Simple Day-Ahead Bidding Algorithm for a System with Microgrids and a Distributor

Leonardo Jelenković, Toni Budrović
Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
Email: leonardo.jelenkovic@fer.hr

Abstract—In this paper we present a heuristic distributed algorithm for day-ahead bidding between collection of microgrids that sell their surplus energy to a power grid controller – distributor, who can use that energy with its consumers. Each microgrid is modeled as a node with consumers, energy generator (from renewable sources) and energy storage capabilities. For each time period, distributor negotiate amounts and prices with each microgrid separately, in rounds. Although both sides are modeled, we focus our research on the microgrid’s bidding algorithm for monetary gain from sold surplus energy. Presented algorithm is simple and produces good results from microgrid perspective. Due to model simplicity, produced results aren’t always optimal, and convergence had to be additionally helped. In comparison to similar distributed bidding algorithms for systems with storage capacity presented algorithm is simplest and can be used for day-ahead but with presented adaptations also for more complex microgrids and distributor, real-time adjustments and asynchronous negotiations.

I. INTRODUCTION

Using renewable energy sources is the only long term strategy that will ensure energy stability. Therefore, today we put a lot of effort to identify renewable energy sources and use them. Such researches are being promoted and widely supported by governments and organizations, as in [1]. In electric power systems, on global scale, power plants that use fossil fuels should be complemented or replaced with ones that use renewable energy. On a smaller scale, renewable energy could be collected with small power plants (e.g. wind turbines, solar panels) and used locally, in areas called microgrids, for its own consumers.

However, renewable energy production might be very variable even over single day. In some periods (hours) produced energy might be sufficient or insufficient for local consumers. Therefore, additional energy sources might be required. Such sources may include battery or other type of energy storage, diesel generator (or similar), or/and connection to a global power grid. Those sources can be used when renewable sources don’t produce sufficient amount of energy for local microgrid. On the other side, when renewable sources produce more energy than its locally needed, surplus can be used to charge local storage or/and it can be sold through global power grid.

Microgrids will not significantly influence global power grid (or smart grid), at least not in near future since their surplus energy is not significant when comparing with power grid needs. However, since they use renewable energy, microgrids might be supported by governments. In this paper we assume a model (Fig. 1) where a power grid controller – distributor (in rest of the paper), possibly under influence of government, should offer to buy some of microgrids’ surplus energy (up to predefined amount for each period of day). That’s where a trade market could be created – between a distributor and connected microgrids. To optimize usage of energy bought from microgrids, distributor defines different price ranges for different periods of day ahead (e.g. highest prices for peak demand periods) giving incentive to microgrids to adjust their selling strategy, to provide more energy in high-price periods. In this paper we focus on microgrid’s control method that will maximize profit from sold surplus energy. Storage management influence microgrid’s offers to distributor, and distributor’s prices influence microgrid’s storage management. This is where ‘the game’ is defined.

Similar problems were being addressed in literature. In [2] and [3] smart grid is simulated consisting of users (consumers) and suppliers (generators). Generator models include thermal and hydro generators. Thermal generators are modeled with: maximal power, minimum stable load, ramps and variable, no-load and startup costs. Hydro generators are modeled with maximal and minimal power outputs and daily energy reserves. Demand side was modeled as static, inelastic – does not changes with price. In [2] bidding is performing iteratively, decrementing price in each iteration, while in [3] price was chosen at intersection of supply and demand curves. Convergence problems in [3] was addressed with heuristic approaches.

In [4] a system is modeled with (a simple) supply-side (distribution network) connected to central unit, which is connected to users (consumers). Users may be passive (just consumers) or active, that besides consumers also have energy generators and storage. Their demand side management consists of day-ahead bidding and real-time adjustments for variations of user needs and production. Their algorithm is distributed and iterative.

Methods from cited papers, and similar ones, with some effort might be adjusted even for our system model. However,
we wanted to define and test a simpler models and highlight storage management in microgrids. Furthermore, our system model has some subtle difference. Price adjustment ("bidding process") is performed differently. Distributor proposes prices (one per each period) and sends offers to microgrids – each microgrid receive its own offers, one for each period. Microgrids evaluate offers individually, and send back replies to distributor. Each reply, from single microgrid and for single period, consists of accept or reject decision of appropriate offer, with units and prices distributor should use in next iteration of bidding process. For each period of day ahead distributor tries to buy up to previously calculated required amount of energy (which is fixed for whole bidding process). In bidding process, when microgrids (cumulatively) don’t provide required amount, distributor might raise price for that period, if price isn’t already on predefined limit. Similarly, when distributor detects that offered amount for particular period exceeds distributor demands, distributor might lower price for that period (if not already on predefined lower limit).

One of our main requirement for model was simplicity. We are trying to find a simple algorithm for a microgrid in previously described setup. Therefore, in our initial work on the subject, we present a simple heuristic algorithm that is ready to be ‘copy-pasted’ in control program. Simplicity enable usage of this algorithm even in automatically controlled small scale system consisting of small microgrids managed by micro-controllers (e.g. a house with solar panels, battery and consumers) connected together with one controlling node that acts as distributor and can sell energy to third party. We think that most difference from real systems (generators, storages, consumers) and presented model can be accounted locally on particular microgrid, adjusting only amounts, prices and capacity.

As we didn’t yet put enough focus on distributor’s side, we are working on adaptation in section V and conclusion is given in section VI.

Rest of the paper is structured as follows. In sections II and III a system model and negotiation algorithm are presented. Experimental results are presented in section IV, possible adaptation in section V and conclusion is given in section VI.

II. THE SYSTEM MODEL

The system model consists of collection of microgrids and a distributor, as shown on Fig. 1. We do not model power grid with its generators and consumers. However, we assume they influence initial settings for distributor (demanded energy per periods, with price ranges).

A. Microgrid model

Every microgrid is defined with energy generators, local consumers and storage. Since we model renewable sources, energy production is based on weather conditions (e.g. wind or solar intensity) and can be (accurately) predicted for day ahead for each period (e.g. hour). Energy that can be produced in single period could be used for local microgrid consumers, for charging local energy storage (e.g. battery) and for selling it to distributor. First priority is to ensure enough energy for local consumers (cover it completely if possible). For that purpose microgrid can use storage: charge it when surplus is produced and discharge it when produced energy is insufficient. What energy is left can be sold to distributor. Storage can also be used for optimizing sale: store surplus energy when energy is ‘cheap’ and sell it later when its price is higher. In our model we didn’t account storage manipulation costs and loses. Also, price of energy generation is not modeled directly. Instead, minimum selling price is defined for each period that might include those production costs.

Let $M = \{m_1, m_2, \ldots, m_i, \ldots, m_N\}$ represent system with $N$ microgrids. Day (ahead) is divided into $P$ periods (e.g. hours).

Parameters that define single microgrid $m_i$ are:

- $ST_i$ – storage capacity (kWh)
- $SC_{i,0}$ – storage initial charge (kWh)
- for each period of next day $j$:
  - constant (given as input):
    - $CE_{i,j}$ – local consumers need (kWh)
    - $PE_{i,j}$ – production (generator) capacity (kWh)
    - $MP_{i,j}$ – minimal selling price (kn/kWh)
  - calculated within algorithm in $k$-th iteration:
    - $SC_{i,j}^k$ – starting storage charge (kWh)
    - $RC_{i,j}^k$ – reserved storage charge (kWh)
    - $RS_{i,j}^k$ – reply status: ACCEPT or REJECT
    - $RP_{i,j}^k$ – price for next iteration (kn/kWh)
    - $RU_{i,j}^k$ – units for next iteration (kWh)

Amount $RC_{i,j}^k$ represent storage charge that must be left in storage for next period. If $RC_{i,j}^k > SC_{i,j}^k$ than storage must be additionally charged in $j$-th period.
B. Distributor model

We assume that distributor has prediction for day-ahead demands from connected consumers and producing capabilities from connected generators, both from power grid side (excluding microgrids). Base on this data distributor can define appropriate demands toward microgrids (given per periods). Those demands with price ranges are inputs into our model. Distributor is defined with parameters:

- constant (given as input) for each period $j$:  
  - $RE_j$ – required energy (kWh)
  - $LP_j$ – minimal (lowest) price (kn)
  - $HP_j$ – maximal (highest) price (kn)
- calculated in $k$-th iteration for each $m_i$ and period $j$:
  - $OU_{i,j}^k$ – offered amount (kWh)
  - $OP_{i,j}^k$ – offered price (kn)
  - $OS_{i,j}^k$ – status: MORE or ENOUGH

Status is as a signal to microgrids in which periods demands are not met ($OS_{i,j}^k = MORE$) and in which periods offered amount exceeds demand ($OS_{i,j}^k = ENOUGH$).

C. Additional system constraints

For simplification and feasibility of result of each iteration, additional constraints are placed upon system model.

1) Fairness: In our model of distributor we defined that for each period, energy unit prices must be the same for each microgrid, i.e.:

$$OP_{i,j}^k = OP_{i,j}^k, \forall i, j, k$$

(1)

Condition (1) doesn’t affect microgrid’s model. Used microgrid’s model can be used in systems when previous condition isn’t valid, i.e. where different microgrids gets different prices for same period.

2) Feasibility I: Amounts in offers for next iterations must not exceed units provided in replies, i.e.:

$$OU_{i,j}^{k+1} \leq RU_{i,j}^k, \forall i, j, k$$

(2)

Therefore, initial offers in first iteration have:

$$OU_{i,j}^1 = 0, \quad OS_{i,j}^1 = MORE, \quad \forall i, j$$

(3)

Initial price must be from interval $[LP_j, HP_j]$. In our implementation we used middle value $(LP_j + HP_j)/2$.

3) Feasibility II: Cumulatively, offers for same period must not exceed required amount for that same period, i.e.:

$$\sum_{i=1}^{N} OU_{i,j}^k \leq RE_j, \forall i, j, k$$

(4)

4) Feasibility III: The microgrid in its reply must not offer more energy that it can sell, i.e.:

$$RU_{i,j}^k \leq SC_{i,j}^k + PE_{i,j} - CE_{i,j} - RS_{i,j}^k, \quad \forall i, j$$

(5)

5) Feasibility IV: When status in particular reply is ACCEPT, amount in that reply must not be lower that amount in corresponding offer, and price must not be higher than one in same offer:

$$RP_{i,j}^k \leq OP_{i,j}^k, \quad \forall i, j \quad | \quad RS_{i,j}^k = ACCEPT$$

(6)

$$RU_{i,j}^k \geq OU_{i,j}^k, \quad \forall i, j \quad | \quad RS_{i,j}^k = ACCEPT$$

(7)

No restriction on amount and price is set when offer is rejected.

6) No buying from power grid: If local consumers can’t be satisfied with local production and storage management we remodel microgrid with local consumers we can satisfy locally. Additionally energy must be bought, but we do not model this process. Algorithm 1 presents microgrid validation and remodel procedure. We assumed such procedure was performed on microgrids before negotiation process begun.

Algorithm 1 Discard excess consumers’ demand in $m_i$ for $j = 1 \text{ to } P$ do

if $SC_{i,j}^k + PE_{i,j} < CE_{i,j}$ then \{not enough energy?\}

$$CE_{i,j} = SC_{i,j}^k + PE_{i,j} \quad \text{(remodel consumers)}$$

end if

if $j < \text{periods}$ then \{calculate storage charge for $j+1$\}

$$SC_{i,j+1} = SC_{i,j}^k + PE_{i,j} - CE_{i,j}$$

if $SC_{i,j+1} > ST_i$ then \{can’t overcharge\}

$$SC_{i,j+1} = ST_i$$

end if

end for

III. THE NEGOTIATION ALGORITHM

Negotiations shown on Fig. 2 are initiated by distributor, who creates offers and sends them to microgrids. Each microgrid gets its own offers, one for each period. After evaluating offers, microgrids send back replies, one for each period.

![Fig. 2. Price negotiation process](image-url)
A. The microgrid’s algorithm

Microgrids evaluate offers separately, each microgrid looks at offers sent only to him. Main optimization criteria for a microgrid is maximization of profit from energy sold. However, the microgrid must first take into account its own local energy consumptions (per period). Units to sell are calculated iteratively: starting from period with the highest price per unit in offers. All surplus from that period and surplus from periods before that can be saved in storage and not already reserved can be included in offer for that period.

After any change in energy for sale, reservations on storage are first calculated, as presented in Algorithm 2.

Algorithm 2 Update_storage_and_reservations(i)

\[
\text{for } j = 2 \text{ to periods do } \{ \text{calculate } SC_{i,j} \} \\
SC_{i,j} = SC_{i,j-1} + PE_{i,j-1} - CE_{i,j-1} - RU_{i,j-1} \\
\text{if } SC_{i,j} > ST_i \text{ then } \{ \text{can’t overcharge} \} \\
SC_{i,j} = ST_i \\
\text{end if} \\
\text{end for} \\
\text{for } j = \text{periods} - 1 \text{ to } 1 \text{ do } \{ \text{calculate } RC_{i,j} \} \\
RC_{i,j} = CE_{i,j+1} + RC_{i,j+1}^k + RU_{i,j+1} - PE_{i,j+1} \\
\text{if } RC_{i,j} < 0 \text{ then } \\
RC_{i,j} = 0; \\
\text{end if} \\
\text{end for}
\]

First loop in Algorithm 2 calculate storage state after each period, accounting: initial storage state, production capability, local consumers and already allocated energy for selling. Second loop calculates amount of energy that must be left in storage for next period. This information is required in offer evaluation algorithm, Algorithm 3.

After optimization process (maximizing cumulative sell price), remaining units in periods are offered as well, because they might be requested in next iteration. Even if they are not requested, the microgrid can’t use them (store for later) so its better to offer them anyway – distributor might ask for it in next iteration.

B. The distributor’s algorithm

The distributor’s behavior is simulated with simple model, presented in Algorithm 4. The main objective of algorithm is to get as much energy as can be obtained per period, up to defined value. Secondary objective is to minimize total cost per period. Used rules to achieve objectives were simple: when offered amount is below required amount – price is raised (for that period), and vice versa, when offered amount is equal or above required amount – price is lowered.

Optimization is stopped when no change is detected in successive iterations or when predefined maximum number of iterations is reached (for situations where algorithm don’t converge), as shown in Algorithm 5.

Algorithm 3 Optimization algorithm for \( m_i \)

\[
\text{for } j = 1 \text{ to } P \text{ do } \{ \text{reset replies} \} \\
RS_{i,j} = \text{REJECT} \{ \text{status} \} \\
RP_{i,j} = \max(\text{MP}_{i,j}, OP_{i,j}^k) \{ \text{price} \} \\
RU_{i,j} = 0 \{ \text{units} \} \\
sort[j] = j \\
\text{end for} \\
\text{for } l = 1 \text{ to } P \text{ do } \{ \text{optimize for period sort}[l] \} \\
\text{for } n = l + 1 \text{ to } P \text{ do } \{ \text{search for higher price} \} \\
\text{if } OP_{i,\text{sort}[l]} < OP_{i,\text{sort}[n]} \text{ then} \\
\text{swap(sort}[l], sort[n]) \{ \text{swap values} \} \\
\text{end if} \\
\text{end for} \\
\text{end for} \\
\text{end for} \\
\text{for } j = 1 \text{ to } P \text{ do } \{ \text{offer remaining power in periods} \} \\
\text{Update_storage_and_reservations}(i) \\
\text{end for}
\]

IV. EXPERIMENTAL RESULTS

Proposed algorithm is evaluated on many examples. However, since single example requires large data set, only results from single example are presented in this paper. Microgrids and distributor parameters are just example, not chosen from real environment. In presented example ten microgrids are used. Each microgrid has storage capability for storing 50 kWh energy, but various predicted production capabilities during a day. In our simulation price change is limited to five discrete values defined by distributor separately for each period: lowest, low, normal (initial), higher and the highest, linearly distributed between \([LP_j, HP_j]\).

Cumulative input values for all ten microgrids and distributor are shown on Fig. 3 with: required energy from distributor (RE), production potential of microgrids (PE), local microgrid demands (LE) and initial price per period (IP). Initial price is set to roughly follow distributor’s demands, but also taking into account microgrids’ production capabilities, and to force
them to save as much as they can for peak periods.

Presented algorithm produced results shown on Fig. 4 with: energy offered to distributor with acceptable price (RU), storage charge at period beginnings (SC) and final price change from initial one (OP). We also put required energy (RE) on this figure to easier compare offers with demands.

Algorithm 4 Optimization algorithm for distributor

\[
\text{for } j = 1 \text{ to } P \text{ do } \text{[for each period]}
\]
\[
\quad \text{accepted} = \text{reserve} = 0
\]
\[
\quad \text{for } i = 1 \text{ to microgrids } \text{do } \text{[sum offers]}
\]
\[
\quad \quad \text{if } RS_{i,j}^{k} = \text{ACCEPT} \text{ then}
\]
\[
\quad \quad \quad \text{accepted} \leftarrow OU_{i,j}^{k}
\]
\[
\quad \quad \quad OR_{i,j}^{k+1} = OR_{i,j}^{k} \text{[offer to buy same amount again]}
\]
\[
\quad \quad \text{else}
\]
\[
\quad \quad \quad OR_{i,j}^{k+1} = 0
\]
\[
\quad \quad \text{end if}
\]
\[
\quad \quad \text{if } RP_{i,j}^{k} \leq OP_{i,j}^{k} \text{ then } \text{[reply acceptable?]}
\]
\[
\quad \quad \quad \text{reserve} \leftarrow RU_{i,j}^{k}
\]
\[
\quad \quad \text{end if}
\]
\[
\quad \text{end for}
\]
\[
\quad \text{if } \text{reserve} < RE_{j} \text{ then } \text{[not enough? raise price]}
\]
\[
\quad \quad \text{new}_\text{price} = \min(\text{OP}_{i,j}^{k} + \text{increment}, HP_{j})
\]
\[
\quad \quad \text{new}_\text{flag} = \text{MORE}
\]
\[
\quad \text{else } \text{[have enough, lower price]}
\]
\[
\quad \quad \text{new}_\text{price} = \max(\text{OP}_{i,j}^{k} - \text{decrement}, LP_{j})
\]
\[
\quad \quad \text{new}_\text{flag} = \text{ENOUGH}
\]
\[
\quad \text{end if}
\]
\[
\quad \text{for } i = 1 \text{ to microgrids } \text{do } \text{[update price and flags]}
\]
\[
\quad \quad \text{OP}_{i,j}^{k+1} = \text{new}_\text{price}
\]
\[
\quad \quad \text{OS}_{i,j}^{k+1} = \text{new}_\text{flag}
\]
\[
\quad \text{end for}
\]
\[
\quad \text{need} = RE_{j} - \text{accepted}
\]
\[
\quad \text{change} = \text{true}
\]
\[
\quad \text{while } \text{need} > 0 \text{ and } \text{change} \text{ do } \text{[increase offers]}
\]
\[
\quad \quad \text{change} = \text{false}
\]
\[
\quad \quad \text{for } i = 1 \text{ to microgrids } \text{do}
\]
\[
\quad \quad \quad \text{if } \text{need} > 0 \text{ and } \text{OU}_{i,j}^{k+1} \geq RU_{i,j}^{k} \text{ and }
\]
\[
\quad \quad \quad \quad \text{(RP}_{i,j}^{k} \leq \text{OP}_{i,j}^{k} \text{ or } \text{RP}_{i,j}^{k} \leq \text{OP}_{i,j}^{k-1}) \text{ then}
\]
\[
\quad \quad \quad \quad \quad \text{OR}_{i,j}^{k+1} = \text{OR}_{i,j}^{k} + 1
\]
\[
\quad \quad \quad \quad \quad \text{need} = \text{need} - 1
\]
\[
\quad \quad \quad \quad \text{change} = \text{true}
\]
\[
\quad \quad \text{end if}
\]
\[
\quad \text{end for}
\]
\[
\quad \text{end while}
\]
\[
\text{end for}
\]

Algorithm 5 Stopping criteria

\[
\text{change} = \text{false}
\]
\[
\text{for } i = 1 \text{ to } N \text{ do}
\]
\[
\quad \text{for } j = 1 \text{ to } P \text{ do}
\]
\[
\quad \quad \text{if } RS_{i,j}^{k-1} \neq RS_{i,j}^{k} \text{ or } RP_{i,j}^{k-1} \neq RP_{i,j}^{k} \text{ or }
\]
\[
\quad \quad \quad \text{RU}_{i,j}^{k-1} \neq RU_{i,j}^{k} \text{ or } OU_{i,j}^{k-1} \neq OU_{i,j}^{k} \text{ or }
\]
\[
\quad \quad \quad \quad \text{OP}_{i,j}^{k-1} \neq OP_{i,j}^{k} \text{ or } OS_{i,j}^{k-1} \neq OS_{i,j}^{k} \text{ then}
\]
\[
\quad \quad \quad \quad \text{change} = \text{true}
\]
\[
\quad \text{end if}
\]
\[
\quad \text{end for}
\]
\[
\quad \text{end for}
\]
\[
\text{if } k \geq k_{\text{MAX}} \text{ or } \text{change} = \text{false} \text{ then}
\]
\[
\quad \text{STOP}
\]
\[
\text{end if}
\]

Fig. 3. Cumulative values per period for: required energy (RE), production potential (PE), local microgrid demands (LE) and initial price (IP)

Fig. 4. Cumulative values per period for: required energy (RE), offered energy (RU), storage charge(SC), and price change from initial one (OP)
Analyzing presented results, we can extract several properties of proposed algorithm. Firstly, it behaves as expected: in periods where offered amount (RU) was not sufficient price reached the highest value, while in periods that offers exceed required amount (RE) price was lowered to the lowest possible value (periods 11 to 16). Periods with the highest price (7th and 18th) are covered as much as possible (completely in this example) using storage for difference between required and produced amount. Therefore, periods that precede ones with higher price may not be completely covered because storage was charged.

V. POSSIBLE IMPROVEMENTS

Major problem with presented algorithm is its stability. Single price change may trigger significant change in replies since microgrids sort periods by price and optimize periods in that order. We mitigated this problem by slowing price changes – enabling them only once in a few consecutive iterations, and by disabling of price lowering after predefined number of iterations. Those mitigations are not shown in Algorithm 4.

Fairness and cheating scenarios aren’t analyzed. However, since single microgrid can’t predict when negotiations will be completed, it should offer everything as soon as possible (as in proposed microgrid’s algorithm). Otherwise, distributor might buy larger amounts from other microgrids (included behavior in proposed distributor’s algorithm).

As already mentioned, optimality is not guaranteed with our algorithm. Within examples we used we found (not presented here) suboptimal solutions. We believe that better results might be achieved if distributor model (algorithm) is upgraded. However, even that might not guarantee optimal results for all cases.

Simple models for microgrids and distributor are used. In real application many parameters might be different and more complex. For example, minimal price per energy unit at which a microgrid is willing to sell energy might be calculated with a complex formula that take into account storage charging and discharging cost. Also, distributor may not offer all microgrids same price – price can be negotiated separately per microgrid. Used simplicity provides easier upgrade of particular model, distributor or microgrid. As long as constraints 2-6 from section II are met in all used algorithms in all nodes, every node might run different algorithm. That means that some microgrids might use one algorithm, other another, and so on. Same goes for distributor.

Presented algorithm is iterative and synchronous: distributor sends offer and collects all replies before evaluating them. However, algorithm could be adapted for asynchronous usage if constraint 1 is not enforced. In that case, communication between distributor and single microgrid could be achieved with three messages: 1) reply from microgrid, 2) offer from distributor and 3) a new reply from microgrid. All messages must conform to constraints 2-6. Microgrids replies, first and second one could be generated with same algorithm as presented. Distributor’s algorithm must be adapted since price in offers for same period might not be the same to different microgrids (e.g. new_price could be replaced with $O_i^{k+1}$).

What if prediction from any side proved to be incorrect? If we limit real-time adaptation only to energy redistribution from that moment forward, and ignore penalties imposed on node with wrong prediction that reflects on whole system (e.g. penalties could be handled as exception, separately), presented algorithm could be adapted even for real-time corrections. For example, algorithm will use current state of system (storage charge at that moment) as initial state, already negotiated prices as minimal prices (that can only increase), and only future periods in evaluations.

VI. CONCLUSION

In this paper we presented a simple algorithm for day-ahead negotiation between microgrids and a distributor. Separate algorithm is given for a microgrid and separate for a distributor. It’s assumed that all entities in negotiations have accurate prediction on energy requirements and production, given per period.

Presented algorithm produces expected solutions – profit optimization for microgrids’ surplus energy. Algorithms are simple, both microgrid’s, which was primary objective, and distributor’s, that was created for simulation purposes (and thus not fully optimized).

Stability problem was mitigated on distributor’s side. When mitigating elements are activated stable solution is obtained within next ten to fifteen iterations. Therefore, at least twenty iterations should be put as upper limit, if one is required at all. In simpler examples, stability was reached within only four iterations.

Negotiation results weren’t always optimal, even if Nash equilibrium was reached. Part of that problem lies with distributor’s algorithm, which must be improved. However, optimal solution guarantee (as one presented in [4]) is generally hard to provide in distributed algorithms. Therefore, we are satisfied with a occasional ‘only good’ result which is provided by presented algorithm.

Simplicity of algorithm provides several advantages. Firstly, it enables algorithm implementation even on simplest hardware (microgrid’s micro-controller). Secondly, adaptation for specific microgrid or distributor is simpler – not all nodes must run same algorithm, as elaborated in section V.

REFERENCES