a price called the *market price*.

B. Problem Formulation

The MIRABEL scheduling problem is defined with:
  – the scheduling interval,
  – mismatch and imbalance prices, which are given for every time step interval in the scheduling interval,
  – the possibilities of selling/buying energy on the market and the corresponding market prices, and
  – (aggregated) flex-offers with all their defining information.

The task is to fix time and energy flexibilities of all given flex-offers and establish the amount of energy to be bought (sold) on the market so that all constraints are satisfied and the cost for the BRP is minimized.

In the continuation of this section, the problem is formally defined in terms of decision variables, constraints and the objective function, using the notation from Table I.

TABLE I
NOTATION USED IN PROBLEM DESCRIPTION.

Variable	Meaning
n	number of flex-offers
m	number of time step intervals in the scheduling interval
E_I^i	remaining mismatch amount at time step interval i
p_{I+}^i, p_{I-}^i	price of positive/negative imbalance at time step interval i
$\delta_{M+}^i, \delta_{M-}^i$	possibility of selling/buying energy on the market at time step interval i (can only be 0 or 1)
E_M^i	market energy amount at time step interval i
p_{M+}^i, p_{M-}^i	price of energy sold/bought on the market at time step interval i
s_k	schedule for the k-th flex-offer
t_k	start time of the k-th flex-offer
n_k	number of energy intervals of the k-th flex-offer
E_k^j	energy amount of the j-th energy interval of the k-th flex-offer
p_k^j	price of the j-th energy interval of the k-th flex-offer

1) Decision Variables: The decision variables of the scheduling problem are represented with the vector of flex-offer schedules

$$S = \{s_1, s_2, \ldots, s_n\}.$$

Each flex-offer schedule s_k is defined with its start time and energy amounts for each energy interval of the flex-offer:

$$s_k = (t_k, E_k^1, E_k^2, \ldots, E_k^{n_k}).$$

The vector of market energy amounts

$$M = \left\{E_M^1, E_M^2, \ldots, E_M^m\right\}$$

needs to be specified too, however their values depend on flex-offer schedules and are therefore not considered as decision variables (we explain how they are determined later on).

The pair (S, M) of flex-offer schedules and market energy amounts is a *solution* to the scheduling problem.

2) Constraints: Flex-offer flexibilities are in fact constraints on flex-offer schedules. Each flex-offer schedule s_k is subject to the following constraints:

$$t_k \in \left[\underline{t_k}, \overline{t_k}\right],$$

where $\underline{t_k}$ and $\overline{t_k}$ are the earliest and latest start times for t_k, and

$$E_k^j \in \left[\underline{E_k^j}, \overline{E_k^j}\right] \text{ for all } j = 1, \ldots, n_k,$$

where $\underline{E_k^j}$ and $\overline{E_k^j}$ are the minimum and maximum energy amounts for E_k^j. A prosumer may also provide a total energy constraint that determines the sum of energy that needs to be produced (or consumed) in the entire flex-offer schedule:

$$\sum_{j=1}^{n_k} E_k^j \in \left[\underline{E_k}, \overline{E_k}\right],$$

where $\underline{E_k}$ and $\overline{E_k}$ are the minimum and maximum total energy constraint amounts for the k-th flex-offer. Any of the specified constraint intervals can degenerate into a point, leaving no flexibility. For example, the production flex-offer from Fig. 1 has no flexibility in the second energy interval.

A flex-offer schedule is considered *feasible* if its start time and energy amounts are fixed within the given flexibilities, and all the energy constraints are satisfied. A solution is feasible if all of its schedules are feasible.

3) Objective Function: The mismatch that remains after all flex-offers have been scheduled can be treated either as mismatch for which a penalty will have to be paid or as energy that can be put on the market (if this is possible). This decision can be made independently for each time step interval. Since the price of selling/buying energy on the market is as a rule lower than the price of imbalances, when there is a possibility of selling/buying energy on the market, all remaining mismatch energy will be put on the market:

$$E_M^i = \begin{cases} E_I^i, & \text{if } (E_I^i > 0 \text{ and } \delta_{M+}^i = 1) \text{ or } (E_I^i < 0 \text{ and } \delta_{M-}^i = 1) \\ 0, & \text{otherwise} \end{cases}$$

for all $i = 1, \ldots, n$. This also means that for each time step interval i, where $\left|E_M^i\right| > 0$, the mismatch energy $E_I^i = 0$.

The objective of the scheduling problem is to minimize the cost for the BRP. The cost for the BRP c of a solution (S, M) consists of the cost of remaining negative imbalances c_{I-}, the cost of remaining positive imbalances c_{I+}, the cost of flex-offers c_{FO} and the cost of the energy bought on the market c_{M-} minus the profit from the energy sold on the market c_{M+}:

$$c(S, M) = \underbrace{\sum_{\substack{i=1 \\ E_I^i<0}}^{m} p_{I-}^i \left|E_I^i\right|}_{c_{I-}} + \underbrace{\sum_{\substack{i=1 \\ E_I^i>0}}^{m} p_{I+}^i E_I^i}_{c_{I+}} + \underbrace{\sum_{k=1}^{n} \left(\sum_{j=1}^{n_k} p_k^j E_k^j\right)}_{c_{FO}}$$
$$+ \underbrace{\sum_{\substack{i=1 \\ E_M^i<0}}^{m} p_{M-}^i \left|E_M^i\right|}_{c_{M-}} - \underbrace{\sum_{\substack{i=1 \\ E_M^i>0}}^{m} p_{M+}^i E_M^i}_{c_{M+}},$$

where each cost is calculated as the product of energy and its price and the notation from Table I is used. Note that while buying energy on the market increases the cost for the BRP, selling energy decreases it (hence minus in the formula above). Similarly, the BRP must buy the energy produced by the production flex-offers (which increases the total cost), while the energy consumed by the consumption flex-offers represents profit for the BRP and decreases the total cost. This is expressed by the sign of the energy amount E_k^j.

IV. SCHEDULING ALGORITHMS

The scheduling problem formulated in this work differs from the scheduling problems treated in the literature either in the context of production systems (e.g., [28]) or energy sector (e.g., [29]). Unlike the usually scheduled activities, flex-offers are structured, consisting of several energy intervals, each

interval with its own properties. Regarding the decision variables, in addition to start time, flex-offer scheduling involves determining energy amounts for each energy interval of every flex-offer. This substantially increases the problem complexity in terms of the number of candidate solutions. Similarly, besides the start time and energy amount flexibilities that act as constraints on decision variables, total energy constraints may be defined, which contribute to the problem specificity. Finally, the objective function is not related to a time measure, but is rather a composed cost function.

These characteristics and the expected large number of flex-offers to be processed make the MIRABEL scheduling problem non-standard and highly complex. Since known heuristics cannot be applied, we choose stochastic metaheuristic algorithms for tackling this problem, namely greedy search, an evolutionary algorithm and a hybrid between the two.

All presented scheduling algorithms use the same *solution representation*, *fitness function* and *stopping criterion*. A solution is represented with the vector of feasible flex-offer schedules, while the vector of market energy amounts is used only in solution evaluation. The fitness function is equal to the cost function, where the best solution has the lowest cost. The stopping criterion is the time elapsed from the start of scheduling.

A. Randomized Greedy Search

A non-randomized greedy search algorithm for this problem would schedule flex-offer sequentially, selecting at each step the flex-offer that would minimize the total fitness. Such a procedure cannot efficiently handle a large number of flex-offers, therefore we propose a randomized version that works as presented in Fig. 2.

Randomized greedy search
1) Until the stopping criterion is met, repeat the following steps:
   a) Repeat until all flex-offers have been scheduled:
      i) Randomly select a flex-offer.
      ii) Find the best schedule for this flex-offer.
      iii) Add this schedule to the solution.
   b) Save the best solution found so far.
2) Return the best solution.

Fig. 2.   Pseudocode of the randomized greedy search.

Selecting a random sequence of flex-offers instead of an optimal one removes the most computationally expensive part of the non-randomized version of the algorithm. In fact, finding the best schedule for one flex-offer (step 1(a)ii in Fig. 2, see also Fig. 3) requires only a few operations.

Suppose flex-offer k consists of a single energy interval that expands over n_k time step intervals. Finding the best energy amount for this flex-offer (step 1b in Fig. 3) requires to inspect only a limited number of possibilities[1]: the minimal energy amount, the maximal energy amount and all energy amounts that eliminate at least one underlying mismatch (these can be

---
[1]The proof to this statement is omitted due to space limitations.

Finding the best schedule for a given flex-offer
1) For each possible start time of the given flex-offer repeat the following steps:
   a) Set the start time of the given flex-offer to the current time.
   b) Find the best energy amounts for the flex-offer in this time positioning.
   c) Correct the energy amounts if they do not satisfy the total energy constraint.
   d) Evaluate this schedule.
2) Return the best schedule.

Fig. 3.   Pseudocode for finding the best schedule for a given flex-offer.

at most n_k). If the flex-offer consists of several energy intervals, each is processed individually, which means that the number of different possibilities is added, not multiplied.

Note that when the total energy constraint is present, the energy amounts might need to be corrected to accommodate it. This too is performed in a greedy manner keeping in mind the goal of minimization of the total cost.

While it is possible to find the best schedule for a single flex-offer, there is no guarantee of optimality on larger problems even for the non-randomized version of the greedy search algorithm. The key problem is the sequential handling of flex-offers. Because flex-offers have energy intervals that expand over multiple time step intervals, sequential fixing of energy amounts does not produce optimal solutions. Therefore, we tried to solve this problem also with an evolutionary algorithm that handles all flex-offers simultaneously.

B. Evolutionary Algorithm

Evolutionary algorithms (EAs) are stochastic optimization algorithms that start with a population of random solutions and use evolutionary principles of selection, crossover and mutation to find better and better solutions [30]. Here we use a non-generational EA as described in Fig. 4.

Evolutionary algorithm
1) Create and evaluate an initial population of random solutions.
2) Until the stopping criterion is met, repeat the following steps:
   a) Select two parents using tournament selection.
   b) Cross the parents to get two offspring solutions.
   c) For each offspring solution repeat:
      i) Mutate and evaluate the offspring.
      ii) Replace the worst solution of the population with the offspring, if it is better than the worst solution.
3) Return the best solution in the population.

Fig. 4.   Pseudocode of the evolutionary algorithm.

EAs require an initial population of solutions, where the number of solutions in the population is given by the parameter n_{pop}. In our case, the initial population consists of random solutions where the start times and energy amounts of all schedules are chosen randomly.

Tournament selection is a popular selection scheme of EAs, where parent solutions are chosen using a tournament among solutions (the one with the best fitness wins the tournament).

The parameter n_{tour} determines how many solutions take part in the tournament.

Crossover is a procedure of creating two offspring solutions from two parent solutions by swapping some of their "genetic material". There exist many different ways to perform crossover. Here, we use the multi-point crossover, where the flex-offer schedules of the two parent solutions are swapped between the first and second, third and fourth (and so on) crossover points. The number of points used in the multi-point crossover is an algorithm parameter, called n_{cross}. Crossover has an additional parameter p_{cross}, which defines the probability of applying crossover to pairs of parent solutions.

Usually, mutation in EAs is used to perform small random perturbations of solutions. In our scheduling problem, where fine-tuning of a large number of schedules is needed, mutation can be used to accelerate the search for optimal schedules. This is carried out by performing local optimization of flex-offer schedules as mutation. Mutation has a single parameter called p_{mut} that defines the probability of mutating a flex-offer schedule. When a flex-offer schedule is mutated, one of the three different local optimization procedures is applied to it (each can be chosen with a probability of 1/3):

- optimize the start time and energy amounts,
- optimize only the start time, or
- optimize only the energy amounts.

The first local optimization procedure is the same as already presented in Fig. 3. The second local optimization procedure leaves the energy amounts of the flex-offer schedule as they are and explores only different start times. The placement that yields the best fitness is chosen. Finally, if the third local optimization procedure is used, the energy amounts are optimized for the current start time of the flex-offer schedule. If by changing the energy amounts of the flex-offer schedule the total energy constraint is violated, the same correction as in greedy search is applied.

Since the offspring replace only the worst individuals of the population, at any time the best solution in the population is also the best solution found by the algorithm.

Because of its crossover and mutation operators, the EA is able to explore a larger set of solutions than the randomized greedy search algorithm. While this is no guarantee for optimality, EAs should have an advantage over greedy search.

C. Evolutionary Algorithm with Randomized Greedy Search

While the presented EA and randomized greedy search already have some characteristics in common (for example, the local optimization of a single flex-offer schedule), they can be further hybridized by including greedy search in the initialization of the EA. Specifically, some of the solutions in the initial population of the hybrid algorithm are created using randomized greedy search instead of randomly. The share of such solutions is set using the parameter p_{greedy}. Apart from the initialization, the hybrid algorithm is the same as the EA described previously.

TABLE II
COMPLEXITY OF THE SIMPLE BENCHMARK PROBLEMS WITH REGARD TO THE NUMBER OF COMBINATIONS OF START TIME SETTINGS AND THE TIME NEEDED TO EXPLORE ALL OF THEM USING EXHAUSTIVE SEARCH.

| | Simple benchmark problems (# of flex-offers) | | | | |
	1	2	3	4	5
# of combinations	1	2	300	5400	31500
time [s]	0.01	0.02	0.03	0.19	0.76

| | Simple benchmark problems (# of flex-offers) | | | | |
	6	7	8	9	10
# of combinations	2.7×10^5	3.2×10^6	6.5×10^6	2.2×10^8	3.6×10^8
time [min]	0.11	0.79	2.61	121.78	128.91

V. EXPERIMENTS

The three presented algorithms were experimentally evaluated on two problems that try to simulate the scheduling problems expected to arise in the MIRABEL system: the day-ahead (scheduling is done for the whole day ahead) and intra-day (scheduling is done for a few hours ahead) scheduling problems. As we cannot assess the optimal solution for these two problems, a few simple benchmark problems with known optima are introduced and the algorithms are tested to see how well they approximate the optima on these problems.

A. Data

As the MIRABEL system is not yet fully integrated, scheduling is not performed on true aggregated flex-offers and with actual forecast mismatch. Instead, artificial data manufactured to resemble the properties of real data are used as input to scheduling. Roughly half of the mismatch is positive and half negative, while the prices of imbalances are set to mimic the real ones (with higher prices in periods of peak energy demand). Similarly, half of the received flex-offers are production and half consumption ones. Both sets include flex-offers without total energy constraints as well as flex-offers with total energy constraints. Selling and buying energy on the market is possible only for some time step intervals.

B. Setup

1) Simple Benchmark Problems: The simple benchmark problems are instances of the scheduling problem where there are no dependencies among energy values from different time step intervals. This means that the flex-offers have no total energy constraint and all their energy intervals have duration of a single time step interval. If the number of flex-offers is very small, exhaustive search of the problem space is possible, which gives us the true optimum of the problem. We experimented with 10 instances of the simple benchmark problem—having $1, 2, \ldots, 10$ flex-offers, respectively. As the number of combinations of start time settings grows exponentially with the number of flex-offers (and so does the time needed to explore them using exhaustive search[2], see Table II), we could not perform the benchmark tests on larger problems. For all

---

[2]All experiments presented in this work were run on a computer with a 2.8 GHz Intel Core i7 CPU and 8 GB RAM.

instances, the scheduling interval is set to 24 hours (with underlying 15 minute time step intervals).

2) Day-Ahead and Intra-Day Scheduling Problems: The day-ahead and intra-day scheduling problems differ only in the length of the scheduling interval (they all have 15 minute time step intervals). For the day-ahead problems, the scheduling interval is set to 24 hours, while for the intra-day it equals 3 hours. Each problem had four instances with 10, 100, 1000 and 10 000 flex-offers. These enable us to inspect the performance of scheduling algorithms when dealing with problems of different magnitude.

3) Parameters of the algorithms: Regardless of the problem instance, parameters of the EAs are set as follows:

- $n_{pop} = 100$,
- $n_{tour} = 3$,
- $n_{cross} = \lceil n \times 5\% \rceil$,
- $p_{cross} = 0.5$,
- $p_{mut} = 1$.

Note that $p_{mut} = 1$ means that all schedules are always subject to local optimization. Additionally, the parameter p_{greedy} of the hybrid algorithm is set to 0.5 (which means half of the population is initialized randomly), while the randomized greedy search has no parameters. All algorithms were run on all problem instances 10 times. The time available for the algorithms (the stopping criterion) depends on the size of the problem. It is set to 1 second for problems with up to 10 flex-offers, 5 seconds for problems with 100 flex-offers, 1 minute for problems with 1000 flex-offers and 15 minutes for the largest problems (with 10 000 flex-offers).

C. Results and Discussion

1) Simple Benchmark Problems: Table III presents the results of the three algorithms on the 10 simple benchmark problems. The figures denote the number of times an algorithm was able to reach the optimal solution of the corresponding problem instance (out of 10 runs). We can see that on problems with 4 or more flex-offers the randomized greedy search cannot reach the optimum in the specified time of 1 second, but gets stuck in a local optimum (usually not much worse than the global one). The EAs perform better, however it is interesting to observe that the hybridized version does not improve the results of the original one (it often gets stuck in the same local optimum as the randomized greedy search). Note also that for problems with 6 flex-offers or more, our algorithms save (a lot of) time compared to the exhaustive search.

Results of the EA show that on problems with up to 8 flex-offers the algorithm is able to reach the optimum in the majority of the runs, while this happens only approximately half of the times on the problems with 9 and 10 flex-offers. Nevertheless, this makes us confident that the EA can efficiently solve also more complex problems, although not always optimally.

2) Day-Ahead and Intra-Day Scheduling Problems: The results of the algorithms on the day-ahead and intra-day scheduling problems are presented in Figs. 5 and 6, while their statistical significance is shown in Table IV. The random

TABLE III
RESULTS OF ALGORITHMS ON SIMPLE BENCHMARK PROBLEMS. THE FIGURES DENOTE THE NUMBER OF TIMES AN ALGORITHM WAS ABLE TO REACH THE OPTIMUM OF THE CORRESPONDING PROBLEM INSTANCE (OUT OF 10 RUNS).

Algorithm	Simple benchmark problems (# of flex-offers)									
	1	2	3	4	5	6	7	8	9	10
Greedy	10	10	10	0	0	0	0	0	0	0
EA	10	10	10	10	9	10	10	10	5	6
EA + Greedy	10	10	10	10	1	10	10	6	4	0

TABLE IV
RESULTS OF THE STATISTICAL SIGNIFICANCE TESTS—THE P VALUES OF THE 2-TAILED UNPAIRED T-TEST (P VALUES LOWER THAN 0.05 MEAN THERE IS A STATISTICAL SIGNIFICANT DIFFERENCE BETWEEN THE TWO SAMPLES).

Problem	Algorithm	Greedy	EA
Day-ahead 10 flex-offers	EA	1.47×10^{-15}	
	EA + Greedy	1.78×10^{-41}	0.05
Day-ahead 100 flex-offers	EA	1.23×10^{-28}	
	EA + Greedy	2.02×10^{-13}	7.48×10^{-3}
Day-ahead 1000 flex-offers	EA	7.05×10^{-35}	
	EA + Greedy	1.16×10^{-19}	0.26
Day-ahead 10 000 flex-offers	EA	1.64×10^{-35}	
	EA + Greedy	3.04×10^{-3}	2.23×10^{-5}
Intra-day 10 flex-offers	EA	2.21×10^{-20}	
	EA + Greedy	1.59×10^{-17}	0.96
Intra-day 100 flex-offers	EA	3.59×10^{-21}	
	EA + Greedy	6.15×10^{-15}	0.80
Intra-day 1000 flex-offers	EA	2.16×10^{-30}	
	EA + Greedy	1.76×10^{-19}	0.54
Intra-day 10 000 flex-offers	EA	1.70×10^{-38}	
	EA + Greedy	5.40×10^{-14}	2.48×10^{-8}

search algorithm (where solutions are created the same way as in the random initialization of the EA) was included in these tests and serves as a lower bound estimate on the performance of the other algorithms.

We can see that the randomized greedy search finds a good solution very quickly, but fails to substantially improve it in the remaining time. In fact, both EAs statistically significantly outperform the randomized greedy search on all problem instances. Better improvement of solutions in time is achieved by the EA, while hybridization with randomized greedy search, which was expected to accelerate the convergence of the EA, statistically significantly outperforms the EA only on two problem instances. The comparison with random search shows how the larger problems are in fact more difficult to solve.

It is important to look at these results also from the point of view of the entire MIRABEL system, where scheduling is closely knit with aggregation. Good schedules can be constructed only with the appropriate balance between aggregation and scheduling. If aggregation is performed to a very small degree, aggregated flex-offers keep most of their flexibilities, but their high number makes it impossible for scheduling to find good schedules in a limited amount of time. On the other hand, if aggregation is performed to a very large degree, scheduling will be able to optimally schedule the few aggregated flex-offers it gets, but these have already lost almost all their
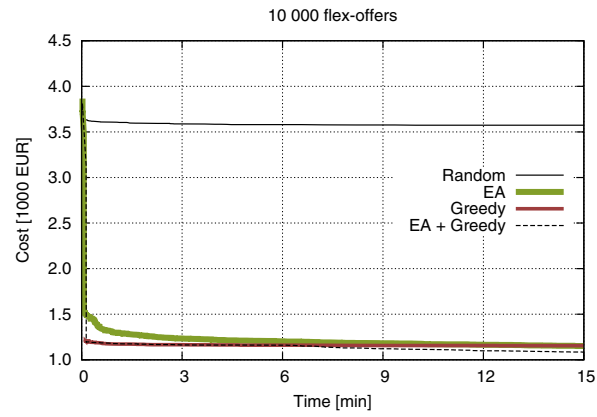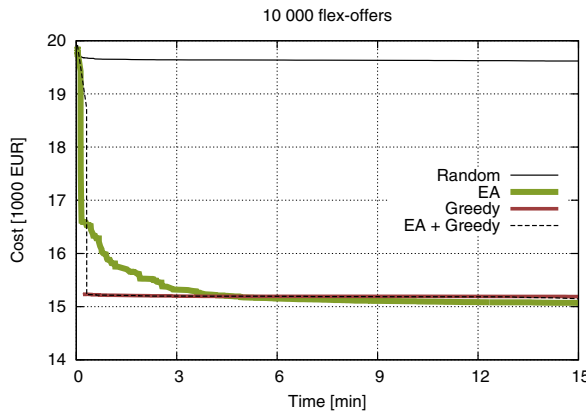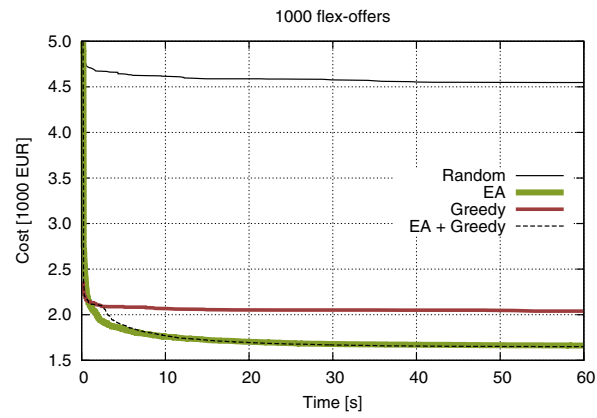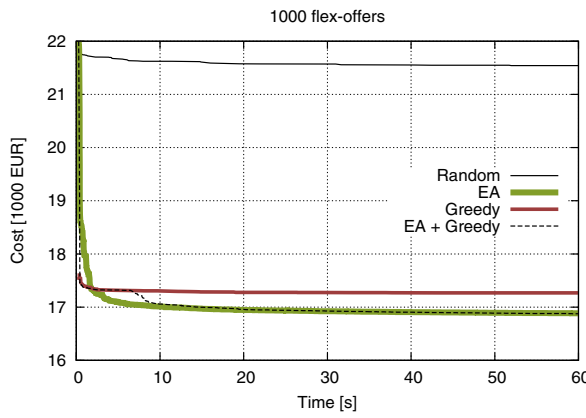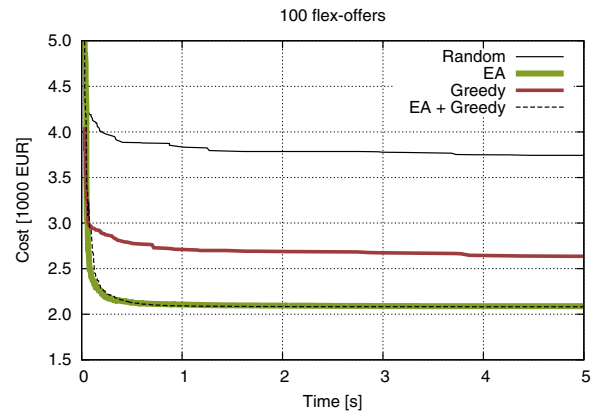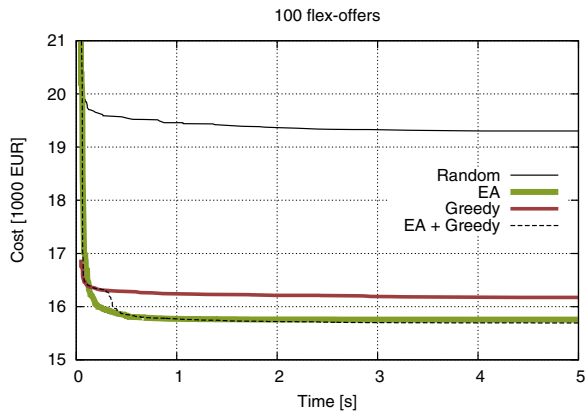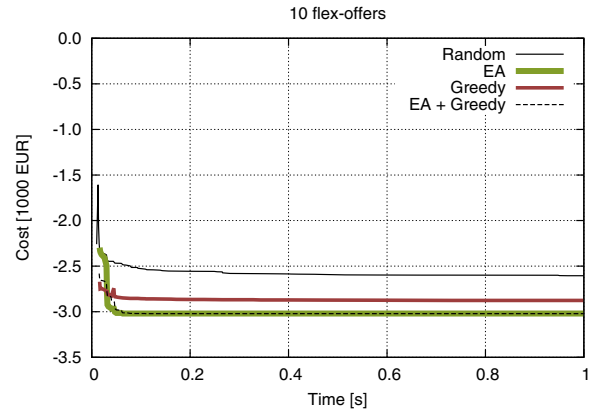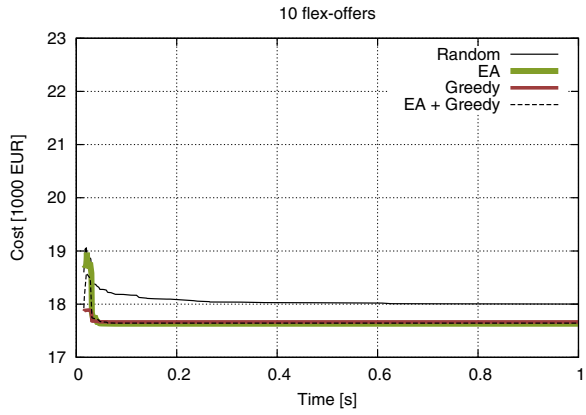
Fig. 5.   Average results of the algorithms on the day-ahead problem instances.

Fig. 6.   Average results of the algorithms on the intra-day problem instances.

flexibility making the final schedule of disaggregated flex-offers far from optimal. The results show that both EAs are quick to find good solutions on the smaller problem instances, while on the largest problem instances, the algorithms must be ran for minutes to achieve convergence. This might not be reasonable in practice, therefore we suggest aggregation should be performed in such a way, that at most a few thousands of aggregated flex-offers are passed to scheduling.

VI. CONCLUSION

This paper introduced the scheduling problem handled by the MIRABEL energy data management system, which assists the BRP in balancing the electricity supply and demand using flex-offers from prosumers. Three metaheuristic algorithms for solving this problems were presented and their performance was evaluated on two realistic and one benchmark test problem. Two conclusions can be drawn from their results: first, to solve this problem, evolutionary algorithms should be preferred over the randomized greedy search as they achieve better results, and second, scheduling is efficient for a thousand (or a few thousands) of flex-offers and aggregation must be used appropriately to assure the number of aggregated flex-offers does not exceed this amount. However, this is a preliminary guideline as testing of the integrated aggregation and scheduling is needed to achieve a better approximation of the real world conditions of this problem. Another direction for future work is to explore additional scheduling algorithms and try to improve the efficiency of the current ones.

REFERENCES

[1] "Mirabel project," Available from http://www.mirabel-project.eu/, retrieved on January 6, 2012.

[2] European Transmission System Operators, "The harmonized electricity market role model, version 2009-01," Available from http://www.scribd.com/doc/48492720/Role-Model-v2009-01, retrieved on January 6, 2012.

[3] Z. Marinšek, G. Černe, H. Berthold, C. Nychtis, F. J. Rumph, M. Konsman, H. Frey, L. Šikšnys, T. B. Pedersen, D. Kaulakiene, B. Filipič, and M. Böhm, "D1.2 Final Role Model and Process Specification," MIRACLE Consortium, Tech. Rep., 2010.

[4] T. Tušar, E. Dovgan, and B. Filipič, "Scheduling of flexible electricity production and consumption in a future energy data management system: problem formulation," in *Proceedings of the 14th International Multiconference Information Society – IS 2011*, vol. A. Jožef Stefan Institute, 2011, pp. 96–99.

[5] L. Sun, Y. Zhang, and C. Jiang, "A matrix real-coded genetic algorithm to the unit commitment problem," *Electric Power Systems Research*, vol. 76, no. 9–10, pp. 716–728, 2006.

[6] S. Sen and D. P. Kothari, "Optimal thermal generating unit commitment: a review," *Electrical Power and Energy Systems*, vol. 20, no. 7, pp. 443–451, 1998.

[7] A. J. Wood and B. F. Woolenberg, *Power Generation, Operation and Control*. New York: John Wiley and Sons, 1984.

[8] R. H. Kerr, J. L. Scheidt, A. J. Fontana, and J. K. Wiley, "Unit commitment," *IEEE Transactions on Power Apparatus and Systems*, vol. 85, no. 5, pp. 417–421, 1966.

[9] F. N. Lee, "Application of commitment utilization factor (CUF) to thermal unit commitment," *IEEE Transactions on Power Systems*, vol. 6, no. 2, pp. 691–698, 1991.

[10] C. K. Pang, G. B. Sheble, and F. Albuyeh, "Evaluation of dynamic programming based methods and multiple area representation for thermal unit commitments," *IEEE Transactions on Power Apparatus and Systems*, vol. 100, no. 3, pp. 1212–1218, 1981.

[11] W. L. Snyder, H. O. Powell, and J. C. Rayburn, "Dynamic programming approach to unit commitment," *IEEE Transactions on Power Systems*, vol. 2, no. 2, pp. 339–350, 1987.

[12] L. L. Garver, "Power generation scheduling by integer programming development of theory," *Transactions of the American Institute of Electrical Engineers Power Apparatus and Systems, Part III*, vol. 81, no. 3, pp. 730–734, 1962.

[13] J. A. Muchkstadt and R. C. Wilson, "An application of mixed-integer programming duality to scheduling thermal generating systems," *IEEE Transactions on Power Systems*, vol. 87, no. 12, pp. 1968–1978, 1968.

[14] S. Takriti and J. R. Birge, "Using integer programming to refine Lagrangian-based unit commitment solutions," *IEEE Transactions on Power Systems*, vol. 15, no. 1, pp. 151–156, 2000.

[15] A. Farag, S. A. Baiyat, and T. C. Cheng, "Economic load dispatch multi-objective optimisation procedures using linear programming techniques," *IEEE Transactions on Power Systems*, vol. 10, no. 2, pp. 731–738, 1995.

[16] S. J. Wang, M. Shahidehpour, D. S. Kirschen, S. Mokhtar, and G. D. Irisarri, "Short-term generation scheduling with transmission and environmental constraints using an augmented Lagranian relaxation," *IEEE Transactions on Power Systems*, vol. 10, no. 3, pp. 1294–1301, 1995.

[17] A. G. Bakiritzis and D. E. Zoumas, "Lambda of Lagrangian relaxation solution to unit commitment problem," in *IEE Proceedings – Generation, Transmission and Distribution*, vol. 147, no. 2, 2000, pp. 131–136.

[18] A. I. Cohen and X. Yoshimura, "A branch and bound algorithm for unit commitment," *IEEE Transactions on Power Apparatus and Systems*, vol. 102, no. 2, pp. 444–451, 1983.

[19] C. L. Chen and S. C. Wang, "Branch-and-bound scheduling for thermal generating units," *IEEE Transactions on Energy Conversion*, vol. 8, no. 2, pp. 184–189, 1993.

[20] C. Wang and S. M. Shahidehpour, "Effects of ramp rate limits on unit commitment and economic dispatch," *IEEE Transactions on Power Systems*, vol. 8, no. 3, pp. 1341–1350, 1993.

[21] V. B. A. Kasangaki, H. M. Sendaula, and S. K. Biswas, "Stochastic Hopfield artificial neural network for electric power production costing," *IEEE Transactions on Power Systems*, vol. 10, no. 3, pp. 1525–1533, 1995.

[22] M. P. Walsh and M. J. Malley, "Augmented Hopfield network for unit commitment and economic dispatch," *IEEE Transactions on Power Systems*, vol. 12, no. 4, pp. 1765–1774, 1997.

[23] F. Zhuang and F. D. Galiana, "Unit commitment by simulated annealing," *IEEE Transactions on Power Systems*, vol. 5, no. 1, pp. 311–318, 1990.

[24] D. Dasgupta and D. R. McGregor, "Thermal unit commitment using genetic algorithms," in *IEE Proceedings – Generation, Transmission and Distribution*, vol. 147, no. 5, 1994, pp. 459–465.

[25] S. A. Kazarlis, A. G. Bakirtzis, and V. Petridis, "A genetic algorithm solution to the unit commitment problem," *IEEE Transactions on Power Systems*, vol. 11, no. 1, pp. 29–36, 1996.

[26] T. T. Maifeld and G. B. Sheble, "Genetic-based unit commitment algorithm," *IEEE Transactions on Power Systems*, vol. 11, no. 3, pp. 1359–1370, 1996.

[27] X. Bai and S. M. Shahidehpour, "Hydro-thermal, scheduling by tabu search and decomposition method," *IEEE Transactions on Power Systems*, vol. 11, no. 2, pp. 968–974, 1996.

[28] M. L. Pinedo, *Scheduling: Theory, Algorithms and Systems*, 3rd ed. Springer, New York, 2008.

[29] H. Y. Yamin, "Review on methods of generation scheduling in electric power systems," *Electric Power Systems Research*, vol. 69, no. 2–3, pp. 227–248, 2004.

[30] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. Springer, Berlin, 2003.