USING AGGREGATION TO IMPROVE THE SCHEDULING OF FLEXIBLE ENERGY OFFERS

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

Laurynas Šikšnys, Torben Bach Pedersen
Department of Computer Science, Aalborg University, Denmark
{siksnys; tbp}@cs.aau.dk

Erik Dovgan, Bogdan Filipič
Department of Intelligent Systems, Jožef Stefan Institute, Ljubljana, Slovenia
{erik.dovgan; bogdan.filipic}@ijs.si

Abstract Changing electricity markets call for new ways of handling supply and demand. The desired goal is to increase the utilization of renewable energy while ensuring reliable supply and minimizing the costs. We present an approach aiming at this goal that handles a large number of flexible energy offers from producers and consumers by aggregating them and scheduling these aggregates to minimize a cost function. We explore the influence of aggregation on the performance of scheduling, establishing that a trade-off between keeping the flexibilities of flexible offers and reducing their number is what yields the best results.

Keywords: Aggregation, Energy system, Flexible energy, Optimization, Scheduling.

1. 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 amount 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, 7]. 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 production/consumption start time and energy amount.

To assist the *balance responsible party* (BRP) in balancing electricity supply and demand, the MIRABEL 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, scheduling of electricity production and consumption based on aggregated flex-offers, and disaggregation of the scheduled aggregated flex-offers for the purpose of their contracting,
- a distributed, decentralized and scalable computer infrastructure to handle the data load from the prosumers.

We focus on the tasks of aggregation, scheduling and disaggregation.

The problem of scheduling flex-offers is similar to the *unit commitment problem*, where 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 [2, 4, 5]. This paper presents the concept of flex-offers, which is different from the units and requires customized methods for their aggregation and scheduling. The novel contributions of this paper are the use of aggregation to empower scheduling of a high number of flex-offers and a study of the effect of aggregation parameters on the scheduling results.

The paper is further organized as follows. First, the MIRABEL system, all relevant concepts, and the flex-offer scheduling problem are introduced. Then, the aggregation procedure is explained, followed by the presentation of the scheduling algorithms. Next, we present a use case where the aggregation parameters are experimentally evaluated. Finally, the paper concludes with a summary of the presented work.

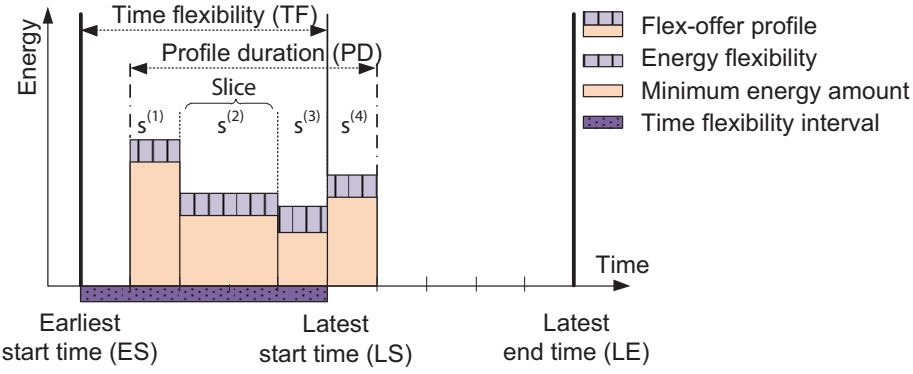

Figure 1. Example of a flex-offer with denoted attributes.

2. MIRABEL System

The central concept in MIRABEL is a *flex-offer*, which 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 defines the following elements (see Fig. 1): start time flexibility TF (from the earliest start time ES to the latest start time LS) and energy profile consisting of several time slices. Each slice is defined with its duration, price and energy flexibility. The sum of all slice durations is called profile duration PD.

The MIRABEL energy data management system processes the *prosumer* (producer or consumer) flex-offers as follows (see Fig. 2). When a prosumer sends a flex-offer, it is accepted or rejected depending on its price and energy fitting some predefined constraints. The accepted flex-offers are stored in the pool of flex-offers and periodically processed by aggregation. Aggregation generalizes many individual flex-offers, producing fewer *aggregated flex-offers*. Simultaneously, *mismatch* (difference between produced and consumed energy) and imbalance prices are forecast based on the past energy production, consumption, imbalance prices and weather forecast. Scheduling is performed on aggregated flex-offers in order to minimize the total cost of the schedule by taking into account the energy amount and prices of flex-offers and mismatch. When scheduling completes, a "coarse schedule" is produced. It is represented by *scheduled aggregated flex-offers*, which are then disaggregated into many *scheduled flex-offers*, thus producing a "fine schedule". Flex-offer aggregation and disaggregation are performed so that aggregated scheduled flex-offers can always be converted into scheduled flex-offers while respecting the initial flex-offer constraints. Moreover, the coarse and fine schedules are equal, i.e., total energy values at every time interval

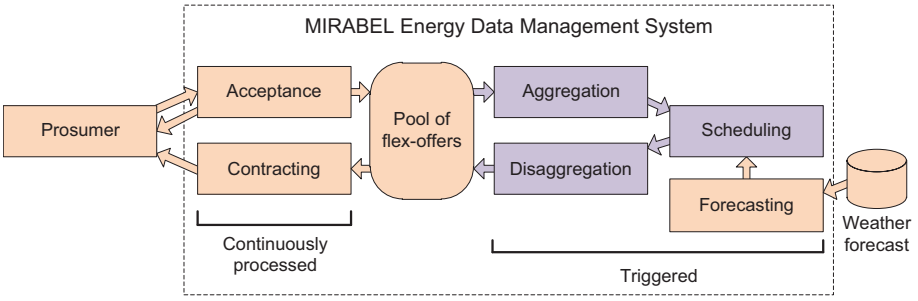

Figure 2. Flex-offer processing in the MIRABEL system.

are equal for both schedules. The disaggregated scheduled flex-offers are stored in the pool of flex-offers. When the start time of a flex-offer is approaching, the flex-offer is retrieved from the pool and the contract containing the fixed start time, energy price and energy amount for each slice is produced and sent to the prosumer.

The MIRABEL *scheduling problem* tackled in this paper corresponds to fixing the start time and energy flexibilities of all given (aggregated) flex-offers so that all constraints are satisfied and the cost for the BRP is minimized. The cost for the BRP consists of the cost of remaining negative imbalances, the cost of remaining positive imbalances and the cost of flex-offers. Note that 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.

3. Aggregation

Aggregation takes a set of flex-offers F as input and produces a set of aggregated flex-offers A ($|A| \leq |F|$) as output. The aggregation is performed in two general steps: *grouping* and *N-to-1 aggregation*.

In the *grouping step*, the flex-offer set F is partitioned into N groups $G_1, G_2, \ldots, G_N$, $1 \leq N \leq |F|$, each containing mutually similar flex-offers. The criterion for flex-offer similarity is user-specified: two flex-offers f_1 and f_2 are similar (and may potentially be included into the same group) if, for every flex-offer attribute a, their respective values f_1^a and f_2^a differ by no more than a user-specified tolerance T^a, i.e., $|f_1^a - f_2^a| \leq T^a$ for all attributes a. In this study, tolerances are defined for flex-offer attributes ES (earliest start time), TF (time flexibility) and PD (profile duration), and are called *aggregation parameters*. Such a grouping of flex-offers is similar to the way the similarity group-by op-

erator [8] for relational databases groups tuples based on a user-specified maximum group diameter.

In the *N-to-1 aggregation step*, flex-offers from group G are aggregated into a single aggregated flex-offer f_A according to the following procedure:

i. Set time flexibility interval as follows: $f_A^{\mathrm{ES}} = \min_{f \in G} f^{\mathrm{ES}}$ and $f_A^{\mathrm{TF}} = \min_{f \in G} f^{\mathrm{TF}}$, i.e., set the earliest start time and time flexibility values of f_A equal to the to lowest earliest start time and time flexibility values of the flex-offers in G.

ii. Build a new profile for f_A by adding minimum and maximum energy amounts for slices at the respective time intervals across profiles of all flex-offers from G. If the corresponding slices at some time step have different durations, they are partitioned in order to unify their durations.

The aggregation parameters control the "shape" of aggregated flex-offers. For example, if no tolerances are set, then all flex-offers from F will be passed to the N-to-1 aggregation step, thus producing a single aggregated flex-offer as output. The profile of such an aggregate is expected to be (relatively) long as it would span throughout the time interval including all flex-offers from F; and the time flexibility of such flex-offer would be equal to that of the flex-offer with the smallest time flexibility. If $T^{\mathrm{ES}} = 0$, then only those flex-offers with equal earliest start time values will be aggregated together. In this case, $|A| = l$, where l is the total amount of distinct ES values of flex-offers in F. If, in addition to T^{ES}, the time flexibility tolerance T^{TF} and the profile duration tolerance T^{PD} are set to 0, then the flex-offers with equal earliest start time, time flexibility, and profile duration values will be aggregated together (in this case latest end values will also be equal). In general, one or more tolerances can be set to any value higher than 0, thus obtaining different "shapes" of aggregated flex-offers.

4. Scheduling

Scheduling an aggregated flex-offer means setting its start time and energy amounts for every slice so that the cost for the BRP is minimized. Since multiple flex-offers need to be scheduled simultaneously, it is not possible to try every possible setting for each of them and heuristic algorithms need to be employed to efficiently solve this optimization problem. In this work we present two heuristic algorithms: local optimization and the evolutionary algorithm. Both use the following three optimization functions:

i. *Optimizing start time.* The flex-offer already has a defined start time and energy amounts for each slice. While keeping the energy amounts the same, we try to place the flex-offer at all its possible start times and store the setting that minimizes the total cost for the BRP.

ii. *Optimizing energy amounts.* Here, we keep the start time fixed and optimize only the energy amounts for each slice independently. While the energy amount can be set to any real number in the interval between the minimum and maximum amount, only a finite number of settings can yield the optimal result (the minimum amount, the maximum amount and the amount that eliminates the underlying imbalance). The energy amounts that result in the minimal cost for the BRP are stored.

iii. *Optimizing start time and energy amounts.* This heuristic optimizes energy amounts of the flex-offer for every possible start time setting (it combines the above two heuristics). The chosen settings are again the ones that result in the minimal cost for the BRP.

Local optimization (LO) constructs a solution as follows. It starts with a random solution (consisting of flex-offer schedules with randomly set start times and energy amounts) and sorts its flex-offers in a random order. For each flex-offer from the first to the last, a randomly chosen optimization function (see above) is applied. This construction procedure is repeated until the stopping criterion is satisfied.

The *evolutionary algorithm* (EA) starts with a population of random solutions. Until the stopping criterion is met, the EA selects two solutions using tournament selection, swaps their schedules using multi-point crossover and finally (instead of applying mutation) optimizes the solution the same way as local optimization does. The two newly constructed solutions replace the worst two solutions in the population. The basic difference between the LO and the EA is the use of evolution principles of population, selection and crossover in the EA [3].

As a benchmark, we apply also *random search* (RS), which constructs random solutions until the stopping criterion is met.

5. Experimental Evaluation

In this section, we experimentally evaluate the quality of scheduling when it is used with and without aggregation.

5.1 Use case

We assume a scenario which is typical in the MIRABEL context: on the day-ahead market, the BRP buys a certain amount of energy for all 24 hours of the following day and thus commits itself to balance the acquired production with the respective consumption at every hour. If for a particular hour the energy bought by the BRP does not match the consumed one, the BRP has to pay a penalty that is calculated based on the imbalance energy and its price. Therefore, one hour before the energy delivery day starts, the BRP utilizes flex-offers to balance the energy demand and supply for the subsequent 24 hours with the objective to minimize the total imbalances and thus to maximize its profit. In our experimental setting, we assume that the BRP collects flex-offers from energy consumers (but not producers) only. The maximum scheduling time is fixed to 10 min leaving at least 50 min to aggregation and contracting. The scheduling algorithm stops earlier if its best result has not been improved for one minute.

We use a synthetic flex-offer dataset from the MeRegio project [6], which is also used as a test dataset in the MIRABEL project. The dataset contains 100 000 flex-offers. The time is discretized at every 15 min (96 time stamps per 24 hours). The flex-offer attributes follow these distributions and have the following bounds: $\text{ES} \sim \mathcal{U}(0, 96)$, $1 \leq \text{ES} \leq 92$; $\text{TF} \sim \mathcal{N}(8, 4)$, $4 \leq \text{TF} \leq 12$; and $\text{PD} \sim \mathcal{N}(10, 10)$, $1 \leq \text{PD} \leq 20$. Profile slice durations are fixed to 1 time unit (15 min). In other words, flex-offers start between 0:00 and 23:00, their start time flexibility varies from 1 up to 3 hours, and their profile durations vary from 15 min to 5 hours. We assume that the BRP buys an amount of energy equal to that defined by the 100 000 flex-offers; this energy is distributed following the typical daily energy usage pattern (more energy used in day time, less at night). Additionally, we use real imbalance and retail energy prices from the Slovenian electricity market. All experiments were run on a computer with Quad Core Intel®Xeon®E5320 CPU, 16GB of RAM, and OpenSUSE 11.4 (x86_64) OS. We used Java 1.6 for all implementations.

5.2 Parameter settings

Experiments were performed with three scheduling algorithms: evolutionary (EA), local optimization (LO), and random search (RS). Every scheduling algorithm was executed ten times in order to obtain a more reliable estimate. The scheduling was performed with and without the prior aggregation of flex-offers. For experiments with the aggregation, three aggregation parameters were used: earliest start time tolerance

T^{ES}, time flexibility tolerance T^{TF}, and profile duration tolerance T^{PD}. For each of these parameters, two extreme values, 0 and ∞ (no bound is set), and the set of intermediate values were used in the experiments.

5.3 Results and Discussion

Figure 3 shows the average scheduling result when all combinations of the aggregation parameters are used and, in addition, when no aggregation is performed (see the marks at 100 000 flex-offers). As we can see, the flex-offer count has almost no direct influence on the scheduling result. Moreover, the RS performs worse compared to the LO and the EA. The evolution principles of the EA bring an advantage when more than 40 aggregated flex-offers need to be scheduled. When there is no aggregation, the LO does not find a single solution in the given amount of time, while the EA computes only the initial random population, achieving the same results as the RS. This means that some aggregation is needed to produce good results.

The remaining mismatch in the nonaggregated case is compared to the best found mismatch by the EA in Fig. 4. The combination of aggregation and scheduling successfully minimizes the cost for the BRP (and consequently the remaining mismatch) leaving some mismatch only in the beginning and in the end of the 24-hour interval. More specifically,

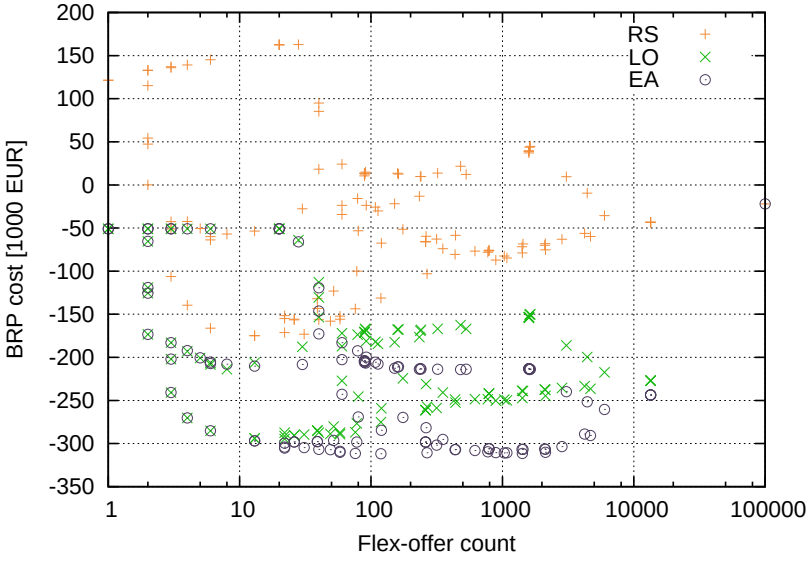

Figure 3. Influence of the flex-offer count on the average scheduling result.

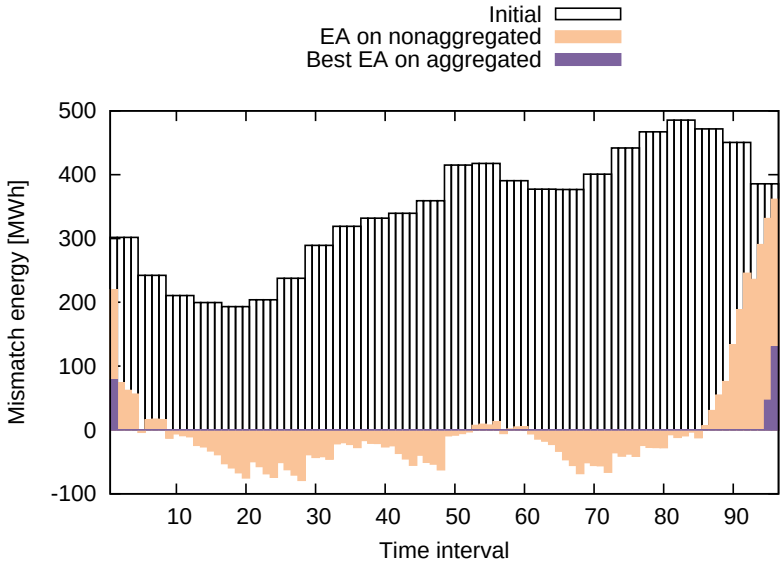

Figure 4. Aggregation impact on the remaining mismatch found by the EA.

if the flex-offers are favorably aggregated, the remaining mismatch equals only 5% of the remaining mismatch in the nonaggregated case.

In order to study the effect of aggregation on the scheduling results, we need to take a closer look at the aggregation parameters. Figure 5 presents the individual aggregation parameter impact on the aggregated flex-offer count and the average scheduling results found by the EA. As we can see, aggregation parameters T^{ES}, T^{TF}, and T^{PD} contribute similarly to flex-offer count reduction, but they have different impact on the quality of results. Specifically, keeping the value of T^{TF} as low as possible almost always guarantees better scheduling results compared to higher T^{TF} values. This means that in order to obtain better results, aggregation should preserve as much time flexibility as possible, which is exactly what low T^{TF} values achieve. The T^{ES} parameter has a different impact depending on whether $T^{\mathrm{TF}} = 0$ or $T^{\mathrm{TF}} = \infty$. When $T^{\mathrm{TF}} = \infty$, then long aggregated flex-offers (obtained with high T^{ES} values) normally result in worse scheduling results comparing to short aggregated flex-offers (obtained with low T^{ES} values). However, when $T^{\mathrm{TF}} = 0$, the scheduling result can be improved by lowering T^{ES} value until the increased aggregated flex-offer count starts to dominate and thus negatively influence scheduling. For example, the overall best scheduling results were achieved with the EA when $T^{\mathrm{TF}} = 0$ and $T^{\mathrm{ES}} = 7$ or 11. Finding the best T^{ES} value is another optimization problem. The

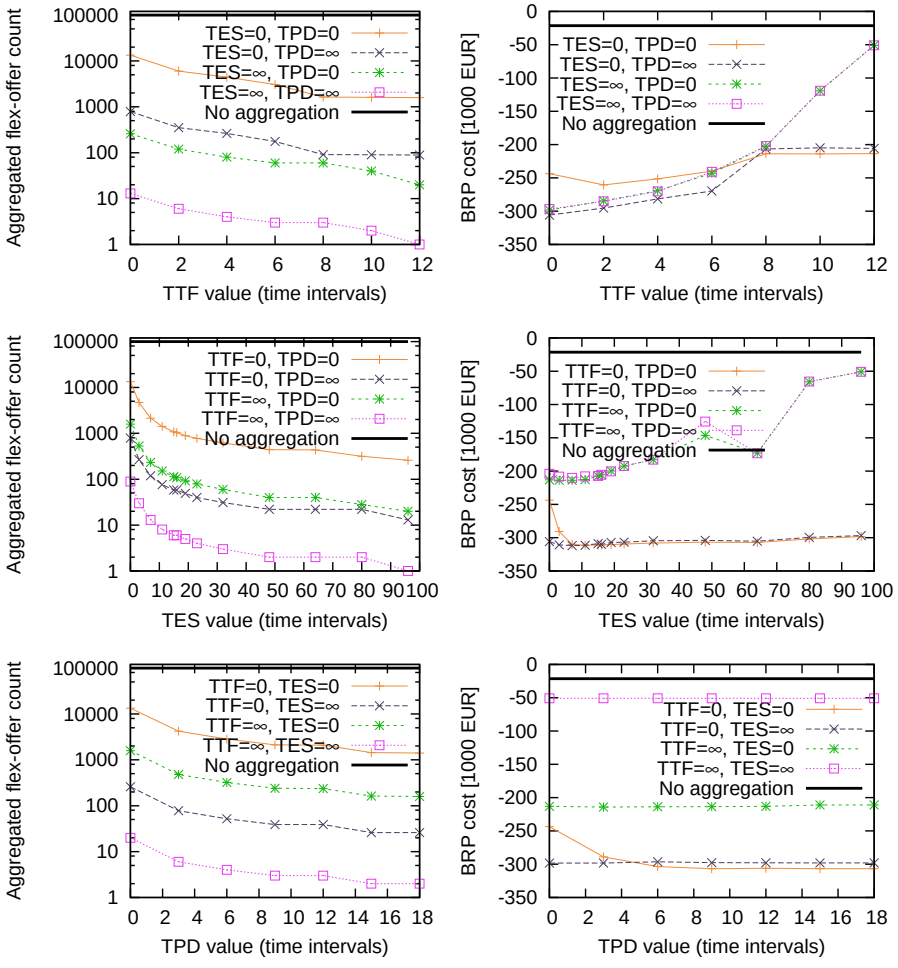

Figure 5. Aggregation parameters impact on flex-offer count (left column) and average scheduling result by the EA (right column).

third parameter T^{PD} has little impact on the scheduling result. While it decreases the aggregated flex-offer count, this does not contribute to achieving better scheduling results.

Finally, a note on the runtime of the scheduling algorithms. All algorithms stop when the best solution has not been improved in the last minute or 10 minutes have elapsed. In most of the cases with less than 300 aggregated flex-offers, the algorithms stop within 2 minutes. On average, the runtime of RS is longer than the runtime of the LO, which is in turn longer than the runtime of the EA. However, when more than 300 flex-offers need to be scheduled, the runtime of the EA becomes longer (this is not the case for the RS and the LO), approaching the limit 10 minutes for more than 600 flex-offers. This suggests the EA does not converge yet on such problems, continuing to improve its result until forced to stop. For the best reported result the EA spent 84 seconds on average.

6. Conclusion

The paper presented the aggregation and scheduling of flex-offers as carried out by the MIRABEL system. The focus was on exploring the influence of aggregation parameters on the quality of obtained flex-offer schedules. The application of aggregation and scheduling on a realistic use case has shown that the mere reduction of the flex-offer count is not enough to produce good scheduling results. It is important that the aggregated flex-offers keep as much time flexibility as possible, too. Therefore, a trade-off between keeping time flexibilities and reducing the number of aggregated flex-offers is what yields the best results for this problem (leaving only 5% of the mismatch of the nonaggregated case).

Acknowledgement

The work presented in this paper has been carried out in the project *Micro-Request-Based Aggregation, Forecasting and Scheduling of Energy Demand, Supply and Distribution* (MIRABEL) funded by the European Commission under the grant agreement number 248195, and under the research programme P2-0209 *Artificial Intelligence and Intelligent Systems* funded by the Slovenian Research Agency.

References

[1] H. Berthold, M. Boehm, L. Dannecker, F.-J. Rumph, T. Bach Pedersen, C. Nychtis, H. Frey, Z. Marinšek, B. Filipič, and S. Tselepis. Exploiting renewables by request-based balancing of energy demand and supply. In *Proc. IAEE European Conference*, 2010.

[2] D. Dasgupta and D. R. McGregor. Thermal unit commitment using genetic algorithms. *IEE Proc.-C*, 147(5): 459–465, 1994.

[3] Á. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. Springer, 2003.

[4] S. A. Kazarlis, A. G. Bakirtzis, and V. Petridis. A genetic algorithm solution to the unit commitment problem. *IEEE T. Power Syst.*, 11(1):29–36, 1996.

[5] T. T. Maifeld and G. B. Sheble. Genetic-based unit commitment algorithm. *IEEE T. Power Syst.*, 11(3):1359–1370, 1996.

[6] "MeRegio Project". Available from `http://www.meregio.de/en/`, retrieved on March 1st, 2012.

[7] "MIRABEL project". Available from `http://www.mirabel-project.eu/`, retrieved on March 1st, 2012.

[8] Y. N. Silva, W. G. Aref, and M. H. Ali. Similarity group-by. In *Proc. IEEE International Conference on Data Engineering*, pages 904–915, 2009.