Balancing energy flexibilities through aggregation

Valsomatzis, Emmanouil; Hose, Katja; Pedersen, Torben Bach

Published in:
Data Analytics for Renewable Energy Integration

DOI (link to publication from Publisher):
10.1007/978-3-319-13290-7_2

Publication date:
2014

Document Version
Accepted author manuscript, peer reviewed version

Balancing Energy Flexibilities through Aggregation

Emmanouil Valsomatzis, Katja Hose, and Torben Bach Pedersen

Department of Computer Science, Aalborg University

Abstract One of the main goals of recent developments in the Smart Grid area is to increase the use of renewable energy sources. These sources are characterized by energy fluctuations that might lead to energy imbalances and congestions in the electricity grid. Exploiting inherent flexibilities, which exist in both energy production and consumption, is the key to solving these problems. Flexibilities can be expressed as flex-offers, which due to their high number need to be aggregated to reduce the complexity of energy scheduling. In this paper, we discuss balance aggregation techniques that already during aggregation aim at balancing flexibilities in production and consumption to reduce the probability of congestions and reduce the complexity of scheduling. We present results of our extensive experiments.

Keywords: energy data management, energy flexibility, flex-offers, balance aggregation

1 Introduction

The power grid is continuously transforming into a so called Smart Grid. A main characteristic of the Smart Grid is the use of information and communication technologies to improve the existing energy services of the power grid and simultaneously increase the use of renewable energy sources (RES) [2]. However, the energy generation by renewable sources, such as wind and solar, is characterized by random occurrence and thus by energy fluctuations [5]. Since their power is fed into the power grid and their increased use is a common goal, they may provoke overload of the power grid in the future, especially in peak demand situations [4, 9]. Moreover, the use of new technologies, such as heat pumps and electrical vehicles (EV), and their high energy demand could also lead to electricity network congestions [13].

Within the Smart Grid, the EU FP7 project MIRABEL [3] and the ongoing Danish project TotalFlex [1] are using the flex-offer concept [17] to balance energy supply and demand. The concept is based on the idea that the energy consumption and production can not only take place in specific time slots, but be flexible and adjustable instead. For instance, an EV is parked during the night from 23:00 until 6:00. The EV could be charged or alternatively also act as an energy producer and feed its battery energy into the grid [15]. So the EV is automatically programmed to maximally offer 30% of its current battery energy

to the grid, corresponding to 2 hours of discharging. So, in a case of an energy demand or favorable energy tariffs, the EV will be discharged, e.g. from 1:00 to 2:00, offering 15% of its battery energy.

In the MIRABEL project, an Energy Data Management System (EDMS) is designed and prototyped. The EDMS aims at a more efficient utilization of RES by the use of the flex-offer concept. Such an EDMS is characterized by a large number of flex-offers that have to be scheduled (assign a specific time and energy amount) so that balancing supply and demand is feasible. Since it is infeasible to schedule a large number of flex-offers individually [17], an aggregation process is introduced, so that the number of flex-offers is decreased and consequently also scheduling complexity [18]. In the proposed EDMS architecture, the scheduling component is responsible for properly scheduling the aggregated flex-offers in order to balance out energy fluctuations. The TotalFlex project additionally considers balancing goal during aggregation so that imbalances are partially being handled by the balance aggregation.

The balance aggregation aims to aggregate flex-offers derived from consumption and production in order to create flex-offers with low energy demand and supply requirements. Thus, violations of the network's electricity capacities could be avoided. At the same time, the aggregated flex-offers still maintain flexibility that could further be used during scheduling to avoid grid congestions, reassure a normal grid operation, and amplify RES use. For instance, in the above mentioned EV example, there could also be another EV that needs 4 hours of charging, corresponding to 70% of its battery capacity. Charging could take place during the night from 22:00 to 5:00. So, the energy of the first EV could be used to partially recharge the second one, for example, from 23:00 to 1:00. Thus, instead of the two EVs, we consider an aggregated flex-offer that represents the demand of 2 hours charging, corresponding to 40% of battery capacity and the charge could take place from 23:00 to 5:00. In this work, we perform an extensive experimental investigation of the behavior of balance aggregation. We also propose alternative starting points for the techniques and evaluate the impact of aggregation parameters.

The remainder of this paper is structured as follows. Section 2 is describing the theoretical foundations and Section 3 related work. Section 4 explains how exactly balance aggregation works. Section 5 discusses results of our extensive experiments. We conclude in Section 6 with a discussion of our future work.

2 Related Work

The unit commitment problem [8,14], where balancing energy demand and supply is taken into consideration, has been extensively investigated through either centralized (e.g. [6,16]) or distributed approaches (e.g. [11,12]). Moreover, in [17], the unit commitment problem has been examined by handling the units as flex-offers and by using centralized metaheuristic scheduling algorithms. In [17], the economic dispatch stage of the unit commitment problem is also elaborated by applying a cost function and thus confronting potential imbalances.

Furthermore, aggregation that takes into account flexibilities with a flex-offer use case evaluation has been investigated in [19]. Scheduling aggregated flex-offers that only represent energy consumption and introducing aggregation as a pre-step of scheduling has been investigated in [18]. However, in this paper, we examine aggregation of flex-offers that takes into account one of the goals of scheduling, i.e., balancing. We do not address imbalances by using a cost function as in [18], but instead we handle imbalances as an effort to directly balance out energy amounts derived from supply and demand. To achieve that, we integrate balancing into the aggregation process. As a result, imbalances are partially handled and flexibility still remains to be used by the scheduling procedure. In this paper, we evaluate the techniques by taking into consideration energy flexibility representing not only from consumption as in [18, 19] but from production as well. Moreover, this work provides an extensive experimental evaluation of balanced aggregation techniques.

3 Preliminaries

We use the following definition based on [19].

Definition 1 *A flex-offer f is a tuple $f = (T(f), profile(f))$ where $T(f)$ is the start time flexibility interval and profile(f) is the data profile. Here, $T(f) = [t_{es}, t_{ls}]$ where t_{es} and t_{ls} are the* earliest start time *and* latest start time*, respectively.*

The data profile $profile(f) = s^{(1)}, \ldots, s^{(m)}$ where a slice *$s^{(i)}$ is a tuple $([t_s, t_e], [a_{min}, a_{max}])$ where $[a_{min}, a_{max}]$ is a continuous range of the amount and $[t_s, t_e]$ is a time interval defining the extent of $s^{(i)}$ in the time dimension.*

We consider three types of flex-offers: positive, negative, and mixed ones. Positive flex-offers have all their amount values of all their slices positive and correspond to energy consumption flex-offer. Negative flex-offers have all their amount values of all their slices negative and correspond to energy production flex-offers. All the other flex-offers are considered as mixed and express hybrid flex-offers. Figure 1 illustrates a mixed flex-offer f with four slices, $f = ([1, 7], s^{(1)}, s^{(2)}, s^{(3)}, s^{(4)})$.

The time is discretized into equal size units, e.g., 15 minutes and the amount dimension represents energy. Every slice is represented by a bar in the figure. The below and above bars of each slice represent the minimum and maximum amount value, a_{min} and a_{max}, respectively. A flex-offer also supports a lowest and a highest total amount that represents the minimum and the maximum energy required respectively [10].

Moreover, as defined in [19], the *time flexibility*, $tf(f)$, of a flex-offer f, is the difference between the latest and earliest start time, the *amount flexibility*, $af(s)$, is the sum of the amount flexibilities of all slices in the profile of f, and the *total flexibility* of f is the product of the time flexibility and the amount flexibility, i.e. $flex(f) = tf(f) \cdot af(s)$. For instance, the flex-offer in Figure **??** has $tf(f)$=7-1=6, $af(s)$=(3−1)+(4−2)+(2−(−4))+(−1−(−3))=12, and total flexibility: 6∗12=72.

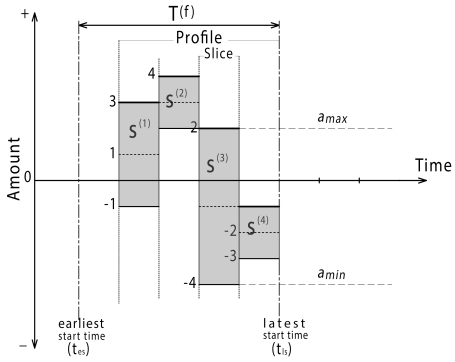


Figure 1. A mixed flex-offer

Furthermore, we consider as aggregation the process in which the input is a set of flex-offers and the output is also a set of flex-offers (aggregated) with a smaller or equal number of flex-offers. An aggregated flex-offer encapsulates one or more flex-offers and is able to describe the flex-offers that created it. Furthermore, a measurement that we use to evaluate the aggregation is the flexibility loss that is defined according to [19] as the difference between the total flex-offer flexibility before and after aggregation. We will further discuss the aggregation process, introduce balance aggregation, and two of its techniques in Section 4.

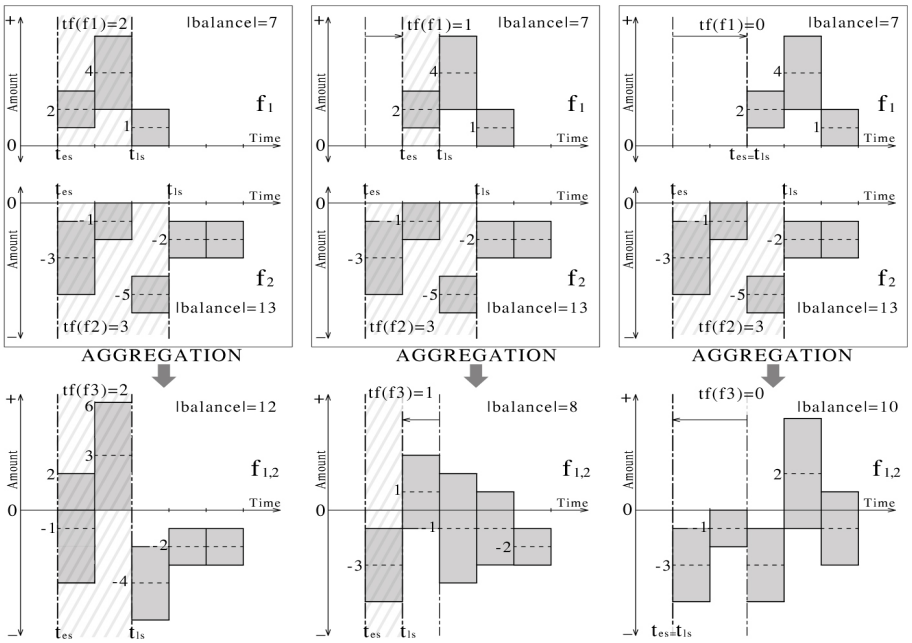
4 Balance aggregation

In order to describe balance aggregation we define the balance, $balance(f)$, of a flex-offer f, as the sum of the average amount values of each slice, and absolute balance, $absolute\_balance(f)$ as the sum of the average absolute values of the amounts for each slice of the flex-offer. For example in Figure 1, the flex-offer f=$([1,6], s^{(1)}, s^{(2)}), s^{(3)}, s^{(4)})$ has $balance(f)$=1+3−1−2=1 and $absolute\_balance(f)$=$|1|+|3|+|-2|\ +|-1|$= 9. Goal of the balance aggregation is to create aggregated flex-offers with low values of absolute balance. In this section, we sketch the aggregation process and describe approaches to achieve balance aggregation.

4.1 Flex-offer Aggregation

In Figure 2 we present a simple aggregation scenario where we aggregate two flex-offers, f_1 and f_2, creating the aggregated flex-offer $f_{1,2}$. Both f_1 and f_2, have time and amount and so does the aggregated one. As illustrated, the time flexibility (hatched area) of f_1 is 2 and of f_2 is 3. In the first column of the figure we show the *start alignment* aggregation [19]. In that case, we align the two flex-offers so that their profiles start at the earliest start time and then we sum the minimum and maximum amounts of each aligned slice. The time flexibility of the aggregated flex-offer is the minimum flexibility of the non-aggregated flex-offers. This reassures that all the possible positions of the *earliest start time* of the aggregated flex-offer will not violate the time constraint that the non-aggregated flex-offers have.

However, because the time flexibility of the aggregated flex-offer depends on the minimum flexibility of the flex-offers that participate in the aggregation, the flex-offers are grouped according to 2 different parameters, *EST* (*Earliest Start*



(a) Alignment1 (b) Alignment2 (c) Alignment3

Figure 2. Different alignments for aggregation

time Tolerance) and TFT (*Time Flexibility Tolerance*) to minimize flexibility losses [19]. The value of EST represents the maximum difference of the earliest start times that the flex-offers could have in order to belong to the same group. The value of TFT represents the maximum time flexibility differences that the flex-offers could have in order to be grouped together.

As we can see in the second and the third column of Figure 2, there are different start time profile combinations for the flex-offers that participate in the aggregation and result in different aggregated flex-offers. Note that the absolute balance of the aggregated flex-offer also depends on the start time profile combinations. For instance, in the second column where we shift the first flex-offer for one time unit, we see that the absolute balance of the aggregated flex-offer has reduced. On the other hand, continuing shifting the first flex-offer will increase again the absolute balance of the aggregated flex-offer, see third column in Figure 2. However, with only two flex-offers there are few combinations and the number of combinations increases exponentially with the number of flex-offers and larger time flexibilities. The balance aggregation aims at identifying start time profile combinations and aggregate flex-offer in a manner that will minimize the absolute balance of the aggregated flex-offer. At the same time, the two balance aggregation techniques considered in this work do not not explore the whole solution space and thus avoid in-depth search.

4.2 Balance aggregation

We examine two different approaches of implementing balance aggregation, the exhaustive greedy and the simple greedy. The techniques are trying to find start time combinations between positive and negative flex-offers that will lead to an aggregated one with a minimum absolute balance.

Both these greedy techniques have the same start point but exhaustive greedy aims to examine a larger solution space than simple greedy. After grouping the flex-offers according to the grouping parameters, both techniques sort all the flex-offers inside each group in a descending order regarding their balance and start the aggregation choosing the one with the minimum balance. The flex-offer with the minimum balance will be the most negative one representing a flex-offer derived from production that is usually less flexible than the positive ones. Afterwards, exhaustive greedy considers the maximum flexibility for the selected flex-offer and then examines all the possible earliest start time combinations with all the remaining flex-offers. The technique chooses the alignment of the flex-offer that provides the minimum absolute balance. It continues until the absolute balance is no longer reduced. If the absolute balance is not reduced, it restarts with the remaining flex-offers.

On the other hand, simple greedy also starts the aggregation by choosing the flex-offer with the minimum balance. However, aggregation continues with the flex-offer that has the balance which is closest to the opposite $(+/-)$ balance of the first one. It examines all the possible start time combinations of the flex-offer and chooses the aggregation that has the minimum absolute balance. It also continues the aggregation until the absolute balance of the aggregated flex-offer is not further reduced. In case the absolute balance is not reduced, it considers the aggregated flex-offer and continues with the remaining flex-offers. In Figure 3 we show how exhaustive greedy and simple greedy work in the same

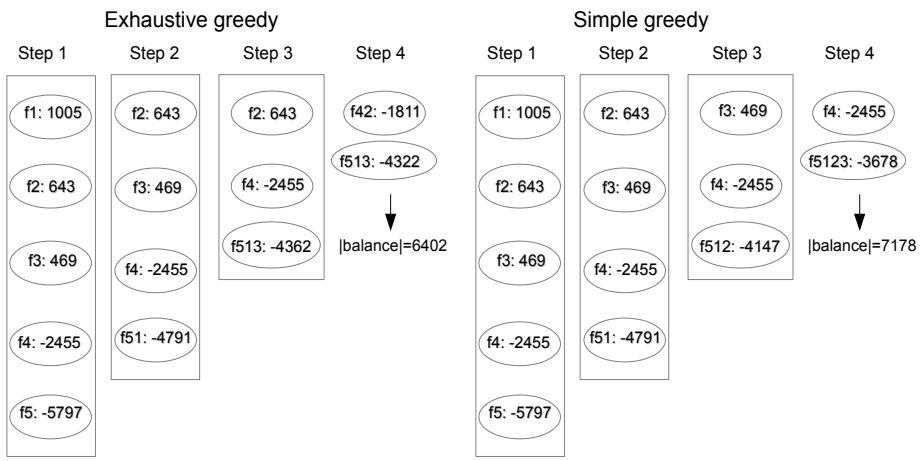


Figure 3. Exhaustive and simple greedy examples

group of flex-offers. The example is from one of our datasets and in the figure we show the balance of each flex-offer. We see that both the techniques start their aggregation by choosing the most negative flex-offer, f_5, and aggregate it with flex-offer f_1. However in step 3, exhaustive greedy chooses to aggregate with f_3 because it gives a better absolute balance and simple greedy with flex-offer f_2 since it is closer to the opposite of the f_{51} balance. As a result, exhaustive greedy continues the aggregation separately for f_2 and f_4 and simple greedy continues aggregation with f_3 leaving f_4 non-aggregated. Therefore, the two techniques lead to different absolute balance values.

5 Experimental Evaluation

In this section, we present an extensive experimental evaluation of the balance aggregation techniques a comparison to start alignment aggregation discussed in Section 4.

5.1 Experimental setup

For the evaluation of the balance aggregation techniques, we used 4 extensive experimental setups of 10 groups of 8 datasets, 320 datasets in total. Each setup is characterized by different time and amount probabilistic distributions corresponding to different energy scenarios.

The first experimental setup is based on the one described in [19]. It consists of 10 groups and each group has 8 datasets, 80 in total. In order to create each group of the 8 datasets, we first select flex-offers derived from the historical consumption time series of 5 random customers. Then, we incrementally add to each dataset flex-offers corresponding to the number of 5 more random customers. The last one, the eighth dataset, has flex-offers derived from historical consumption time series of 40 random customers. Afterwards, for every dataset we apply start alignment aggregation according to [19], with EST and TFT equal to zero, resulting in an aggregated positive flex-offer. In every aggregated flex-offer, a random number of amount slice, between zero and its time flexibility value is added. Finally, all the positive aggregated flex-offers are converted to negative ones. As a result, there are always one or more positive flex-offers that if being aggregated have the same opposite balance as a negative one. Since we add in each dataset flex-offers derived from five more customers, the datasets have an incremental number of flex-offers that approximately corresponds to 11K additional flex-offers for every 5 customers. The way the dataset is created reassures that whenever we apply aggregation with the parameters EST and TFT set to 0, it is feasible to create an aggregated flex-offer with zero absolute balance. The time flexibility values of the flex-offers follow a normal distribution N (8, 4) in the range [4, 12] and the number of the slices a normal distribution N (20, 10) in the ranges [10, 30]. Those profiles are from 2.5 to 7.5 hours long, with one to three hours time flexibility, which could represent flex-offers derived

mostly from EVs. The results of the experiments of this setup are illustrated in the first row of Figures 4- 11.

Regarding the second experimental setup, we also created 10 groups of 8 datasets. The number of the flex-offers is similar to the one of the previous dataset. Furthermore, historical consumption time series of customers and the flex-offer generator tool described in [7] were used to create the datasets. The flex-offer generator tool was used to generate both positive and negative flex-offers. For all the datasets the number of the positive (consumption) flex-offers is twice the number of the negative (production) ones. In addition, the number of the slices of the positive flex-offers follows the normal distributions $\mathcal{N}(20, 10)$ in the ranges $[10, 30]$ and of the negative flex-offers the normal distributions $\mathcal{N}(40, 20)$ in the ranges $[20, 60]$. The time flexibility values $(t_{ls} - t_{es})$ of the positive flex-offers and of the negative flex-offers follow a discrete uniform distribution on the interval $[1, 10]$ and [1,8] respectively. This setup aims to explore a scenario in which balancing out energy and production is theoretically feasible. Thus, the negative flex-offers are half the positive ones but with double profile length. Such negative flex-offers with long profiles and less flexibility than the flex-offers representing the consumption could simulate RES. On the other hand, flex-offers characterized by more time flexibility and shorter profiles represent flex-offers derived from mostly recent technological achievements such as EVs and heat pumps. The results of the experiments of this setup are illustrated in the second row of Figures 4- 11.

Our third experimental setup is created as the second one. These datasets are variations of the second one with a deviation regarding the length and the time. More specifically, the slices of the positive flex-offers follow the same distribution as before, but the negative flex-offers follow the normal distribution $\mathcal{N}(50, 10)$ in the ranges $[40, 60]$, which makes them longer. The time flexibility values $(t_{ls} - t_{es})$ of the positive flex-offers and of the negative flex-offers follow a discrete uniform distribution on the interval $[2, 18]$ and [1,6] respectively, making the positive flex-offers much more flexible regarding time. Such kind of flex-offers could represent not only EVs and heat pumps but also electronic devices as well. The results of the experiments of this setup are illustrated in the third row of Figures 4- 11.

Our last experimental setup is similar to the third one. However, the negative flex-offers in this setup are characterized by less energy flexible profiles compared to the positive ones, reflecting a scenario in which the RES are not that flexible regarding energy. Moreover, the positive flex-offers are twice the number of the negative ones. More specifically, the number of the slices for the positive and the negative flex-offers follow the normal distributions $\mathcal{N}(5, 2)$ and $\mathcal{N}(10, 2)$ in the ranges $[1, 10]$ and $[5, 15]$, respectively. The energy flexibility values of the positive flex-offers follow the normal distributions $\mathcal{N}(30, 10)$ in the range $[0, 50]$ and of the negative flex-offer $\mathcal{N}(20, 10)$ in the same range over the same amount of flexibility. The results of the experiments of this setup are illustrated in the fourth row of Figures 4- 11..

In the experiments we investigate the three aggregation techniques regarding the absolute balance, the flexibility loss, the number of the aggregated flex-offers,

and the processing time. We examine all four aspects in terms of scalability and grouping parameters, *EST*, and *TFT*. In terms of scalability, we set both the grouping parameters to zero, and examined the techniques in datasets with incremental numbers of flex-offers, starting with minimum 11K (approximately) and maximum 90K (approximately) flex-offers (Figures 4, 6, 8, and 10). For each experimental setup, we created 10 groups of datasets that have the same number of flex-offers to reduce any effect of the randomness that characterizes the dataset generation. Regarding the effect of the grouping parameters, we used datasets with almost 90K flex-offers and set one of the parameters stable and set to zero, and varied the other values from zero to six, respectively (Figures 5, 7, 9, and 11). For illustrating purposes, we show the average behavior of the similar datasets that there are in each group.

We also investigate the performance of exhaustive and simple greedy after alternating their starting point referring to them in all the figures as "exhaustive greedy" and "simple greedy1". We illustrate their performance when they both start by selecting the flex-offer with the maximum absolute balance instead of the one with the minimum balance. In the first row, and the first and second column of Figures 4, 6, and 8 there is an overlap between the illustrated lines of the techniques because the techniques showed similar behavior. The experiments were conducted on a 2.9GHz Intel core i7 processor with two cores, L2 Cache of 256 KB, L3 Cache of 4 MB and physical memory of 8 GB (4 of 4 GB of 1600MHz DDR3).

5.2 Absolute balance

The results for absolute balance are shown in Figures 4 and 5. In Figure 4, absolute balance scales almost linearly with the number of input flex-offers. All the techniques, exhaustive greedy, simple greedy and start alignment have almost the same performance (first row of Figure 4) in all the setups except the first one. In the first setup, the two greedy techniques, exhaustive and simple greedy, achieve a very low balance, first row of Figures 4 and 5.

More specifically, we see in the first column of Figure 4 that both techniques achieve a very close to zero balance when both EST and TFT are set to zero. When TFT is set to 6, exhaustive greedy achieves a lower balance than simple greedy (first row, second column in Figure 4), but close to zero for both. Due to the nature of the first experimental setup, zero absolute balance can be achieved and therefore the two greedy techniques achieve it. However, we observe that start alignment achieves the minimum absolute balance among the techniques in the last three setups when there is a large number of flex-offers, approximately 90K, see Figure 5. In the second to fourth line of Figure 5 we see that exhaustive greedy and simple greedy have absolute balance similar to start alignment, with exhaustive greedy achieving a slightly smaller value between the two greedy techniques.

In the last three experimental setups (second to fourth row in Figures 4 and 5), there is an overlap in all the slices between the positive and the negative flex-offers since the profiles of the negative flex-offers have at least double profile

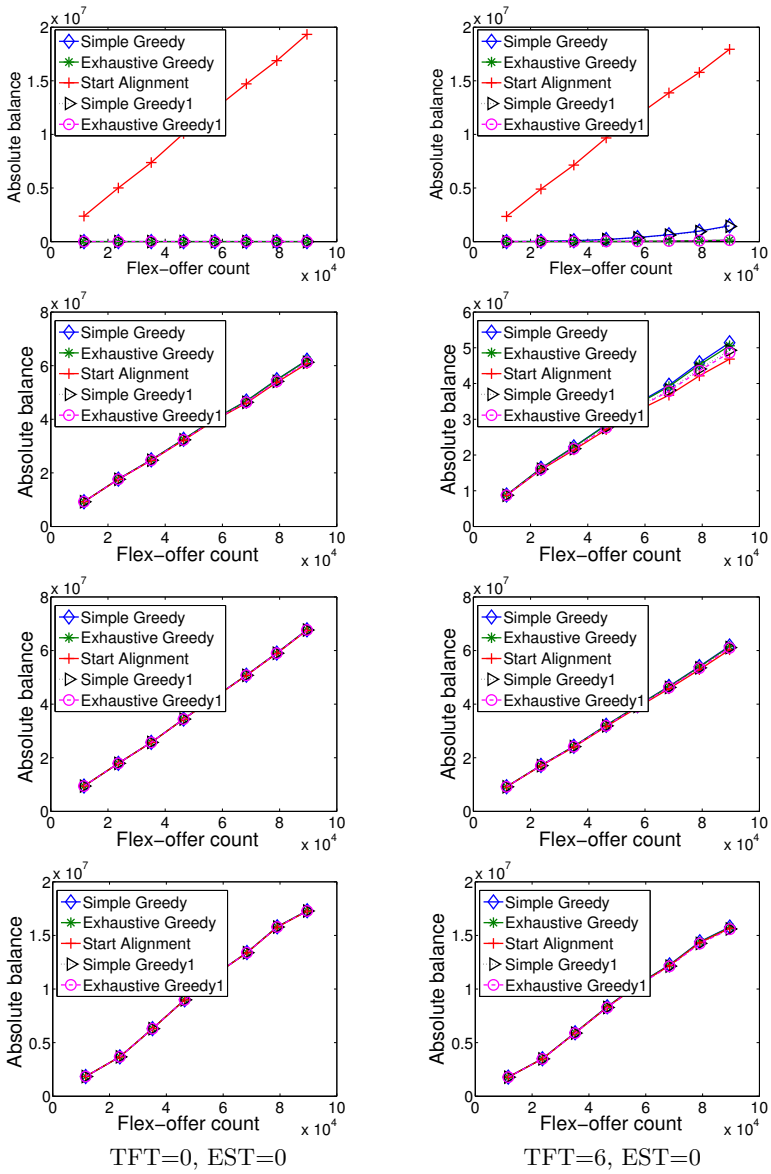


Figure 4. Results of the absolute balance in terms of scalability effect

length in comparison to the positive ones. As a result, start alignment achieves a low absolute balance and the two greedy techniques do not take advantage of the exploration of the solution space since all the differences in absolute balance between each possible aggregation are very low. They are very low because even if there might be an earliest start time combination between a positive and a negative flex-offer that reduces the total absolute balance of the mixed flex-offer, the percentage will be too low because the absolute balance is mostly reflected

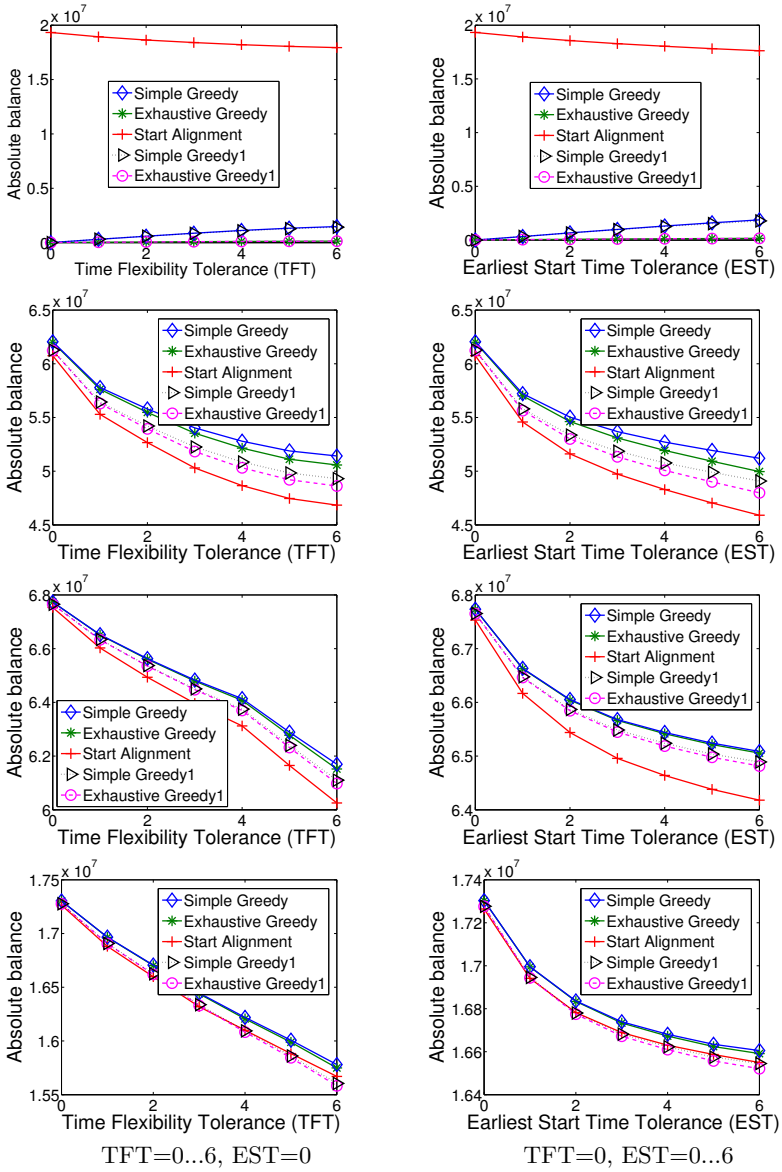


Figure 5. Results of the absolute balance in terms of grouping parameters effect

by the long profiles of the negative flex-offer. This fact combined with the large number of aggregated flex-offers that the two greedy techniques produce after aggregation in all the datasets, (Figures 8 and 9) produce a lower balance for start alignment.

The number of the flex-offers that is produced by aggregation influences the result of the absolute balance. The fewer flex-offers participate in the aggregation, the more aggregated are produced. As a result less compensations between

positive and negative slices take place and that leads to a higher absolute balance. Furthermore, start alignment shows a decreasing behavior of the produced absolute balance when the values of the grouping parameters are greater than zero, first row of Figure 9. This occurs because when the values of the grouping parameters increase, more flex-offers participate in the aggregation and thus more positive and negative flex-offers are aggregated, achieving a lower absolute balance.

5.3 Flexibility loss

Another aspect that we examine for the aggregation is the flexibility loss. In Figures 6 and 7 we see that the techniques show a divergent behavior in all the different setups. We see, in the first column of Figure 6, that start alignment has zero absolute balance when TFT=0, followed by exhaustive greedy that has a small difference to simple greedy. The flexibility loss is mainly affected by the time flexibility and since both EST and TFT are set to zero, it means that in each group the flex-offers have the exact same earliest start time and the same time flexibility as well. That leads to no time flexibility loss for start alignment and thus no flexibility loss at all. Furthermore, we notice a low percentage of flexibility loss for exhaustive and simple greedy in the three last setups (second column, second and third row of Figure 6 and first column, second and third row of Figure 7). The low time flexibility of the negative flex-offers of the third and the fourth setups reassures a low flexibility loss. This happens because the solution space is narrowed down, low time flexibility leads to fewer combinations, and that TFT set to 0 reassures that all the flex-offers in the group have the same low time flexibility. A higher percentage of flexibility loss for both greedy techniques is shown in the second setup (second row of Figures 6 and 7), because in this dataset the time flexibility of the flex-offers is higher.

Both exhaustive and simple greedy behave similarly to start alignment, producing almost the same number of aggregated flex-offers (first column of Figure 8). However, we notice a high percentage of the flexibility loss for both greedy and simple greedy in the first setup (first row of Figures 6 and 7). We see that start alignment has, as before, zero flexibility loss, and the nature of the setup favors an exploration of the solution space for both greedy techniques. Eventually, the greedy techniques identify aggregations that lead to less time flexibility and thus to flexibility loss.

Based on the second column of Figure 6, the first and partially the second column of Figure 7, we notice that for all the techniques, when the grouping parameters are increased, the flexibility loss is also increased. This happens because flex-offers with different time flexibilities are in the same group. For start alignment, this will lead to an aggregated flex-offer with the lowest time flexibility and hence to flexibility loss. Regarding exhaustive and simple greedy, larger grouping parameters result in a larger solution space since more flex-offers participate in the aggregation and more earliest start time combinations exist. As a result, the techniques will most probably create an aggregated flex-offer with a lowest absolute balance and a lowest time flexibility. However, no matter how

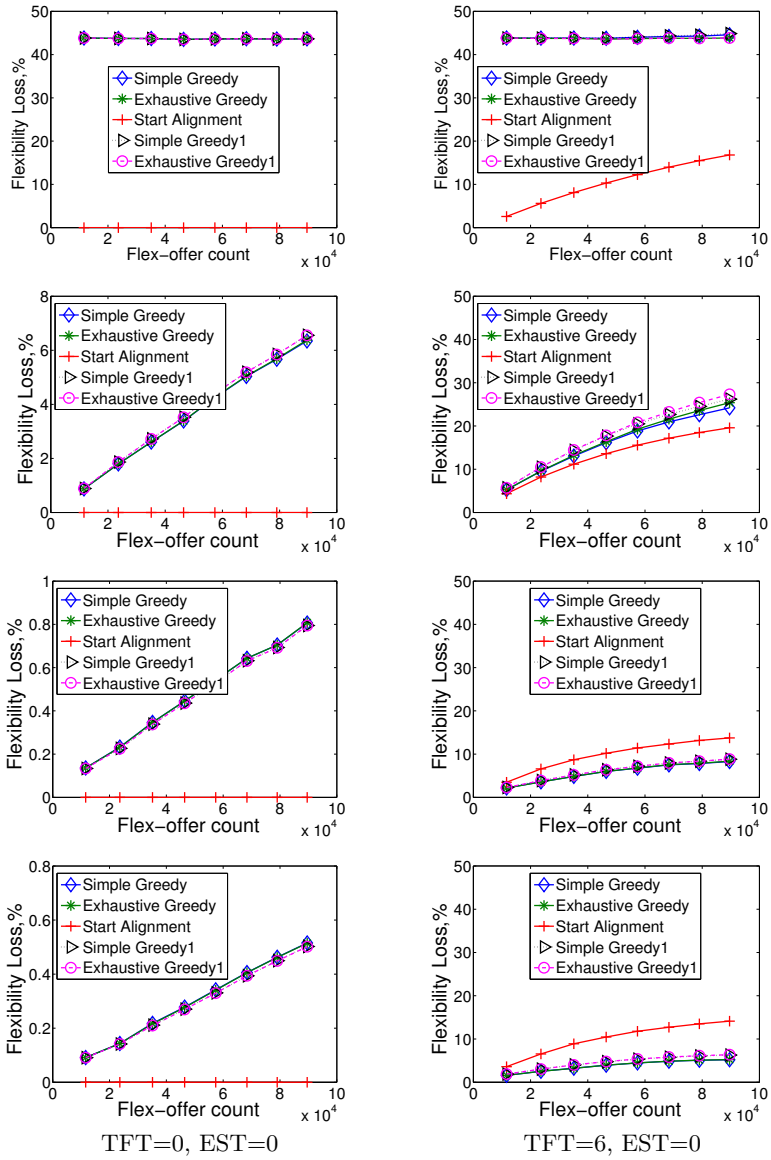


Figure 6. Results of the flexibility loss in terms of scalability effect

much we increase the value of the EST parameter, the fact that TFT is zero will reassure the maintenance of the time flexibility for start alignment and thus no flexibility loss will occur. In almost all the datasets, start alignment shows the best behavior compared to the other two techniques. In the third and the fourth experimental setup, we see that while the number of the flex-offers increases and especially while TFT increases (third and fourth row of Figure 7), the two greedy techniques show a result that is competitive to start alignment result and

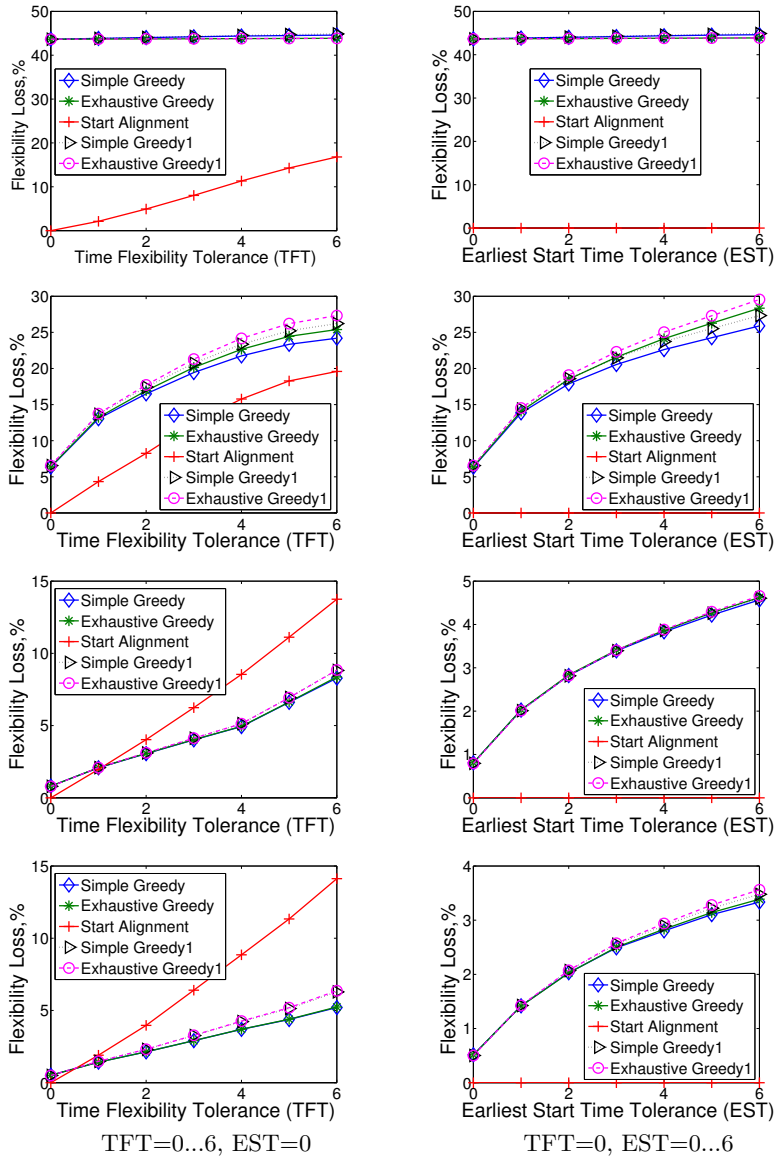
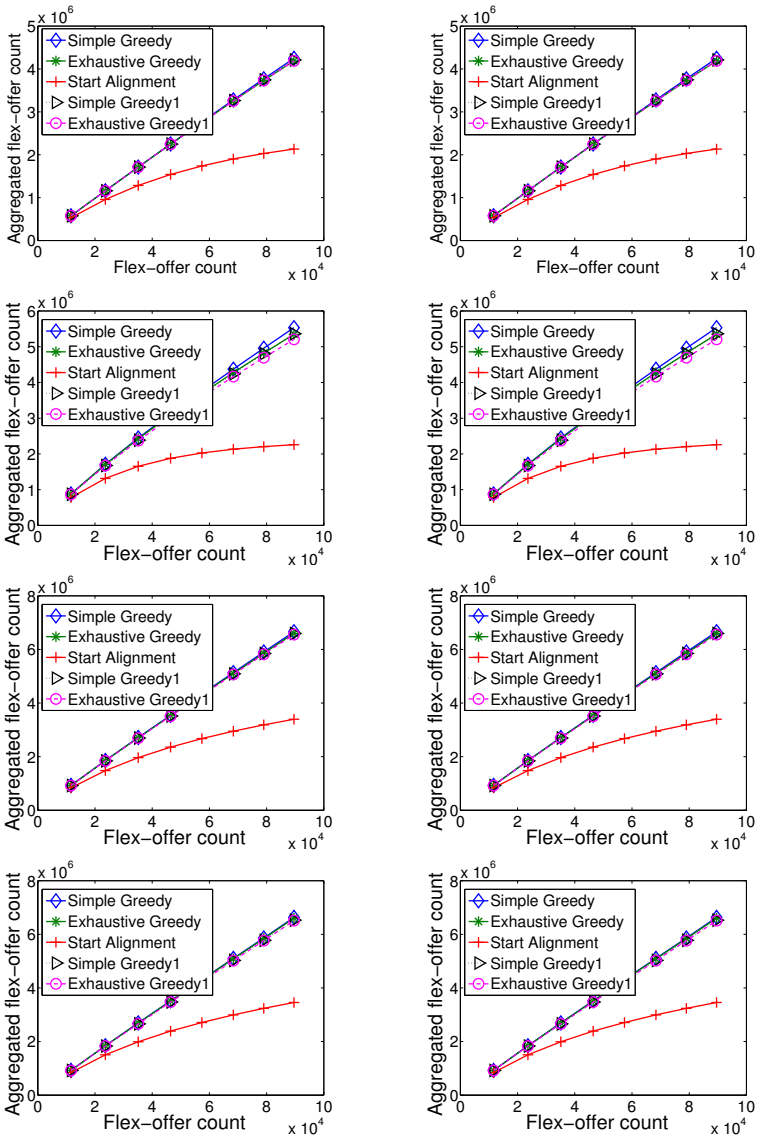


Figure 7. Results of the flexibility loss in terms of grouping parameters effect

even better, achieving a lower flexibility loss. The low flexibility losses occur due to the high value of the time flexibility that the positive flex-offers are characterized with, compared to the negative ones in the third and the fourth setup. Therefore, there are flexibility losses for start alignment, since the aggregated flex-offers have the lowest time flexibility.

The two greedy techniques achieve a lower flexibility loss because some of the flex-offers in the group do not participate in the grouping. Hence, exhaustive and

simple greedy create more aggregated flex-offers than start alignment (Figures 8 and 9), thus fewer flex-offers participate in the aggregation and less flexibility loss will occur.



TFT=0, EST=0 TFT=6, EST=0

Figure 8. Results of the aggregated flex-offer count in terms of scalability effect

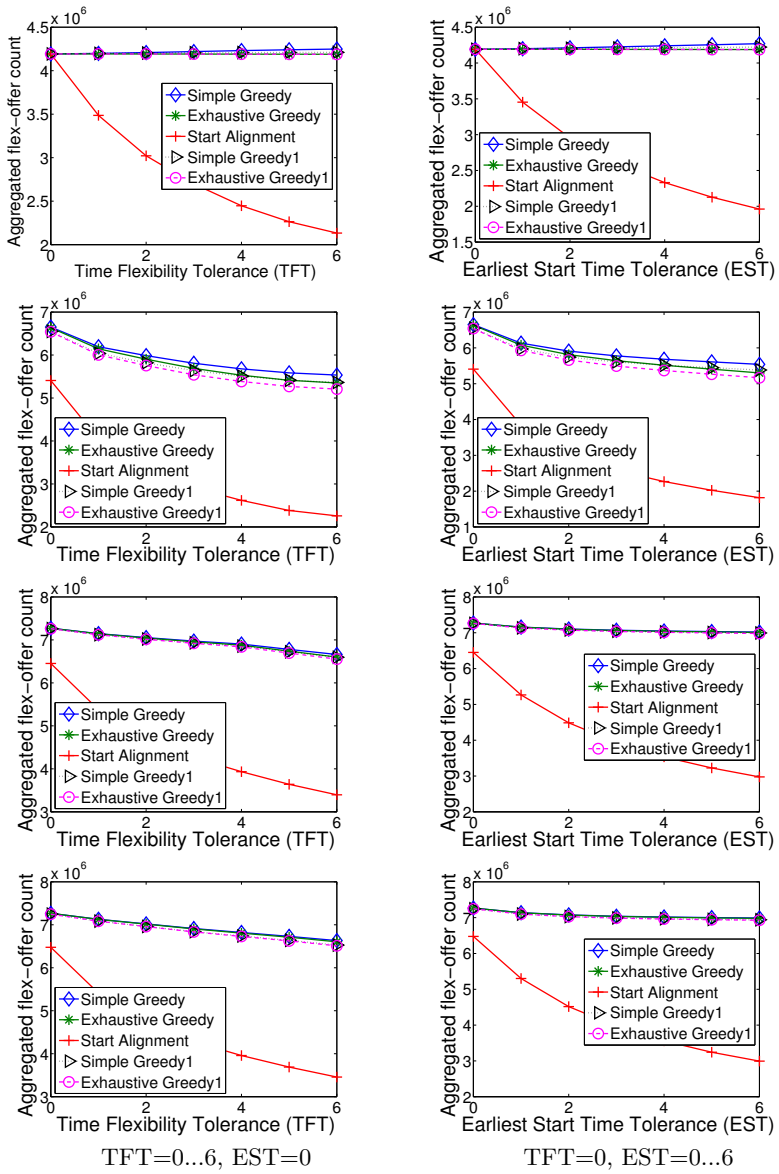


Figure 9. Results of the aggregated flex-offer count in terms of grouping parameters effect

5.4 Execution time and aggregated flex-offers count.

Regarding the processing time of all the techniques, we see in Figures 10 and 11 that start alignment has the best performance followed by simple greedy and exhaustive greedy. Start alignment is the fastest one since it always applies only one aggregation. It is also possible for start alignment to achieve better execution times, see third row, second column of Figure 11, when the grouping

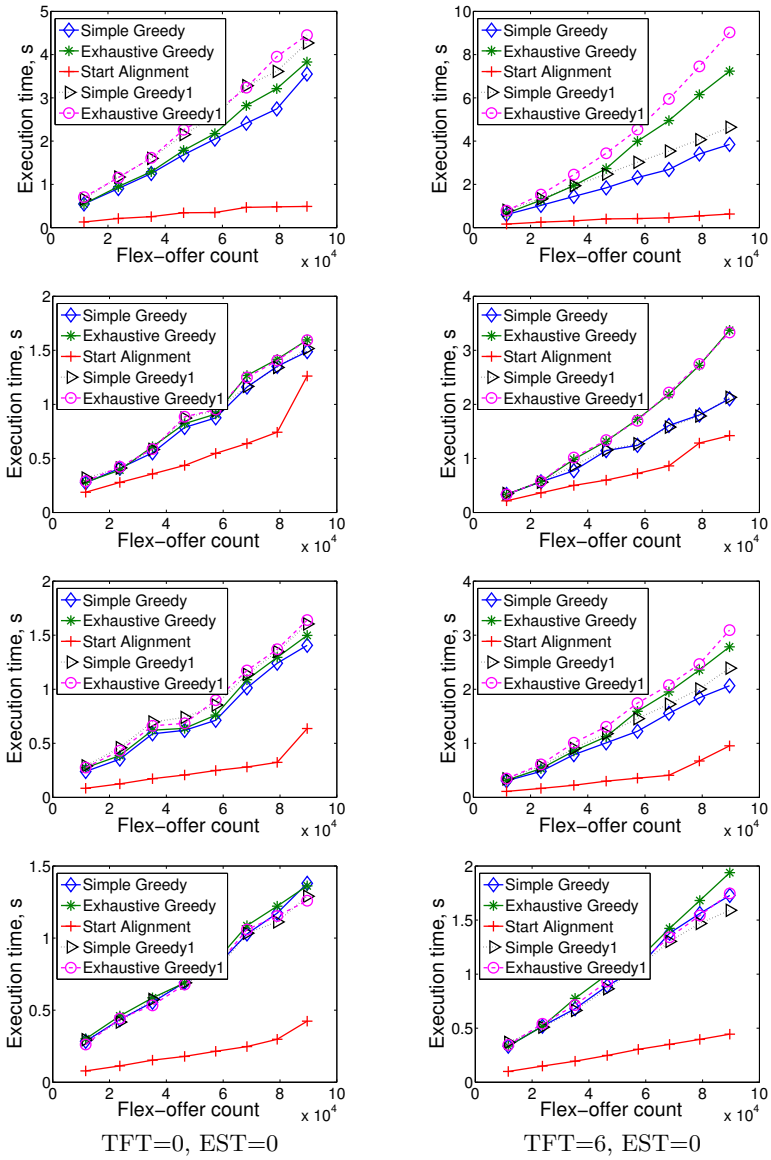


Figure 10. Results of the processing time in terms of scalability effect

parameters are high and thus fewer groups are created. On the other hand, exhaustive greedy demands the most execution time because it creates more than one aggregated flex-offers as simple greedy does, but explores a larger solution space than simple greedy. This results to larger execution times for exhaustive greedy, leaving simple greedy in the second place. Regarding the number of the aggregated flex-offers, we see in Figures 8 and 9 that in all the experimental setups, start alignment has a lower number of aggregated flex-offers than the

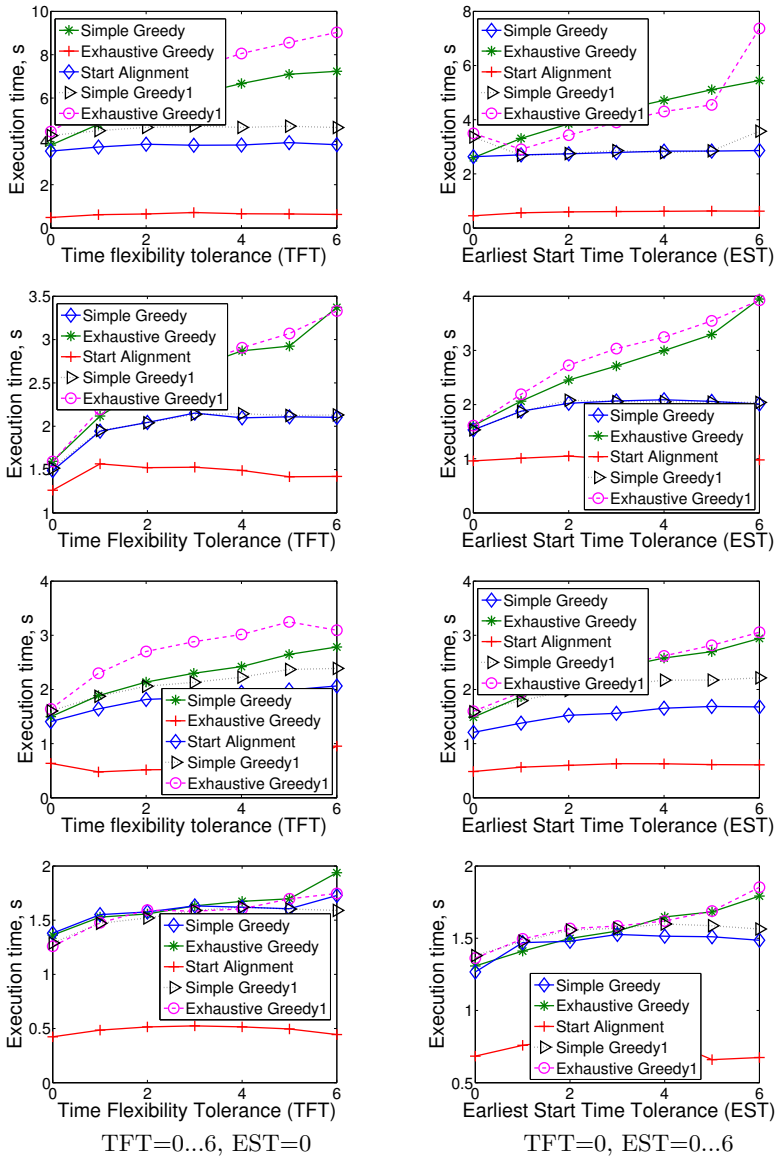


Figure 11. Results of the processing time in terms of grouping parameters effect

two greedy techniques. This is a result of the implementation of the techniques because start alignment will always create one aggregated flex-offer when it is applied to a set of a flex-offers. On the other hand, exhaustive and simple greedy will create at least one aggregated flex-offer if absolute balance is not reduced during the aggregation. Regarding the alternative exhaustive and simple greedy we see no difference for the first experimental setup. However, in all the other setups, the alternate techniques achieve a better absolute balance, higher

flexibility losses, and fewer aggregated flex-offers when the grouping parameters are set to values greater than zero for larger number of flex-offers (second to fourth row of Figures 5, 7, and 9). On the other hand, since they create fewer aggregated flex-offers, they have a bigger solution space to examine and thus they are slower than the original ones (Figures 10 and 11).

6 Conclusion and Future Work

In this paper, we elaborated on aggregation techniques that take into account balancing issues. The techniques discussed in Section 4 reduce the number of the flex-offers that will be the input of the scheduling process and at the same time consider one of its main goals, i.e., achieving balance between energy supply and demand. We conclude through an extensive experimental evaluation that achieving the minimum balance is feasible, but there is always a trade off between balance, flexibility loss and processing time. We show that in order to achieve a good balance, we have to sacrifice time flexibility and also spend more time on processing. The comparisons of the balance techniques with start alignment aggregation showed as well that there are scenarios in which start alignment can achieve very good balance in faster processing times than the greedy techniques. However, flexibility loss between the techniques depends on the grouping parameters without providing a clear winner.

In our future work, we aim to improve the grouping phase that takes place in order to maximize the flexibility that the aggregated flex-offers will have and at the same time improve the balance. It seems also interesting to examine the balance that aggregation will achieve during hierarchical aggregation that is important for an EDMS. In such a scenario, balance aggregation seems more suitable since the input will be mixed flex-offers.

Acknowledgment

This work was supported in part by the TotalFlex project sponsored by the ForskEL program of Energinet.dk.

References

1. Totalflex project, `link:http://www.totalflex.dk/`
2. Bach, B., Wilhelmer, D., Palensky, P.: Smart buildings, smart cities and governing innovation in the new millennium. In: 8th IEEE International Conference on Industrial Informatics (INDIN). pp. 8–14 (2010)
3. Boehm, M., Dannecker, L., Doms, A., Dovgan, E., Filipic, B., Fischer, U., Lehner, W., Pedersen, T.B., Pitarch, Y., Šikšnys, L., Tušar, T.: Data management in the mirabel smart grid system. In: Proc. of EnDM (2012)
4. European Wind Energy Association: Creating the internal energy market in europe. Tech. rep. (2012), `http://www.ewea.org/uploads/tx_err/Internal_energy_market.pdf`

5. Hermanns, H., Wiechmann, H.: Future design challenges for electric energy supply. In: IEEE Conference on Emerging Technologies Factory Automation. pp. 1–8 (2009)
6. Hosseini, S., Khodaei, A., Aminifar, F.: A novel straightforward unit commitment method for large-scale power systems. IEEE Transactions on Power Systems 22(4), 2134–2143 (Nov 2007)
7. Kaulakienė, D., Šikšnys, L., Pitarch, Y.: Towards the automated extraction of flexibilities from electricity time series. In: Proceedings of the Joint EDBT/ICDT 2013 Workshops. pp. 267–272. ACM (2013)
8. Kazarlis, S., Bakirtzis, A., Petridis, V.: A genetic algorithm solution to the unit commitment problem. IEEE Transactions on Power Systems 11(1), 83–92 (1996)
9. Kupzog, F., Roesener, C.: A closer look on load management. In: 5th IEEE International Conference on Industrial Informatics. vol. 2, pp. 1151–1156 (2007)
10. Šikšnys Laurynas, Thomsen, C., Pedersen, T.B.: Mirabel dw: Managing complex energy data in a smart grid. In: DaWaK. pp. 443–457 (2012)
11. Logenthiran, T., Srinivasan, D., Khambadkone, A., Aung, H.N.: Multiagent system for real-time operation of a microgrid in real-time digital simulator. IEEE Transactions on Smart Grid, 3(2), 925–933 (June 2012)
12. Logenthiran, T., Srinivasan, D., Khambadkone, A.M.: Multi-agent system for energy resource scheduling of integrated microgrids in a distributed system. Electric Power Systems Research 81(1), 138 – 148 (2011)
13. Lopes, J., Soares, F., Almeida, P.: Integration of electric vehicles in the electric power system. Proceedings of the IEEE 99(1), 168–183 (2011)
14. Padhy, N.: Unit commitment-a bibliographical survey. IEEE Transactions on Power Systems 19(2), 1196–1205 (May 2004)
15. Rezaee, S., Farjah, E., Khorramdel, B.: Probabilistic analysis of plug-in electric vehicles impact on electrical grid through homes and parking lots. IEEE Transactions on Sustainable Energy, 4(4), 1024–1033 (2013)
16. Srinivasan, D., Chazelas, J.: A priority list-based evolutionary algorithm to solve large scale unit commitment problem. In: International Conference on Power System Technology. vol. 2, pp. 1746–1751 Vol.2 (2004)
17. Tušar, T., Dovgan, E., Filipic, B.: Evolutionary scheduling of flexible offers for balancing electricity supply and demand. In: 2012 IEEE Congress on Evolutionary Computation (CEC). pp. 1–8 (2012)
18. Tušar, T., Šikšnys, L., Pedersen, T.B., Dovgan, E., Filipič, B.: Using aggregation to improve the scheduling of flexible energy offers. International Conference on Bioinspired Optimization Methods and their Applications pp. 347–358 (2012)
19. Šikšnys, L., Khalefa, M.E., Pedersen, T.B.: Aggregating and disaggregating flexibility objects. In: Proceedings of the 24th International Conference on Scientific and Statistical Database Management. pp. 379–396 (2012)