Learning Bidding Strategies in Local Electricity Markets using Ant Colony Optimization

Fernando Lezama, Ricardo Faia, Joao Soares, and Pedro Faria
GECAD Research Group, Polytechnic of Porto (ISEP/IPP),
Porto, Portugal
Email: {flz, rfmfa, jan, pnf}@isep.ipp.pt

Zita Vale
Polytechnic of Porto (ISEP/IPP),
Porto, Portugal
Email: zav@isep.ipp.pt

***Abstract*—Local energy markets (LM) are attracting significant interest due to their potential of balancing generation and consumption and supporting the adoption of distributed renewable sources at the distribution level. Besides, LMs aim at increasing the participation of small end-users in energy transactions, setting the stage for transactive energy systems. In this work, we explore the use of ant colony optimization (ACO) for learning bidding strategies under a bi-level optimization framework that arises when trading energy in an LM. We performed an empirical analysis of the impact of ACO parameters have in the learning process and the obtained profits of agents. After that, we analyze and compare ACO performance against an evolutionary algorithm under a realistic case study with nine agents trading energy in the day-ahead LM. Results suggest that ACO can be efficient for strategic learning of agents, providing solutions in which all agents can improve their profits. Overall, it is shown the advantages that an LM can bring to market participants, thereby increasing the tolerable penetration of renewable resources and facilitating the energy transition.**

***Index Terms*—Ant Colony Optimization; Evolutionary computation; Learning Strategy; Local energy market; Renewable energy.**

I. Introduction

The Clean Energy for all European package is one of the most essential legislation directives to strengthen the European leadership in the energy transition. This is achieved in practice by placing consumers at the heart of the energy system. In addition, the European Green Deal aims to transform Europe as the world's first climate-neutral continent by 2050 [1]. Consumers involvement is key to assure this successful transition, while playing the role of producers of green energy through renewable energy sources, helping to smooth the local and regional energy system transition to solve important challenges. Hence, realization of a secure and reliable future energy system requires development and demonstrations of technical and market solutions keeping the consumers at the centre of the process [2], [3].

In this paradigm, the concept of local energy markets (LM) provides a unique platform to trade renewable energy at the local level while contributing to a climate-neutral Europe, i.e. cutting carbon emissions. Indeed, the proliferation of distributed renewable energy sources has led to a significant interest in local energy transactions at the lower level in order to increase renewable energy use, deffer grid investments and reduce energy costs [4], [5], [3]. Also, LM will empower the end-users to participate as real actors in the energy community while promoting a sustainable electricity transition [5].

The trading mechanism in LM is expected to be complex and computing-intensive. Several mechanisms have been developed to facilitate automated trading, such as peer-to-peer, blockchain, etc. [2], [6]. Evolutionary Computation (EC) provides a powerful set of tools for solving complex optimization problems in the energy domain, which are also well suited for the LM optimal bidding problem [7], [8]. EC methods are in general straightforward to implement when compared with classical approaches. Moreover, learning features can be incorporated by inspiration on the "intelligence" seen in nature.

This paper is based on a prior work [8] that used EC to solve the optimal bidding in LM. The previous results indicated that agents using LM could profit by using a strategy based on social welfare to obtain higher profits with LM. The limitation of the previous proposal is that information from all agents must be disclosed to the decision-maker, the so-called "perfect information", which in turn provides the "best" bidding strategy to each participant. Therefore, in this work, we propose that each agent can learn with their actions in the LM without disclosing all the information (thus, considering a more realistic scenario). To implement the learning strategy we adopt an EC method, namely Ant Colony Optimization (ACO). ACO has different applications in energy, demonstrating its effectiveness [9], [10], [11], [12]. We use the same case study as in [8] to compare the performance of ACO with the centralized proposal, previously published. The realistic case study considers nine agents trading energy in the day-ahead LM. The research outcomes are further discussed in the remainder of the paper and are organized as follows after this introduction: Section II presents the optimal bidding problem formulation; Section III describes the EC approach, namely ACO for learning strategic bidding in LM; Section IV presents the results and discussion; Finally, Section V fully draws the conclusions of the research paper.

This work has received funding from the European Union's Horizon 2020 research and innovation programme under project DOMINOES (grant agreement No 771066) and CENERGETIC (POCI-01-0145-FEDER-028983 and PTDC/EEI-EEE/28983/2017), from FEDER Funds through COMPETE program and from National Funds through (FCT) under the project UIDB/00760/2020, and grants CEECIND/02814/2017, CEECIND/02887/2017, SFRH/BD/133086/2017.

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

II. Problem Formulation

In this section, the formulation of the problem is described. This formulation was firstly proposed in [8]. We adopt the same formulation for the optimal bidding problem, which consists in a day-ahead LM bidding optimization, in which agents submit bids/offers to maximize their profits (producers) or minimize their costs (consumers). We assume the following type of agents: consumers, producers, and prosumers (i.e., consumers with generation capabilities). Also, agents have access to the main grid, which works as a back-up system. Therefore, agents can trade energy in the LM with prices between the feed-in tariff (c_F) and the grid electricity tariff (c_G). It is assumed that $c_F < c_G$ and therefore buy/sell energy from the grid is less beneficial to agents than transacting energy in the LM. Figure 1 illustrates the local market scenario described above.

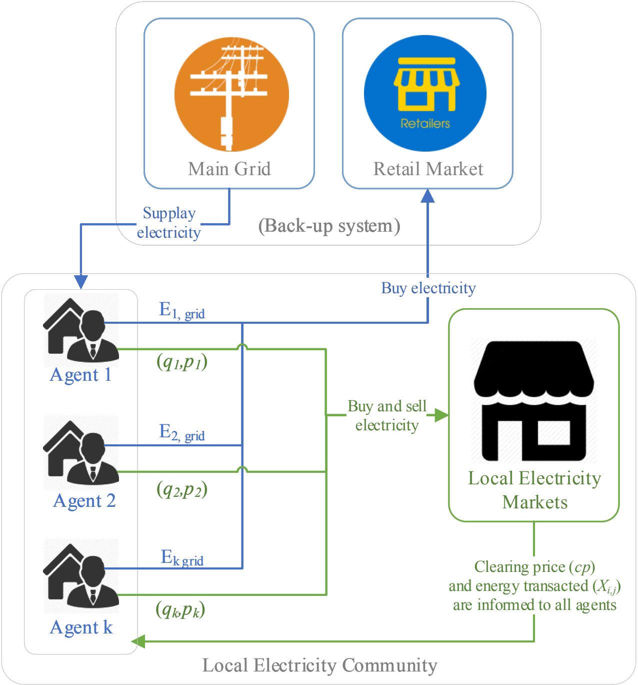

Fig. 1: Considered LM and the grid as back-up system.

A. Bi-level optimization problem

The LM bidding is formulated as a bi-level optimization problem. The upper-level problem is the maximization of agents' profits, and the lower-level problem is the maximization of energy transacted in the LM. Hence, after determining the clearing price, the lower-level affects the upper-level by modifying the profits/costs of all agents.

Consider a set of consumer agents $I = 1, 2, ...N_c$, and producer agents $J = 1, 2, ...N_p$, where each agent i wants to minimize its costs while agents j want to maximize their profits. The upper problem, therefore, is a multi-objective problem in which each agent wants to maximize/minimize their profits/costs.

The optimization problem for each consumer agents (minimization of costs) can be formulated as:

$$\text{minimize} \quad C_i = \sum_j cp \cdot x_{j,i} + c^G \cdot Ebuy_{grid,i} \tag{1}$$

where cp is the LM clearing price, $x_{j,i}$ contains the energy sold by agent j to agent i in the LM, c^G is the grid price, and $Ebuy_{i,grid}$ is the energy bought by agent i from the grid.

On the other hand, producer agents try to maximize their profits regarding their production marginal cost as follows:

$$\text{maximize} \quad P_j = \sum_i cp \cdot x_{j,i} + c^F \cdot Esell_{j,grid} - c^m * G_j \tag{2}$$

where cp is the LM clearing price (equal for buyers and sellers), $x_{j,i}$ contains the energy sold by agent j to agent i, c^F is the feed-in tariff, $Esell_{j,grid}$ is the energy sold by agent j to the grid, and $c^m * G_j$ represents the marginal cost associated to j. We assume that $c^m = 0$ for PV generation, and $c^m = c^m_{CHP}(G_j)$ (i.e., the marginal cost associated to a combined heat and power (CHP) generator) is defined as a monotone decreasing function [3]:

$$c^m_{CHP}(G_j) = \frac{b_{CHP} \cdot \sqrt{G_j}}{G_j} \tag{3}$$

where b_{CHP} is a cost factor of the CHP generation unit and G_j is the energy produced by the CHP.

The agents' profits/costs are influenced by the LM clearing price cp, which is determined in the lower-level problem and depends on the market clearing. The lower-level problem is formulated as an symmetric pool market, in which bids and offers are allocated using a merit order procedure to determine the supply and demand curves [13]. The clearing price is determined as the price in which demand equals supply (i.e., the LM clearing price). Each agent's bid influences the LM price. Since we are interested in increasing the overall mean profits of the system and at the same time provide solutions that distribute the earns among all agents, the objective function is modeled as:

$$\max \sum_j (P_j)/N_p - \sum_i (C_i)/Nc \tag{4}$$

where P_j and C_i are objectives in conflict since agents want to achieve the best result for their own. In this work, we avoid a multi-objective formulation of the problem by adopting the summation of the individual costs. In the next section, we describe how we adapt this equation for the implemented EC methods, namely ACO and Vortex Search.

III. Ant Colony for Learning Strategic Bidding in LM

Ant Colony Optimization (ACO) is a swarm intelligence problem-solving approach that takes inspiration from the social behaviour of some ant species. For instance, ants deposit pheromone on the ground to mark favourable paths to food, that is reinforced by other members of the colony depending on their quality. ACO exploits a similar mechanism when solving optimization problems, reinforcing paths (solutions) with better performance [14].

To apply the ACO algorithm to a given problem, and adequate model is needed. In particular, for a combinatorial optimization problem, a model $P = (\mathbf{S}, \Omega, f)$ consists of:

- a search space $\mathbf{S}$ defined over a finite set of discrete decision variables $X_i, i = 1, ..., D$;
- a set Ω of constraints among the variables;
- an objective function $f : \mathbf{S} \in \mathbb{R}^+$

The generic variable X_i can take value over the set $\mathbf{V}_i = \{v_i^1, ..., v_i^{|\mathbf{V}_i|}\}$. A feasible solution $s \in \mathbf{S}$ is a complete assignment of values that satisfies constrains in Ω. A solution $s^* \in \mathbf{S}$ is optimal iff: $f(s^*) \leq f(s) \forall s \in \mathbf{S}$. In the next subsections, we explain how ACO is used to model strategic bidding in LM.

A. Ant Colony Optimization

Several variants of ACO have been proposed in the literature to solve a wide range of combinatorial optimization problems. Also, there are some modified ACO versions for the continuous domain [15]. The bi-level optimization problem presented in this work was solved by EA in [8], targeting continuous variables for the bids/offers of quantity and prices. Different from [8], in this work, we explore the use of ACO to learn optimal bidding policies from a discrete set of bidding options, and for each independent agent.

To that end, we follow the standard steps of the ACO algorithm presented in algorithm [15]. In fact, the first ACO algorithm was called as ant system (AS) [16], and since then, many variants (such as the MAX-MIN Ant System [17] or the Ant Colony System [18]) have been applied to diverse optimization problems. In this work, we limit our study to adapt the basic AS to the problem of bidding in local markets. In the next subsections, we explain in detail the phases of each iteration of our approach. We expect to explore other ACO variants in further studies.

Algorithm 1 ACO pseudocode

```
Set parameters, initialize pheromone trails.
while termination condition not met do
    Construct Ant Solutions.
    Evaluate fitness of ants.
    Apply Local Search (optional).
    Update Pheromones.
end while
```

B. Solution construction phase

In the bidding optimization problem, a solution for each agent $K = \{1, 2, ...N_k\}$, where N_k is the number of agents, can be represented through a set of T tuples including two variables, where T is the number of considered periods (i.e., $T = 24$ periods in the day-ahead market). Agents aim at determining the best tuple $\{q_{(k,t)}, p_{(k,t)}\} \forall k \in K, t \in T$, representing the optimal price and quantity to bid in the LM for each agent. Therefore, we define a vector $\vec{x} = \{[q_{k,t}] \cup [p_{k,t}]\}$ including the bids for quantity and price that the kth agent will send to the LM. We use a sign convention in which a positive quantity represents a bid (i.e., buying in the market), while a negative quantity represents an offer (i.e., selling in the LM).

In the continuous domain, each producer agent can put a bid of quantity in the LM within allowed bounds $[0, L_{max}]$ (i.e., between 0 and their maximum consumption), while producer agents can send offers within the bounds $[-P_{max}, 0]$ (i.e., between 0 and their maximum production capacity). The bounds for bid/offers of price are the same for all agents and within the range $[c^F, c^G]$. Such bounds are discretized in our application, divided the range of the bound in L equal parts (L is assumed to be a discretization level). Therefore, the continues variables q_i and p_i are mapped to a discrete space $q_{(i,l)} = \{q_{(i,1)}, q_{(i,2)}, ..., q_{(i,L)}\}$ and $p_{(i,l)} = \{p_{(i,1)}, p_{(i,2)}, ..., p_{(i,L)}\}$. This is done in order to turn the problem into a combinatorial one that can be easily adapted to be solved by ACO.

In ACO, a set of m artificial ants constructs solutions from elements of a finite set of available solution components $C = \{q_i, p_j\}$, $i = 1, ..., T$, $j = 1, ...T$. A solution construction starts from an empty partial solution $s^p = \emptyset$. At each construction step, the partial solution s^p is extended by adding a feasible solution component $\mathbf{N}(s^p) \subseteq \mathbf{C}$ (i.e., a bid/offer of quantity and price from the available discrete set). In ACO, the choice of a solution component is guided by a stochastic mechanism biased by a pheromone associated with each element $\mathbf{N}(s^p)$. In particular, the selection of a bid/offer of quantity and price is given by the following probabilities:

$$Prob_{q(i,l)} = \frac{\tau^{\alpha}_{q(i,l)} \cdot \eta^{\beta}_{q(i,l)}}{\sum_{c_{q_i} \in \mathbf{N}(s^p)} \tau^{\alpha}_{q(i,l)} \cdot \eta^{\beta}_{q(i,l)}} \tag{5}$$

$$Prob_{p(i,l)} = \frac{\tau^{\alpha}_{p(i,l)} \cdot \eta^{\beta}_{p(i,l)}}{\sum_{c_{p_i} \in \mathbf{N}(s^p)} \tau^{\alpha}_{p(i,l)} \cdot \eta^{\beta}_{p(i,l)}} \tag{6}$$

where the parameters α and β control the relative importance of the pheromone (i.e., τ) versus the heuristic information (i.e., η). In the next subsection, we define the pheromone and heuristic matrices used for the bidding problem.

C. Pheromone and heuristic information

A pheromone value is associated with each possible solution component; that is, with each potential assignment of a value to a variable. Formally, in this work the pheromone trail is modelled as two matrices τ_q and τ_p, both of dimension $T \times L$, where T is the number of periods, and L is the discrete level (i.e., in how many parts we break the continuous bound of the variables). The pheromone trail stores the information learned from the ants. For this application, a pheromone trail indicates the bid/offer of quantity and price that the agent put in the LM for each period. The information is reinforced according to the fitness of each ant after obtaining a complete solution. The better the fitness, the more the reinforcement of matrices τ_q and τ_p.

Additionally, a heuristic information matrix η_q and η_p, also both of dimension $T \times L$, are defined with specific information of the problem to be solve. The heuristic information related to the decision of quantity is defined as:

$$\eta_{q(i,l)} = \begin{cases} q_{i,l} & \text{if } q_{i,l} \geq 0 \quad \text{(for consumers)} \\ |q_{i,L-l+1}| & \text{otherwise (for producers)} \end{cases} \tag{7}$$

where $q_{i,l}$ is the discrete bid/offer value of energy quantity. As it can be observed, when $q_{i,l} \geq 0$, the matrix reinforce the decision of buying as much as possible in the local market. On the contrary, when $q_{i,l} < 0$ (representing a producer), the matrix is flip to favour major quantities to sell into the LM.

A similar differentiation is done for the heuristic information of price:

$$\eta_{p(i,l)} = \begin{cases} 1/p_{i,l} & \text{if } q_{i,l} \geq 0 \quad \text{(for consumers)} \\ p_{i,l} & \text{otherwise (for producers)} \end{cases} \tag{8}$$

D. Update of pheromone trails

An update of the pheromone trails is performed at each iteration to increase the values associated with promising solutions, and to decrease those that are associated with bad ones. This is typically achieved by reducing all the pheromone values through pheromone evaporation, and by increasing the pheromone levels related to a chosen set of good solutions in function of the fitness in the objective function.

After the construction of all the solutions, the pheromone trails are modified through and evaporation and reinforcement as:

$$\tau_{q(i,l)} \leftarrow (1-\rho) \cdot \tau_{q(i,l)} + \sum_{m=1}^{M} \Delta\tau^{m}_{q(i,l)} \tag{9}$$

$$\tau_{p(i,l)} \leftarrow (1-\rho) \cdot \tau_{p(i,l)} + \sum_{m=1}^{M} \Delta\tau^{m}_{p(i,l)} \tag{10}$$

where ρ is an evaporation rate parameter defined by the user to simulate the evaporation process of pheromone, and $\Delta\tau_{p(i,l)}$ and $\Delta\tau_{q(i,l)}$ are the pheromone contribution of ant m to the trace calculated as:

$$\Delta\tau^{m}_{q(i,l)} = \begin{cases} 1/Fit_m & \text{if } q_{i,l} \in sol^m \wedge Fit_m \geq 0 \\ |Fit_m| & \text{if } q_{i,l} \in sol^m \wedge Fit_m < 0 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

$$\Delta\tau^{m}_{p(i,l)} = \begin{cases} 1/Fit_m & \text{if } p_{i,l} \in sol^m \wedge Fit_m \geq 0 \\ |Fit_m| & \text{if } p_{i,l} \in sol^m \wedge Fit_m < 0 \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where Fit_m is the fitness of the solution constructed by ant m (denoted by sol^m). The inverse and absolute value are used because the fitness is associated with the costs/profits of each agent, which in this applications can take negative values (representing profits). This procedure reinforces the traces of the best solutions found. The iterative process is repeated until a termination condition is met, such as the number of generations.

E. Distributed ACO for each agent

The optimization problem, seen as a whole, searches for the optimal bidding of agents in the LM to maximize their profits (or minimize costs in the case of consumers). The definition of ACO from the previous section can be applied to each agent independently since each of them can have their own pheromone matrix, and reinforce such matrix as a function of their independent profit/cost (i.e., their fitness). Therefore, the ACO algorithm for all the system is showed in 2.

Algorithm 2 Distributed ACO pseudocode

```
Set parameters, initialize pheromone and heuristic information trails.
while termination condition not met do
    for $k = 1 : N_k$ do
        for $m$=1:M do
            Construct Ant Solutions.
        end for
    end for
    for $m$=1:M do
        Apply market clearing considering all agents.
        Evaluate fitness of agents $Fit_{(k,m)} \forall k \in N_k$.
    end for
    Update Pheromones.
end while
```

Notice that the fitness evaluation is independently done by each agent in function of Eqs. 1 and 2. However, we can also have a measurement of the fitness of all the system, similar to [8], by grouping the resulting fitness of all agents as:

$$Profits = [P_1, ...P_k] \forall t \in T \tag{13}$$

This vector includes all the profits of agents, and therefore, a combined fitness of all agents can be expressed as follows:

$$Fitness(\vec{x}) = -mean(Profits) + std(Profits) \tag{14}$$

where $mean(Profits)$ and $std(Profits)$ are functions that compute the average and standard deviation of the profits that all agents obtained considering the bids/offers encoded in the individual. The negative sign in the first term is used to transform the profits maximization problem into a minimization one. The less the value in Eq. (14), the better the mean profits achieved by all agents. Notice that this combined measure considers exchange of information between agents, which in practice cannot be fulfilled if agents are not willing to share information. However, the combined fitness is used in the results section to evaluate the performance of ACO compared to the best values obtained in [8].

IV. Results and Discussion

This section is divided in three parts: A) case study, B) ACO tuning of parameters, C) ACO performance and comparison against other heuristics. The experiments were done in MATLAB 2018a in a computer with Intel Xeon(R) E5-2620v2@2.1 GHz processor with 16GB of RAM running Windows 10.

A. Case Studies

We adopt the same case study used in [8], which allows a straightforward comparison of results. The case study considers nine agents, from which 3 of them are consumers, 3 are prosumers (i.e., consumers with PV generation capabilities), and 3 are CHP small generators. Sample power profiles of residential houses and PV systems are built for the case study using the open datasets available in PES ISS website[1]. Three standard house power profiles and a PV power profile (see Fig. 2) are used to generate data of other agents using a randomized function with uniform distribution, 20% around the standard profiles. Fig. 2 also provides the power ranges of the base profiles. We also consider generator agents corresponding to CHPs with a maximum generation capacity of 2kW and a marginal cost calculated with Eq. (3) with a factor $b_{CHP} = 0.18$ EUR/kWh [3]. Finally, feed-in and grid tariffs are set to $c^F = 0.12$ and $c^G = 0.28$ EUR/kWh as in [3].

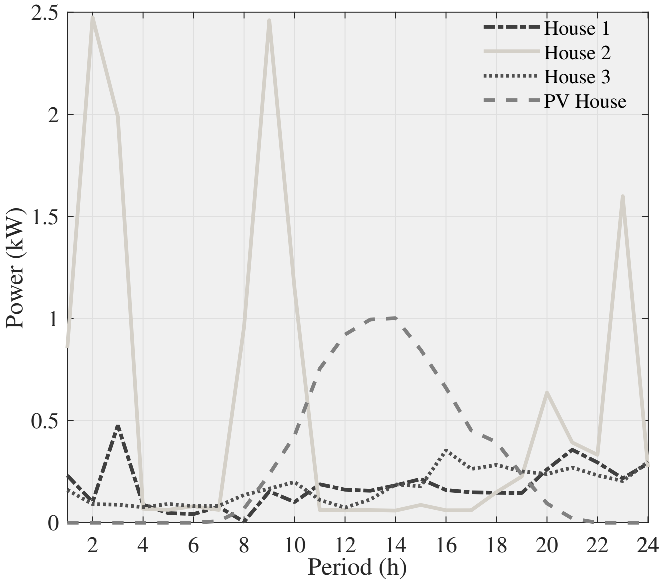

Fig. 2: Profiles used in the case study. Ranges of power (in kW): house 1 [0.18-0.48], house 2 [0.06-2.50], house 3 [0.07-0.36], PV (house) [0-1].

B. ACO tuning of parameters

The ACO algorithm employed in this work has some control parameters that can impact the performance of the learning procedure. In particular, five parameters are fine-tuned exploring a different range of values, namely ρ, α, β, m, and number of iterations.

In the first set of experiments related to fine-tuning of parameters, we fix a default set of values and explore variations around the tuned parameters. Ten trials are done, and the average of fitness value, as well as profits/costs of each type of agent, are reported. Table I present the default value used in this first experiment, as well as the tested values and recommended one.

TABLE I: Exp1 default and tested values for ACO parameter tuning.

Parameter	Default	Tested	Recommendation
ρ	0.5	[0.1,0.3,0.5,0.7,0.9]	0.3
α	1	[0.5,1,2,3,4]	0.5
β	1	[0.5,1,2,3,4]	0.5

Number of ants and iteration are set to $m = 10$ and $Iter = 100$

The ρ parameter, typically in the range [0,1], is used as an evaporation rate (see Eqs. (9) and (10)). Table II presents the results obtained regarding average costs (negative values indicate profits) of all agents (and by type of agents) showing that a better fitness is achieved with $\rho = 0.3$, and that higher values have an impact in the resulting fitness.

TABLE II: Evaporation rate ρ fine tuning.

ρ	Costs	Consumer	Prosumer	Producer	Mean fit.	Time
0.1	6.03	4.47	1.92	-0.36	3.66	0.52
0.3	**5.98**	**4.60**	**1.79**	**-0.40**	**3.65**	**0.51**
0.5	6.10	4.55	1.93	-0.39	3.71	0.52
0.7	6.33	4.70	1.97	-0.34	3.80	0.51
0.8	6.35	4.79	1.88	-0.31	3.81	0.52

The parameters α and β control the relative importance of the pheromone versus the heuristic information in Eqs. (9) and (10). Tables III and IV summarize the results, showing that low values of this parameters are preferred for this application in particular. For instance, the greater the value of these two parameters, the worse the result in terms of fitness and costs. Therefore, a recommended value of $\alpha = 0.5$ and $\beta = 0.5$ is provided.

TABLE III: Relative importance of pheromone α fine tuning.

α	Costs	Consumer	Prosumer	Producer	Mean fit.	Time
0.5	**5.97**	**4.50**	**1.80**	**-0.32**	**3.61**	**0.52**
1.0	6.10	4.55	1.93	-0.39	3.71	0.52
2.0	6.24	4.68	1.87	-0.31	3.77	0.51
3.0	6.38	4.78	1.99	-0.38	3.85	0.53
4.0	6.46	4.80	2.04	-0.38	3.89	0.55

TABLE IV: Relative importance of heuristic info β fine tuning.

β	Costs	Consumer	Prosumer	Producer	Mean fit.	Time
0.5	**5.20**	**4.21**	**1.46**	**-0.48**	**3.30**	**0.50**
1.0	6.10	4.55	1.93	-0.39	3.71	0.52
2.0	7.63	5.47	2.44	-0.29	4.41	0.48
3.0	8.21	5.96	2.59	-0.34	4.68	0.45
4.0	8.38	6.02	2.70	-0.34	4.76	0.44

After determining the best set of values for ρ, α and β, we fixed these values and proceeded with a fine-tuning of parameter m (number of ants) and $Iter$ (iterations as stop criteria). Table V summarize the tested values and the recommendation achieved.

Tables VI and VII present the results of the experiments varying the number of ants and iterations respectively. It

TABLE V: Exp2 default and tested values for ACO parameter tuning.

Parameter	Default	Tested	Recommendation
m	10	[5,10,20,30,40,50]	20, 40
Iter	100	[100,200,300,400,500,1000]	500,1000

Other parameters are set to $\rho = 0.3$, $\alpha = 0.5$ and $\beta = 0.5$

can be noticed that the quality of solutions improves along with the increase of these two parameters. However, these two parameters, in particular, have an impact also in the computation time, so large values should be avoided in case the problem is scaled to larger instances. For this reason, and this particular case, we advise selecting values in the range of $m = 20$ and $m = 40$ which provide solutions in acceptable times (from one to two minutes) and of acceptable quality. The same reasoning can be done for the number of iterations, advising a value of 500 iterations that already provide quality solutions in half of the time of the best-found value.

TABLE VI: Number of ants m fine tuning.

m	Costs	Consumer	Prosumer	Producer	Mean fit.	Time
5	5.64	4.31	1.61	-0.28	3.44	0.26
10	5.68	4.30	1.67	-0.29	3.48	0.51
20	5.42	4.24	1.49	-0.31	3.33	1.07
30	5.24	4.12	1.42	-0.29	3.25	1.61
40	5.30	4.12	1.48	-0.31	3.28	2.13
50	**5.27**	**4.15**	**1.36**	**-0.23**	**3.23**	**2.68**

TABLE VII: Number of Iterations for fine-tuning.

Iter	Costs	Consumer	Prosumer	Producer	Mean fit.	Time
100	5.24	4.12	1.42	-0.29	3.25	1.61
200	5.26	4.13	1.42	-0.29	3.25	3.27
300	5.18	4.04	1.42	-0.28	3.22	4.98
400	5.31	4.07	1.48	-0.24	3.28	6.26
500	5.18	4.08	1.37	-0.27	3.22	7.98
1000	**5.14**	**4.05**	**1.34**	**-0.25**	**3.18**	**15.74**

C. ACO comparison and performance assessment

After defining the best set of parameters for ACO, we compare its performance against the vortex search (VS) [19] algorithm, which obtained the best results in [8]. VS is not strictly speaking a population-based approach but evaluates in each iteration a given number of neighbour solutions (NS), which results in a given number of function evaluations (FE) per iteration. To perform a fair comparison between both approaches, we perform two comparisons with the same number of FE used in [8]. Table VIII presents the setting of iterations and population that results in an equivalent number of FEs for both approaches. In addition, ACO parameters were set to $\rho = 0.3$, $\alpha = 0.5$, $\beta = 0.5$, while VS does not need any other parameter specification.

TABLE VIII: Setting for algorithms comparison.

	NS	Iter	FE
VS	5	2000	10000
	20	2000	40000
	m	**Iter**	**FE**
ACO	20	500	10000
	40	1000	40000

The algorithms were run for 30 trials each, and the reported results correspond to the mean value of those 30 trials. Table IX presents the total costs of the system, the costs by group of agents (i.e., consumers, prosumers, and producers), and the average fitness and time required by each algorithm to perform the optimization. The table also includes the overall costs/profits that agents can attain in the absence of an LM (used as a baseline). It can be seen that ACO outperforms the results obtained with VS in terms of fitness and overall costs of the system, in both cases. Notice, however, that VS can attain better profits for producers as a result of the centralized approach employed that searches for fair distribution of resources among all agents. However, despite the distributed learning employed in ACO, producer agents can still get profits. Also, notice that the optimization times are kept in similar ranges, around 5 minutes for 10000 FEs and 20 minutes for 40000.

To analyze the profits obtained by independent agents, we show plots with the individual agents' profits. Figure 3a shows the individual profits achieved by the agents in the baseline case. It can be seen that in the baseline when no LM is available, the best option for the generator agents is not selling anything to the grid due to their associated marginal cost being higher than the grid feed-in tariff. This turns in zero profits/costs for them, which is bad for the system as a whole. On the contrary, notice that the best solutions found with ACO (Fig. 3b) and VS (Fig. 3c) achieve, on average, a lower mean cost value, and all agents improve their situation, i.e., consumer agents reduce their costs while generator agents (agents 7 to 9) and even prosumers (agents 4 to 6) obtain some profits.

Finally, we analyze the convergence and learning behaviour of ACO in terms of its fitness value evolution and independent profits/costs. Fig. 4 shows the mean convergence curve and standard deviation of the total fitness of the system using $m = 20$ and $Iter = 500$. It can be seen that the algorithm has a fast convergence rate during the first 150 iterations, and presents a wider standard deviation in the final stage of the iterative process. Nonetheless, ACO attains a good average cost in all the cases, which demonstrates its robustness.

Regarding the learning behaviour of independent agents, Fig. 5 shows the average costs evolution during the iterative procedure. It can be seen that consumer and prosumer agents tend to convergence quickly to a low value, while for producers, it turns out to be challenging to find a strategy that improves their profits. This can be explained by the fact that all agents are using the same parameters in their evaporation/reinforcement of pheromone process. Still, their obtained profits/cost vary from agent to agent, so the learning profits are different. Since this is a distributed version of ACO, in which each agent uses its own fitness function to reinforce the pheromone and biased its decisions, it would

TABLE IX: Comparison of ACO vs. VS algorithm. The Table also include the baseline where no LM is considered.

FEs	Algortihm	Overall Costs (EUR)	Costs by group of agents (EUR)			Fit.	Time (mins)
			Consumers	Prosumers	Producers		
10000	ACO	5.24	4.10	1.45	-0.31	3.25	5.17
	VS	5.77	4.79	1.76	-0.79	3.58	4.87
40000	ACO	5.06	4.02	1.33	-0.29	3.15	21.05
	VS	5.29	4.57	1.58	-0.85	3.34	20.31
	No LM	9.00	6.15	2.84	0.00	4.97	-

*Negative values represent profits

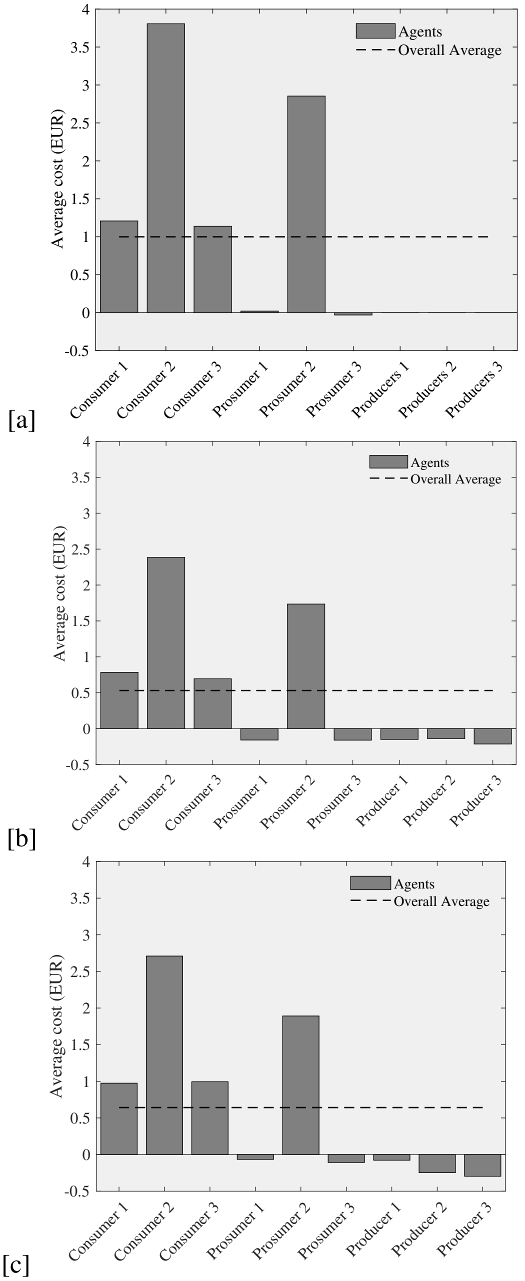

Fig. 3: Overall profits achieved by individual agents. [a] Baseline solution. [b] Best solution found with ACO using 1000 FE. [c] Best solution found with VS using 1000 FE.

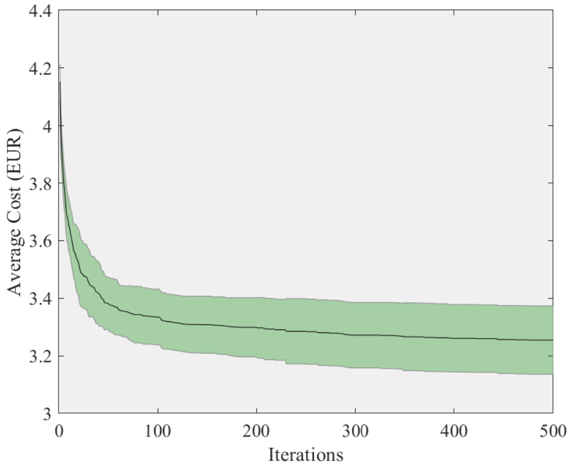

Fig. 4: Average convergence and standard deviation of ACO.

make sense to tune the parameters independently for each agent. An interesting avenue of research is advised concerning the fine-tuning of parameters for each agent.

V. Conclusions

In this paper, a simple ACO algorithm has been applied to solve a bi-level optimization problem that arises in the context of LM. The problem of competitive agents is not trivial since all agents search for a maximization/minimization of their own profits, modifying the clearing price with their actions, and affecting other agents' decisions. The problem was previously solved with a centralized approach in which the share of information was assumed. However, in practice, agents aim at the definition of a bidding strategy without sharing information with their peers. Therefore, we have implemented a distributed version of ACO in which each agent reinforce its own pheromone matrix based on its profit/cost. In this way, agents are able to learn their own bidding strategies, while the whole system converges to a better overall fitness value. As further work, independent definition of ACO parameters for each agent can be explored, since it is expected that the learning rate varies from agent to agent. Also, advanced versions of ACO, such as the min-max ACO or the ACsystem can be implemented under this framework. The application of such model using distributed computing platforms and more realistic scenarios is also worth to be explored.

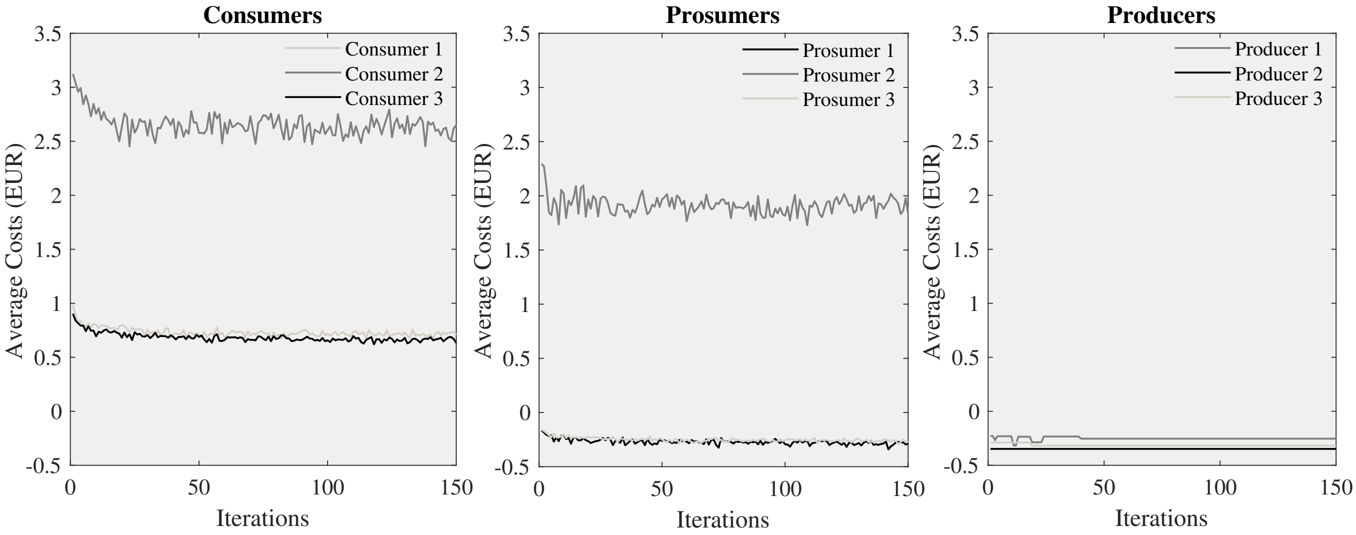

Fig. 5: Learning behavior of independent agents using ACO.

References

[1] European Comission. (2019) A european green deal: Striving to be the first climate-neutral continent. [Online]. Available: https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal_en

[2] P. Siano, G. De Marco, A. Rolán, and V. Loia, "A survey and evaluation of the potentials of distributed ledger technology for peer-to-peer transactive energy exchanges in local energy markets," *IEEE Systems Journal*, vol. 13, no. 3, pp. 3454–3466, 2019.

[3] E. M. Mengelkamp, "Engineering local electricity markets for residential communities," Ph.D. dissertation, Karlsruher Institut für Technologie (KIT), 2019, 37.06.01; LK 01.

[4] P. Olivella-Rosell, E. Bullich-Massagué, M. Aragüés-Peñalba, A. Sumper, S. Ø. Ottesen, J.-A. Vidal-Clos, and R. Villafáfila-Robles, "Optimization problem for meeting distribution system operator requests in local flexibility markets with distributed energy resources," *Applied energy*, vol. 210, pp. 881–895, 2018.

[5] F. Lezama, J. Soares, P. Hernandez-Leal, M. Kaisers, T. Pinto, and Z. Vale, "Local energy markets: Paving the path towards fully transactive energy systems," *IEEE Transactions on Power Systems*, 2018.

[6] X. Jin, Q. Wu, and H. Jia, "Local flexibility markets: Literature review on concepts, models and clearing methods," *Applied Energy*, vol. 261, p. 114387, 2020.

[7] J. Soares, T. Pinto, F. Lezama, and H. Morais, "Survey on complex optimization and simulation for the new power systems paradigm," *Complexity*, vol. 2018, 2018.

[8] F. Lezama, J. Soares, and Z. Vale, "Optimal bidding in local energy markets using evolutionary computation," in *International Conference on Intelligent System Application to Power Systems (ISAP)*, July 2019, pp. 1–8.

[9] K. Y. Lee and J. G. Vlachogiannis, "Optimization of power systems based on ant colony system algorithms: an overview," in *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*. IEEE, 2005, pp. 22–35.

[10] M. Huang, "Improved ant colony algorithm in the distribution of reactive power compensation device and optimization," *Procedia Engineering*, vol. 7, pp. 256–264, 2010.

[11] M. Mavrovouniotis, C. Li, G. Ellinas, and M. Polycarpou, "Parallel ant colony optimization for the electric vehicle routing problem," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2019, pp. 1660–1667.

[12] G. Chen, Z. Li, Z. Zhang, and S. Li, "An improved aco algorithm optimized fuzzy pid controller for load frequency control in multi area interconnected power systems," *IEEE Access*, 2019.

[13] S. Parsons, M. Marcinkiewicz, J. Niu, and S. Phelps, *Everything you wanted to know about double auctions, but were afraid to (bid or) ask*. Department of Computer & Information Science, University of New York, 2006.

[14] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE computational intelligence magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[15] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," *European journal of operational research*, vol. 185, no. 3, pp. 1155–1173, 2008.

[16] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.

[17] T. Stützle and H. H. Hoos, "Max–min ant system," *Future generation computer systems*, vol. 16, no. 8, pp. 889–914, 2000.

[18] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on evolutionary computation*, vol. 1, no. 1, pp. 53–66, 1997.

[19] B. Dogan and T. Olmez, "A new metaheuristic for numerical function optimization: Vortex search algorithm," *Information Sciences*, vol. 293, pp. 125 – 145, 2015.