2020 24th International Conference on System Theory, Control and Computing (ICSTCC)

Peak Shaving Algorithms for Residential Consumers. A Comparative Study.

George Ifrim
Department of Automation and Electrical Engineering
Dunarea de Jos University
Galați, Romania
george.ifrim@ugal.ro

Simona Vasilica Oprea
Department of Economic Informatics and Cybernetics
The Bucharest University of Economic Studies
Bucharest, Romania
simona.oprea@csie.ase.ro

Adela Bâra
Department of Economic Informatics and Cybernetics
The Bucharest University of Economic Studies
Bucharest, Romania
bara.adela@ie.ase.ro

***Abstract*—The aim of this paper is to compare the day-ahead electricity consumption optimization algorithms for smart houses with the widely used linear programming techniques. Both algorithms are designed to work on real data acquired from a small residential community with 11 houses and are validated in simulation. The main purpose of both optimization algorithms is peak shaving, meaning the flattening of the electricity consumption vector. The flattening capacity of both algorithms is evaluated through two indexes, namely the flattening index and the peak-to-average ratio. Whereas the peak shaving is beneficial for the energy producer and providers, it can also bring savings to the consumers if specific pricing plans are imposed. The pricing plans assume expensive energy at peak hours and cheap energy at off-peak hours. Both algorithms were also compared in terms of savings.**

Keywords—linear programming, peak shaving, flattening index, constraints, electricity consumption vector

I. INTRODUCTION

The demand-side management strategies bring benefits to many parties, from producers to consumers [1]. These strategies are investigated by many governments to promote energy efficiency plans. However, the demand-side management strategies do not target the decrease of the consumed electricity, but the modification of the customer's behavior. The effect of these strategies is the peak shaving of the electricity consumption vector that can be achieved either with the direct implication of the customer, if he voluntarily reschedules the operation of some appliances, or through dynamic demand response, when the provider is allowed to control customer's appliances and to reschedule their operation at off-peak hours. The implementation of such strategies should not interfere with the comfort or intimacy of the customer and they should bring benefits to both sides.

The potential of implementing such strategies derives from the progress of smart appliances and sensors that are provided with batteries. Many appliances have now the capacity to be programmed individually or through automation hubs. The future appliances will be able to connect by default to an entire universe of devices, the internet of things.

The demand-side management strategies, together with energy efficiency plans, should result in a set of actions that lead to a responsible usage of the energy resource. Peak shaving helps the energy producer to avoid the use of alternative fuels, such as the methane gas, to produce more energy at peak hours. The flattening of the electricity consumption vector also brings benefits to the network administrator which is not forced to extend the grid and to the energy provider who will not be obliged to buy expensive energy at peak hours. The provider, on its turn, could encourage the consumer to change its consumption behavior by imposing different pricing plans. The result of not consuming energy at peak hours is noticeable savings on the electricity bill.

The optimization of the energy consumption is a topic widely researched by many groups. The variety of optimization strategies and methods derives from the numerous types of appliances and costumers. The optimization techniques for peak shaving are based both on models [2] and data [3, 4]. The peak shaving strategies involve the reschedule of some appliances [3, 4], but they can also integrate various storage devices, such as batteries [5], or use locally generated energy such as solar or wind energy [6].

Among the optimization methods, the linear programming is widely used due to the linear nature of the optimization problem [2, 7, 8]. Some advantages of the linear programming techniques are the simple formulation of the objective function, the ability to easily implement various constraints and the capacity to work with integer numbers. Solvers for linear programming problems are implemented in many programming languages (e.g. Matlab, Python etc.).

The paper presents a fair comparison between a recently published optimization method, the day-ahead optimization algorithm [4] and the common linear programming methods. The paper presents the two algorithms that were implemented in Matlab and their validation on 24-hours datasets that were acquired from a small residential community.

II. THE OPTIMIZATION ALGORITHMS

The aim of this study is to compare the new day-ahead electricity consumption optimization algorithms, recently published by [4], with conventional techniques, such as linear programming. The algorithms were designed to flatten the electricity consumption vector by rescheduling/moving the consumption of the programmable appliances. The appliances that are considered programmable are the ones that are not interfering with the comfort of the customer and can be classified into interruptible and non-interruptible appliances. The day-ahead optimization algorithms are based on data acquired from the customers, either as a schedule sent to the provider or through smart meters and sensors. Peak shaving is beneficial for the provider because it will not be forced to buy expensive energy at peak hours and for the producer since it will not be determined to use alternative ways to produce energy (e.g. methane gas).

To encourage consumers to change their consumption behavior, the producer can impose various Time-of-Use (ToU) plans which involve higher cost for peak energy and noticeable lower rates for off peak energy. The optimization algorithms for the demand-side management strategies have thus advantages for the producer (peak shaving) and for the consumer (reduced electricity cost).

The day-ahead optimization algorithms were validated on real data acquired from a small residential community [4] and, to have a fair comparison, the linear programming optimization algorithm was designed to use the same data. Due to some limitation of the later, only the interruptible programmable appliances were considered here.

A. Input Data

The data employed for validating in simulation the optimization algorithms are real data acquired by smart meters from a small community of 11 residential houses. The resolution with which the data was acquired varies from 1 second to 1 hour and, for this reason, hourly averages were considered. The optimization algorithms were validated and compared on 24-hour datasets. The 11 houses, referred to as Home A to K, gather 311 appliances, of which 229 non-programmable appliances, 17 non-interruptible programmable appliances (not considered in the following simulations) and 65 interruptible programmable appliances.

B. Assessment Tools for the Evaluation of the Optimization Algorithms

The evaluation of the flattening capacity of the optimization algorithms can be done by calculating the flattening index (FI) which is the ratio between the average daily electrical consumption and the maximum daily consumption:

$$FI = \frac{Average\ Consumption}{Peak\ Consumption}$$

A flat vector will have the flattening index equal to one, while a higher amplitude of the vector will result in a lower index.

Another utilized assessment tool is the Peak-to-Average (PAR) index which is the ratio between the squared daily consumption peak and the squared daily consumption average:

$$PAR = \frac{Squared\ Peak\ Consumption}{Squared\ Average\ Consumption}$$

The PAR index is equal to one when a vector is flat but reaches higher values when the amplitude of the vector is higher than 0.

Besides flattening, the algorithms can bring the reduction of the electricity bill for the consumer when specific ToU plans are employed. In many cases, providers practice a standard tariff when no flattening is required. With a standard tariff, savings on the electricity bill are very hard to achieve by the consumer and can come, in fact, with additional costs (e.g. buying thermostats, new energy efficient appliances, automation components etc.). The real price for genuinely saving energy is many times higher than the electricity bill reduction. However, if flattening is required, the provider can impose various ToU tariff plans to encourage the consumer to avoid the peak hours, when the electricity is more expensive. The degree of implementation depends on the technological equipment owned by the consumer, that can go from willingly avoiding operating certain devices at peak hours up to using home automation hubs to program the operation of the appliances (e.g. dehumidifiers, ventilation devices, heating systems, electrical car etc.).

Four ToU tariff plans were considered for the simulations in this study as it can be found in Table 1 [9]. According to the behavior of the consumer, the provider could recommend a mild plan or a more extreme one for higher savings.

TABLE I. THE TOU TARIFF RATES

ToU tariffs	Night rate From 23:00 to 07:59, Monday to Sunday [cents/kWh]	Day rate From 08:00 to 16:59 and from 19:00 to 22:59, Monday to Sunday; from 17.00 to 18:59 Saturday, Sunday and on public holidays [cents/kWh]	Peak rate From 17:00 to 18:59, Monday to Friday without the public holidays [cents/kWh]
Tariff A	12	14	20
Tariff B	11	13.5	26
Tariff C	10	13	32
Tariff D	9	12.5	38

The four ToU tariff plans are compared with a standard tariff of 14.1 cents/kWh.

C. Day-Ahead Optimization Algorithm for Interruptible Programmable Appliances

The day-ahead optimization algorithm was programmed in Matlab and simulated on 24-hour datasets. A detailed description, along with simulations on 24-hour and one-year datasets can be found in [4].

Each house (i.e. A to K) has an individual database from where the electrical consumption data are extracted. The result of this process consists in two matrices:

- The vector with the total consumption of non-programmable appliances, P (short from $P^h, h = \overline{0,23}$, h - hour). The consumption of all 229 non-programmable appliances is summed in a 24 x 1 vector. Because they cannot be rescheduled their individual consumption is not important.
- The matrix with the individual consumption of the interruptible programable appliance, SI (short from $SI_i^h, h = \overline{0,23}, i = \overline{1,n}$, n – total number of interruptible programmable appliances), with the dimension of 24 x 65.

The two matrices were saved as .mat files because it takes considerably less to load. The extraction procedure is separated thus from the optimization algorithm who can take more than ten times less to run.

The interruptible programable appliances are not allowed to operate any time and thus a matrix of state constraint, ν (short from $\nu_i^h \in \{0,1\}$), with the same dimension as SI, is defined. The elements of the state constraint matrix are 0 – appliance not allowed to operate and 1 – the operation is allowed. To create more reliable simulations, approx. 20% restricted hours (~5 restricted hours per day), between 8:00 and 22:59, were imposed in ν. The restricted hours were chosen randomly because there is no technical information on the programmable appliances. In a real process the restrictions are defined and agreed by the consumer according to his preferences (e.g. loud appliances are not allowed to operate at night hours).

The optimization algorithm is straightforward, it shifts the consumption of the programable appliance (if necessary) at the hours with the lowest electrical consumption. To do this, the following steps must be completed:

1. *Define the input block*. The input block consists in loading P, SI and ν from .mat files.

2. *Find appliance consumption*. The hour(s) when an appliance registers electrical consumption must be found in order to reschedule them. A threshold, th, can be imposed here to restrict the appliances with very low consumption from being optimized $(SI_i^h > th)$. Shifting appliances with very low consumption may result in interfering with costumer's behavior without any result in terms of flattening and savings.

3. *Sort P*. The vector with the total consumption of non-programable appliances, P , must be sorted in ascending order to find the indexes of the off-peak hours. The consumption found at the previous step will be rescheduled at the beginning of the sorted P vector.

4. *Verify* ν. The reschedule is possible only if the matrix of state constraint allows the operation of the appliance. If the operation is restricted, the next off-peak hours are considered.

5. *Calculate the amplitude*. The consumption of each rescheduled appliance is added to P, resulting in a new matrix PO_i^h of 24 x n. The amplitude of each column is calculated, resulting in a new vector of amplitudes, PB_i of 1 x n: $PB_i = max(PO_i^h) - min(PO_i^h)$.

6. *Select the minimal amplitude*. The lowest amplitude is selected, $min(PB_i)$. The appliance that reduces the amplitude of the total consumption vector the most is optimized first.

   The day-ahead optimization algorithm for the programmable appliances was designed to work with four objective functions:

   - Amplitude minimization: $PB_i = max(PO_i^h) - min(PO_i^h)$, select $min(PB_i)$,
   - Min/max consumption: $PB_i = min(PO_i^h)$, select $max(PB_i)$,
   - Max/min consumption: $PB_i = max(PO_i^h)$, select $min(PB_i)$,
   - Dispersion minimization: $PB_i = \sum(PO_i^h)^2$, select $min(PB_i)$.

   Any of the four objective functions give similar results as it was described in [4].

7. *Calculate the new P*. After one appliance was optimized, its consumption is added to the vector P which becomes the new vector P that will be used in the next iteration. The optimized appliance is deleted from SI_i^h, the total number of programable appliances for the second iteration being $n - 1$.

8. *Return to step 2*. The procedure is repeated until all devices are optimized, SI is empty.

For further insight the hourly averages of the electrical consumption of all non-programmable and programmable appliances in the residential community is displayed in Figure 1. The selected 24-hour dataset corresponds to a summer day that, as it can be seen, presents two peaks: a bigger one in the evening (around 19:00) and a smaller one in the morning. The non-programmable appliances are annotated with N and their hourly average electrical consumption is represented by the blue bars. The interruptible programmable appliances are annotated with SI (orange bars). The objective of the optimization algorithm is to reschedule the electrical consumption of the programable appliances in the peak hours (between 17:00 and 21:00) to off-peak hours. The advantage of the day-ahead optimization algorithm is that it doesn't require any technical data about the appliance that is optimized (i.e. type, minimal/maximal consumption etc.). For increased intimacy, the consumer can even connect the appliance to a smart plug.

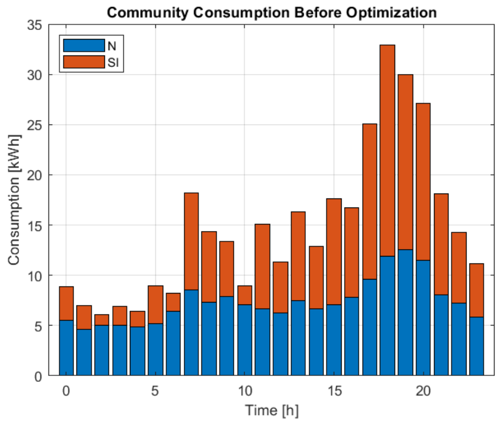

Fig. 1. The electrical consumption of the non-programmable and programmable appliances in a small community before optimization

For a better visualization of the flattening capacity of the day-ahead optimization algorithm, Figures 2 and 3 show one iteration of the algorithm which means the optimization of one programmable appliance. Figure 2 presents the total electrical consumption of the non-programmable appliances and the consumption of one programmable appliance, namely an electric car. The car is plugged-in at peak hours, probably when the owner returns home after work, when the energy is the most expensive. The car is obviously not used in this time frame and thus the recharge can be rescheduled at off-peak hours.

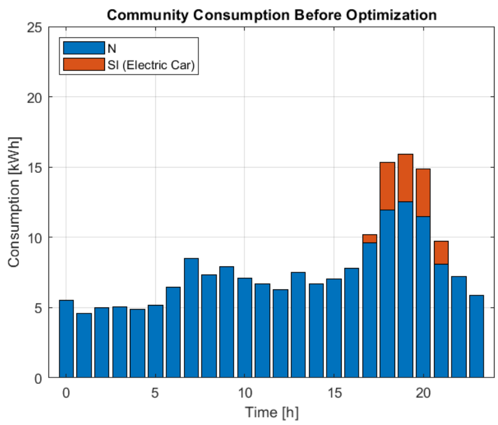

Fig. 2. The electrical consumption of the non-programmable appliances in a small community and one programmable appliance before optimization

Figure 2 shows the rescheduled consumption of the electrical car at the off-peak hours. The matrix of state constraint is a good tool to design the rescheduling according to customer behavior. Knowing in this case that the appliance is an electric car, we imposed the following restrictions: $v_{el.car} = [1\,1\,1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1]$. Most probably the car is used during the day and therefore the time interval 6:00 – 17:00 is restricted for optimization, leaving the recharge for late evening and night hours where the energy is cheaper.

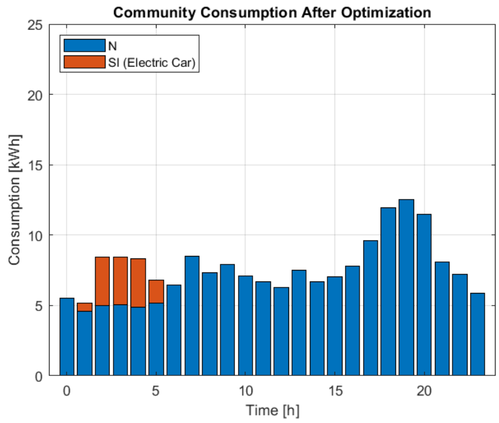

Fig. 3. The electrical consumption of the non-programmable appliances in a small community and one programmable appliance after optimization with the day-ahead optimization algorithm

Rescheduling the consumption from peak hours to off-peak hours flattens the consumption vector from an initial $FI = 0.49$ to $FI = 0.63$ after the optimization of only one appliance. In terms of PAR, the index decreases from 4.11 to 2.53. The consumption vector will be flatter with the optimization of each appliance. The rescheduled consumption of the electrical car will be added to the total consumption of the non-programmable appliances, which will become the mew P matrix for the second iteration of the algorithm.

D. Linear Programming Algorithm for Interruptible Programmable Appliances.

Linear programming (LP) is an optimization approach that is widely used in a variety of fields (e.g. engineering, economics, energetics etc.). The linear programming techniques optimize linear objective functions that are subjected to linear constraints:

$$\min_x f(x) = c^T x \text{ such that} \begin{cases} A \cdot x \le b \\ A_{eq} \cdot x = b_{eq} \\ lb \le x \le ub \end{cases} \quad (3)$$

where f is the linear objective function, $A \cdot x \le b$ are inequality constraints, $A_{eq} \cdot x = b_{eq}$ are equality constraints and lb and ub are the lower and upper bounds. f, x, b, b_{eq}, lb and ub are vectors, and A and A_{eq} are matrices.

LP problems can admit a large number of variables and yet their solving requires a low number of iterations. The result is a feasible value of the objective function or a feasible region if the function is not specified. Linear programming solvers are implemented in many programming languages, Matlab using the `linprog` function to solve linear optimization problems. The linear programming solver allows the user to choose from the following algorithms: dual simplex, interior point and interior point legacy.

Even though these optimization techniques are already implemented in Matlab, the problem must be carefully formulated in order to have a fair comparison with the day-ahead optimization algorithm.

When formulating the optimization problem, the 24 hours were considered dimensions of the objective function. Thus, the multidimensional objective function has the following expression:

$$f(x) = c_1 x_1 + c_2 x_2 + \cdots + c_{24} x_{24} \quad (4)$$

The first dimension, x_1, is the hour 0, x_2 is 1 and so on. In Matlab, the vector f will contain the coefficients, c, of x (i.e. a vector of ones). Calling the `linprog` solver once, means the optimization of one appliance. The solver must be thus added in a repetitive structure and called as many times as the number of appliances. The programmable appliances will be rescheduled one by one, in the order they are in the SI matrix. If necessary, a preliminary sorting of the appliances can be done (e.g. when an appliance must operate before another appliance).

The rescheduling of a programmable appliance must be done considering the total consumption of the non-programmable appliances, vector P, thus the lower bound, lb, will be their electrical consumption (i.e. $lb = P$). There is no need for an upper bound because other constrains will be imposed.

Before considering an appliance to be rescheduled it is important to verify if it consumes during the day. A threshold, th, can be also imposed to optimize only the appliances with higher consume for a bigger impact over the flattening and savings.

The rescheduled consumption of a programmable appliance is added to the total consumption of the non-programmable appliance. The obtained vector becomes the new lb for the second iteration and so on. To avoid building up new peaks, inequality constraints must be set. When the set value is reached (b – Eq. 3), the electrical consumption is scheduled to the next available hour. Each iteration (each call of `linprog` solver) must have a higher imposed value to avoid stopping the solver for not finding a feasible value and to be strict enough not to create new peaks. A good practice is to base the inequality constraint on the average consumption that increases as more appliances are optimized (and added to lb). A small constant value could be added to the average consumption to make sure the solver is not very restricted: $b^h = \overline{P + SI_i^h} + ct$. Using the average consumption is possible only if lb is lower, otherwise no consumption will be added to that hour and the inequality constraint, b, will be the actual value of lb (i.e. $b = P^h$). The maximum consumption of each hour must be restricted, thus A is the identity matrix.

The total consumption of one programmable appliance, $\sum_{h=1}^{24} SI_i^h$, is added to the total consumption of the non-programmable appliances, $\sum_{h=1}^{24} P^h$, and is implemented through the equality constraints. Thus, $b_{eq} = \sum_{h=1}^{24} SI_i^h + \sum_{h=1}^{24} P^h$ and A_{eq} is an all-ones vector with 24 elements. This would be accurate only if the device is allowed to operate any time. If the matrix of state constraint is imposed, $A_{eq}^h = v_i^h$

and b_{eq} will sum only the allowed operating hours, $b_{eq} = \sum_{h=1}^{24} SI_i^h + \sum_{h=1}^{24} P^{logical(A_{eq}^h)}$. The `exitflag` of `linprog` solver can be used to verify that it converged to a solution, otherwise the consumption of the programmable device remains unchanged.

The linear programming algorithm reschedule the consumption of the programmable appliances (if necessary) at hours that are feasible and not necessarily with low consumption. The constraints are in charge of limiting the solution to off-peak hours.

The biggest difference between the day-ahead optimization algorithm and the linear programming algorithms is that the consumption of a programmable appliance is changed when rescheduled. This would require knowledge about the technical specification of the appliance in terms of maximal hourly consumption in order to add new constraints. The linear programming algorithm requires the following steps:

1. *Define the input block*. The input block consists in loading P, SI and v from .mat files.
2. *Find appliance consumption*. Similarly with the previous algorithm, a threshold, th, is imposed to reschedule only the appliances with higher consumption.
3. *Define constraints*. The inequality and equality constraints are defined as described above, along with the lower bound.
4. *Call* `linprog` *solver*. The solver is called in Matlab, requesting at least the value of x and the exitflag. x is in fact the rescheduled SI_i^h.
5. *Calculate the new P*. The consumption of the rescheduled programmable appliance is added to P. The sum will be the new P for the next iteration. Similarly, the number of programmable appliances for the second call is $n-1$, because the optimized appliance is deleted from SI_i^h.
6. *Return to step 2*. The procedure is repeated until all devices are optimized, namely SI is empty.

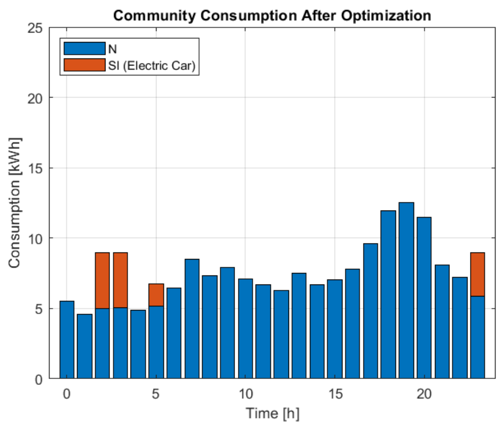

Fig. 4. The electrical consumption of the non-programmable appliances in a small community and one programmable appliance before optimization. Linear Programming, Dual-Simplex Algorithm

Figures 4 and 5 display the flattening capacity the linear programming algorithms (compare to Figure 2 before optimization). Figure 4 uses the default dual simplex algorithm, while Figure 5 shows the interior point algorithm. It is hard to decide which one is the best without having actual restrictions on the maximal consumption of the programmable appliance.

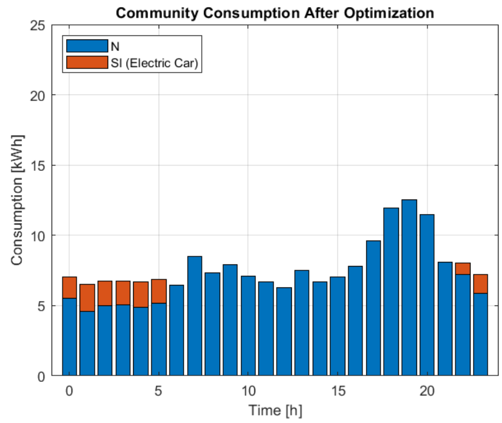

Fig. 5. The electrical consumption of the non-programmable appliances in a small community and one programmable appliance before optimization. Linear Programming, Interior Point Algorithm

In terms of flattening, all algorithms (linear programming and day-ahead optimization) gives the same results on the electric car optimization.

III. Simulations and Results on a Small Residential Community

All algorithms presented above were validated on the same data acquired from 11 residential houses with 229 non-programmable appliances and 65 interruptible programmable appliances. The results are presented graphically (Figure 6 and 7), the algorithms being assessed both in terms of flattening the consumption curve and savings on the electricity bill. 24-hour datasets were used for the validation of the two algorithms. The flattening capacity of the day-ahead optimization algorithm is displayed graphically in Figure 6 (compare with Figure 1) and quantified through FI and PAR. The original consumption vector has a FI of 0.45 and the rescheduling of the programable appliances increases the FI to 0.84. The PAR decreases from 4.93 to 1.42.

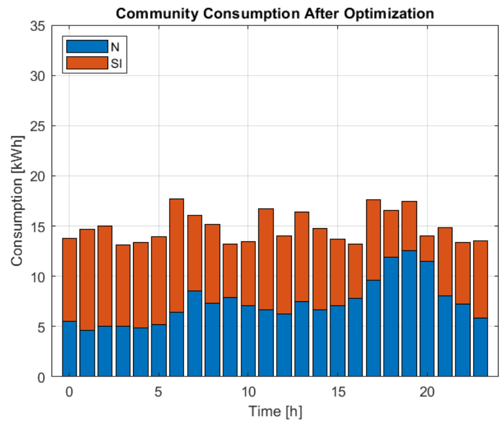

Fig. 6. The electrical consumption of the non-programmable and programmable appliances in a small community after optimization with the day-ahead algorithm

All four objective functions presented above (i.e. amplitude, min/max, max/min and dispersion) produce very similar flattening effect, as described in [4].

The ToU tariffs presented in Table 1 were applied on the 24-hour datasets, before and after optimization, to evaluate the possible savings with the electricity. The biggest savings, 13.8 %, are obtained if an extreme ToU tariff (i.e. D) is imposed. If the consumption is not optimized, the energy is more expensive when compared to the standard tariff of 14.1 cents/kWh. It should be noted that savings are brought with any imposed ToU tariff and that unoptimized consumption is more expensive with any pricing scheme.

TABLE II. COMMUNITY PAYMENT ON A 24-HOUR DATASET BASED ON THE DAY-AHEAD OPTIMIZATION ALGORITHM

	STD Tariff	ToU Tariff A	ToU Tariff B	ToU Tariff C	ToU Tariff D
Payment initial [€/day]	50.16	51.65	53.23	54.82	56.40
Payment shifted [€/day]	50.16	49.24	49.03	48.82	48.62
Gain ToU shifted [%]	0.00	4.67	7.89	10.93	13.80
Gain ToU/STD initial [%]	0.00	-2.97	-6.12	-9.27	-12.42
Gain ToU/STD shifted [%]	0.00	1.84	2.26	2.67	3.09

The dual-simplex linear programming algorithm can produce an even flatter consumption vector with a FI of 0.93 and a PAR of 1.16 (Figure 7). This is possible because the linear programming algorithm can reschedule a fraction of the original hourly consumption. In simulation the results are very good, but in practice this would require more knowledge on the programmable appliance (i.e. type, min/max consumption etc.) which will lead to more restrictions and a lower FI. However, any FI that is higher than 0.7 is convenient for the provider. The results are comparable with the day-ahead optimization algorithm, both in term of peak shaving and savings on the electricity bill. Due to a flatter consumption vector, the savings can be slightly higher than the day-ahead optimization algorithm and can go up to 16% for the extreme ToU tariff plans.

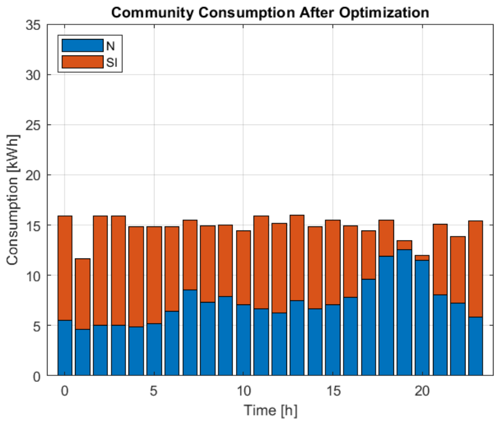

Fig. 7. The electrical consumption of the non-programmable and programmable appliances in a small community after optimization with the linear programming solver (dual-simplex algorithm)

TABLE III. COMMUNITY PAYMENT ON A 24-HOUR DATASET BASED ON LINEAR PROGRAMMING WITH THE

	STD Tariff	ToU Tariff A	ToU Tariff B	ToU Tariff C	ToU Tariff D
Payment initial [€/day]	50.16	51.65	53.23	54.82	56.40
Payment shifted [€/day]	50.16	48.91	48.41	47.90	47.39
Gain ToU shifted [%]	0.00	5.31	9.07	12.62	15.96
Gain ToU/STD initial [%]	0.00	-2.97	-6.12	-9.27	-12.42
Gain ToU/STD shifted [%]	0.00	2.50	3.51	4.52	5.52

In what regards the execution time, both algorithms require similar running times. They were executed several times and the simulation time was evaluated with the `tic toc` function in Matlab

IV. CONCLUSIONS

The present paper is a comparison between two peak shaving algorithms, the new day-ahead optimization algorithm and the widely used linear programming techniques. Both algorithms give comparable results in terms of flattening, the first one gives a FI of 0.84, while the second a FI of 0.93. The two algorithms are also compared in terms of savings based on four ToU tariff plans. The day ahead optimization algorithm can bring up to 14% savings while the linear programming can go up to 16% due to a slightly flatter consumption vector. The better results of the linear programming algorithm derive from the fact that it is allowed to reschedule a fraction of the hourly consumption which would require more data from the customer regarding the programmed appliance. In addition, dealing with the technical specifications of the programmable appliances would require the implementation of several new constraints.

ACKNOWLEDGMENT

This paper presents the scientific results of the project “Intelligent system for trading on wholesale electricity market” (SMARTRADE), co-financed by the European Regional Development Fund (ERDF), through the Competitiveness Operational Programme (COP) 2014-2020, priority axis 1 – Research, technological development and innovation (RD&I) to support economic competitiveness and business development, Action 1.1.4 - Attracting high-level personnel from abroad in order to enhance the RD capacity, contract ID P_37_418, no. 62/05.09.2016, beneficiary: The Bucharest University of Economic Studies.

REFERENCES

[1] A. Pina, C. Silva, and P. Ferrão, “The impact of demand side management strategies in the penetration of renewable electricity,” Energy, vol. 41, pp. 128-137, 2012.

[2] Z-k. Feng, W-j. Niu, W-c. W, J-z. Zhou, and C-t Cheng, “A mixed integer linear programming model for unit commitment of thermal plants with peak shaving operation aspect in regional power grid lack of flexible hydropower energy,” Energy, vol. 175, pp. 618-629, 2019.

[3] S.V. Oprea, A. Bara, and G. Ifrim, “Flattening the electricity consumption peak and reducing the electricity payment for residential consumers in the context of smart grid by means of shifting optimization algorithm,” Comput. Ind. Eng., vol. 122, pp. 125-139, 2018.

[4] S.V. Oprea, A. Bâra, G.A. Ifrim, and L. Coroianu, “Day-ahead electricity consumption optimization algorithms for smart homes,” Comput. Ind. Eng., vol. 135, pp. 382-401, 2019.

[5] U. Prasatsap, S. Kiravittaya, and J. Polprasert, “Determination of Optimal Energy Storage System for Peak Shaving to Reduce Electricity Cost in a University,” Energy Procedia, vol. 138, pp. 967-972, 2017.

[6] G. Hafeez, R. Khalid, A.W. Khan, M.A. Judge, Z. Iqbal, R. Bukhsh, and A. Khan, “Optimal Residential Load Scheduling Under Utility and Rooftop Photovoltaic Units,” In: F. Xhafa, S. Caballé and L. Barolli (eds), Advances on P2P, Parallel, Grid, Cloud and Internet Computing, Springer, vol 13, pp. 142-153, 2018.

[7] M. Moghimi, D. Leskarac, C. Panchal, J. Lu, and S. Stegen, “Energy Management System for Peak Shaving in an Experimental Microgrid Employing MILP Optimization,” Int. J. Eng. Inov. Technol., vol. 7(7), pp. 28-33, 2018.

[8] G. Zakeri, D. Craigie, A. Philpott, and M. Toddc, “Optimization of demand response through peak shaving,” Oper. Res. Lett., vol. 42, pp. 97-101, 2014.

[9] ISSDA Irish Social Science Data Archive. Available online: http://www.ucd.ie/issda/data/commissionforenergyregulationcer (accessed on 5 June 2017).