SN Operations Research Forum (2020) 1: 20
https://doi.org/10.1007/s43069-020-00021-4

ORIGINAL RESEARCH

Particle Swarm Optimization for Computing Nash and Stackelberg Equilibria in Energy Markets

Michael N. Vrahatis[1] · Panagiotis Kontogiorgos[2] · George P. Papavassilopoulos[2]

Received: 2 March 2020 / Accepted: 3 August 2020 / Published online: 19 August 2020

Abstract

Interactions among stakeholders in deregulated markets lead to complex interdependent optimization problems. The present study is motivated by load control programs in energy markets and more precisely by using the power supply interruption as a tool for reducing consumers' demand voluntarily, also known as voluntary load curtailment programs. The problem is formulated as a Stackelberg game, specifically, as a bilevel optimization problem that belongs to the mathematical programs with equilibrium constraints. In this game, a player that acts as leader determines the actions of the players that act as followers and play a Nash game among them through a subsidy program. The corresponding equilibria need to be found and the presence of nonconvex functions makes the use of metaheuristic algorithms attractive. An extension of particle swarm optimization is proposed for solving such problems based on the unified particle swarm optimization that is a variation of the plain particle swarm optimization algorithm. The proposed algorithm is tested by solving some examples of the formulated games in order to study its efficiency and the interactions between the stakeholders of the market.

Keywords Particle swarm optimization · Stackelberg · Nash · Energy market · Bilevel programming · Demand response · Energy management system · Operations research · Management science

Abbreviations

ISO	Independent system operator
KKT	Karush-Kuhn-Tucker
LICQ	Linear independence constraint qualification

✉ Michael N. Vrahatis
vrahatis@math.upatras.gr

Panagiotis Kontogiorgos
panko09@hotmail.com

George P. Papavassilopoulos
yorgos@netmode.ntua.gr

Extended author information available on the last page of the article

MPEC	Mathematical program with equilibrium constraints
PSO	Particle swarm optimization
UPSO	Unified particle swarm optimization
VLC	Voluntary load curtailment

1 Introduction

Deregulation is currently considered to make markets more efficient. That is why it has been implemented in many fields resulting in complex interactions among stakeholders at all levels of a market. The interactions are usually modeled using game theory notions and these models are then solved by applying various optimization methods [1].

In simple problems, deterministic optimization methods can be used in order to find the optimal solution. However, in complex large-scale systems, the convergence to the required solution representing an optimal or some game equilibrium is not guaranteed since nonlinear and nonconvex functions are used; therefore, meta-heuristic algorithms become attractive. Several meta-heuristic evolutionary algorithms have been recently tested for solving complex energy management problems, including among others, artificial bee colony optimization [2], imperialist competition algorithm [3], and ant colony optimization [4], giving promising results. Some of these algorithms are based on collective intelligent behaviors in nature. In the present study, a variation of the standard *particle swarm optimization* (*PSO*) algorithm, which studies the collective behavior of simple interacting agents in small groups, is used. The proposed particle swarm optimizer is based on the *unified particle swarm optimization* (*UPSO*) that also belongs to the category of swarm intelligence algorithms. It is a stochastic algorithm and various tests have shown that it is a very promising algorithm for solving complex optimization problems [5].

The present study is motivated by energy markets where demand response is used in order to motivate consumers to modify their consumption patterns through price signals or other incentives [6, 7]. The game examined concerns the implementation of a subsidy program where a player of the market, who acts as leader, offers subsidies as an incentive to the consumers, which are the followers, in order to influence their actions. A category of such programs is the *voluntary load curtailment* (*VLC*) programs that have been implemented with success in electricity markets during the last years, especially in USA [8]. It has been shown empirically that these load control programs can reduce demand in high price periods or in various system security emergencies [9]. Moreover, many recent studies incorporate demand response programs in complex optimization problems regarding energy market dynamics in order to evaluate their influence on the price formation [10–12]. These experimental validations suggest that demand response can be used in complex real-time energy management systems. There are various energy market entities that are related to final consumers and could act as leader in a demand response program. For example, it could be a power producer that believes his bilateral power supply agreements with large consumers will not be met due to a failure, or a utility company that realizes its demand forecast error will result in high market penalties, but in most cases, it is a system operator that foresees a supply-demand imbalance and needs to take action to prevent a network's black-out. In modern liberalized markets, this entity informs about their request an aggregator. An aggregator has active contracts with consumers enabling him to manage their loads according to mutually agreed terms regarding the maximum load curtailment, the notice prior to curtailment and its duration, compensation terms etc. An aggregator represents a large number of final consumers, so if a system operator or another stakeholder requests a massive load curtailment and this request is financially profitable for final consumers, the aggregators implements the curtailment. Any energy market stakeholder that wishes to manage aggregated consumer loads can become an aggregator, regardless of whether they are related to power production, energy supply, or other energy services, meaning the leader requesting the load curtailment and the aggregator implementing it may be the same entity.

For a VLC program to be successfully implemented, the leader needs to design it in advance. This is the reason why the proposed methodology can be used as a decision support tool in order to analyze the consumers' response and design a suitable demand response program that will become effective if needed, specifically, in the case of real-time operation of a grid [13], system emergency or even long-term planning of investments and security of supply in energy markets [14]. In any case, the market deregulation and the appearance of demand-response programs makes the study of the stakeholders' interactions an interesting and promising area for developing efficient and effective algorithms for complex systems [15, 16].

The type of games used to model the above issue is the so-called *Stackelberg games* or *leader-follower games* and are modeled as *bilevel programming problems* with leaders being the upper level and followers being the lower level. A leader provides the followers with some pieces of information and wants to optimize his objective function by incorporating into his optimization problem the followers' reactions, who believe that the leader's decisions are exogenous and fixed. Stackelberg games are generally difficult to be solved due to nonconvexities that occur in both levels and especially the lower-level problems that are incorporated into the upper level. Many methodologies for solving them have been presented since the first algorithmic attempts [17–19] in order to exploit the specific structure of certain problems. The case with one leader and many followers belongs to the category of *mathematical programs with equilibrium constraints* (*MPEC*) [20] where the equilibrium constraints correspond to the *Nash game* played by the followers that try to optimize their costs based on the feedback from the leader. The results of such a problem may vary depending on whether energy market participants cooperate though coalition formations in supply or demand [21, 22] or not [23]. In the present study, noncooperative game theory is promoted as followers do not cooperate among themselves or with the leader, since in large scale demand response programs individual consumers usually cannot influence market operation except if very large coalitions are formed. There are algorithms for solving this kind of bilevel problems under specific assumptions [1, 24, 25] but even if these do not hold, we still seek the *Nash equilibrium* among the followers. In this way, it is possible to study their interactions, foresee the reactions resulting from a leader's decision and estimate a suboptimal solution for the Stackelberg leader-follower problem.

In the paper at hand, a PSO algorithm is proposed for solving the formulated bilevel programming problem and finding the corresponding Nash and Stackelberg equilibria in order to overcome limitations of the traditional mathematical programming framework. For this reason, the algorithm needs to converge to the optimal solution of each player by considering the other players' actions. In general, deterministic algorithms are not very efficient and effective when they are applied to large-scale complex problems that model the behavior of many market participants. Therefore, a metaheuristic PSO algorithm is used in order to solve the formulated Stackelberg problem of the VLC program implementation. Specifically, the UPSO algorithm is used and it was extended using a multiple-swarm approach in order to address equilibria problems, i.e., multiple swarms tackle different optimization problems simultaneously and every algorithmic iteration provides the sum of all the swarms' solutions as input to the next one. The proposed algorithm was also combined with *Lagrange multiplier methods* resulting in increased efficiency and effectiveness under certain assumptions. These algorithms were tested in various examples of VLC programs by comparing the optimization results with those obtained from a suitable solver of GAMS, that is, specialized software developed for modeling and solving complex optimization problems [26]. Each player in an energy market has his own characteristics that need to be taken into consideration. For this reason, we seek to find his profit and study if and how all the players can be benefited from the implementation of a VLC program. So far, due to lack of smart energy meters in low voltage consumers, such programs target mostly large industrial and commercial consumers with high and more easily controllable loads but the proposed methodology is scalable, thus, it is also applicable to large-scale problems involving residential consumers. The numerical results suggest that the proposed algorithm is efficient and effective especially as far as Nash equilibria are concerned but also that, through the implementation of a suitable VLC program the goal of reducing the total energy demand is achieved while there is a profit both for the leader and the followers. Therefore, this type of demand response programs could be widely used in order to increase system reliability and mitigate possible system risks.

The main contributions of the present paper are the following:

1. Development of a decision-support tool for energy market stakeholders that are interested in designing demand-response VLC programs.
2. The experimental implementation of an extended PSO algorithm towards supporting the decision-making process, with the following characteristics:
   (a) Capability of solving simultaneously large-scale interdependent optimization problems in order to seek game equilibria
   (b) Increased rate of convergence.
3. Conduct of numerical case studies in order to evaluate and validate the proposed framework and the effectiveness of VLC programs.

The rest of the paper is structured as follows: In Sect. 2, the mathematical formulation of the game is presented and the difficulties in solving it are described. In Sect. 3, the mathematical programming framework for solving bilevel programming programs is given. In Sect. 4, the standard UPSO algorithm is described along with the extensions made. In Sect. 5, the proposed algorithm is tested, and various VLC program schemes are examined by providing numerical results. Finally, in Sect. 6, a synopsis, concluding remarks and future extensions of this research are presented.

2 Formulation of the Game

For convenience of the reader, we provide the following synoptic lists which summarize the problem's functions, variables, and parameters:

Problem's functions:	
C_{pr}	Leader's production cost function (€)
C_{ex}	Leader's extra cost function (€)
F	Total offered fee function (€)
C_i	Comfort cost function of follower i (€)
F_i	Subsidy function of follower i (€)
Problem's variables:	
q_{pr}	Quantity produced (kWh)
q_{ex}	Quantity supplied at extra cost (kWh)
q_s	Total supply quantity (kWh)
q_d	Total demand quantity (kWh)
q_c	Total curtailed quantity (kWh)
r_1	Offered fee parameter (€/kWh)
$q_{d,\, i}$	Demand of follower i (kWh)
$q_{c,\, i}$	Quantity curtailed to follower i (kWh)
p	Market price (€/kWh)
Problem's parameters:	
$q_{\max}$	Maximum leader's production (kWh)
c_{pr}	Slope of leader's production function(€/kWh)
c_{ex}	Slope of leader's extra cost function(€/kWh)
q^*	Expected total demand at expected equilibrium (kWh)
q_i^*	Expected demand of follower i at expected equilibrium (kWh)
$c_{1,\, i}$	Comfort cost parameter of follower i (€/kWh)
n_i	Exponent related to follower's i comfort cost
$q_{\min,\, i}$	Minimum load needs of follower i (kWh)
M	Sufficiently large positive number
b_1	Supply function slope (€/kWh)

The equilibrium price and quantity in an energy market can be estimated before the clearing of the market using the expected supply and demand functions. Without a subsidy program and assuming linear market functions, if q represents the quantity of the energy, then the supply function is $f_s = b_1 q$, $b_1 \geq 0$ and the aggregated demand function of the consumers is expected to be $f_d = a_2 - b_2 q$, with a_2, $b_2 \geq 0$. In order the supply to meet demand, it must be hold that $f_s = f_d$; therefore, the expected equilibrium values of energy price and quantity are $p^* = b_1 \frac{a_2}{b_1 + b_2}$ and $q^* = \frac{a_2}{b_1 + b_2}$, respectively.

If the price is expected to rise above a threshold, there is need to interfere and move the equilibrium at a lower price for financial and stability reasons. For this purpose, a subsidy program can be used, such as a VLC program, meaning that the leader of the market can offer a fee to the consumers (followers) in order to incentivize them to reduce their energy demand voluntarily. In this way, the demand curve can be flattened, and peak prices are avoided. Attention must be given to the fact that the followers must be compensated based on historical data of their consumption to discourage them to act strategically. Moreover, such programs should be subject to regulation since they may postpone new investments in the energy sector.

In order to model a subsidy program, we need to take into consideration the costs and gains introduced to the players. The player that offers the subsidy, e.g., a *power producer* or the *independent system operator* (*ISO*) of the market, endows the consumers with an amount of money to prevent high prices in energy supply or/and production. As far as the consumers are concerned, there is a gain from the subsidy, on the other hand, a cost is also introduced. In the case of industrial and commercial consumers this cost represents the financial losses caused by the curtailment so it can be easily quantified, but in the case of residential consumers there is a comfort cost that is not just financial. *Comfort cost* is a term introduced in this study to describe the fact that a consumer hesitates to reduce his energy demand because then he will not be able to perform all the activities that he has planned and thus feeling uncomfortable. It is a qualitative criterion that needs to be quantified in order to be incorporated into mathematical models and optimization algorithms. For example, the *comfort cost function* can be calculated based on some measurable key indicators that are related to energy consumption such as indoor temperature, tasks that need to be shifted, etc. Comfort cost depends on the consumer, thus differentiating the consumers and their response to a subsidy program. For the subsidy program to be meaningful, the fee paid to the consumers must outweigh any cost induced.

If a subsidy program is offered through a total fee F that depends on a fee parameter r_1, then the Stackelberg game is formulated as a bilevel programming problem which is expressed mathematically as follows:

$$\min_{q_{pr},q_s,\mathrm{r}_1} \left\{C_{pr} + M\left(q_{\max} - q_{pr}\right) + C_{ex} - pq_s + F\right\}, \tag{1}$$

subject to

$$q_{pr} \leq q_{\max}, \tag{1a}$$

$$q_s = q_{pr} + q_{ex}, \tag{1b}$$

$$q_s + q_c = q^*, \tag{1c}$$

$$q_d = \sum_i q_{d,i}, \tag{1d}$$

$$q_c = \sum_i q_{c,i}, \tag{1e}$$

$$q_{pr},\ q_s,\ q_{ex} \geq 0, \tag{1f}$$

$$\min_{q_{d,i}}{}^{i} \left\{ p q_{d,i} + C_i - F_i \right\} \text{ subject to } C_i < F_i, q_{c,i} = q_i^* - q_{d,i}, q_{d,i} \geq q_{\min,i}, \forall i \tag{1g}$$

$$q_s = q_d, \tag{1h}$$

$$p = b_1 q_s. \tag{1i}$$

The inner objective functions correspond to the consumers. The variable $q_{d,\,i}$ denotes the quantity of energy demanded by the consumer i, $i = 1, 2, \ldots, n$ which is obtained at price p, C_i is his cost, F_i is the subsidy he receives, $q_{c,\,i}$ is the energy curtailed to the consumer i and q_i^* is his expected demand without subsidy. Each consumer has his own basic needs and preferences, so there is a certain amount of energy $q_{\min,\,i}$ he will not accept to cede. We assume that the cost is a polynomial function of $q_{c,\,i}$ and the subsidy is a linear function respectively with common fee parameter for all consumers, e.g., $C_i = c_{1,i} q_{c,i}^{n_i}$ and $F_i = r_1 q_{c,\,i}$ with $c_{1,\,i}$, $r_1 \geq 0$ and n_i integers.

The outer objective function corresponds to the leader. The production cost C_{pr} is assumed to be a linear function of the quantity produced q_{pr} and the extra cost C_{ex} a quadratic function of the quantity otherwise acquired expensively, q_{ex}, in case of emergency or failure, e.g., $C_{pr} = c_{pr} q_{pr}$ and $C_{ex} = c_{ex} q_{ex}^2$ with c_{pr}, $c_{ex} \geq 0$. The leader has a capacity limit $q_{\max}$ beyond which the supply cost increases faster and M is a sufficiently large positive number to prevent him from supplying energy expensively when it is not necessary. The total demanded and supplied energy is q_d and q_s respectively and its price is p. The total curtailed energy quantity is q_c and F is the total fee paid to the consumers as subsidy. The leader can also calculate the total expected demand q^* based on historical data. The constraints (1h) and (1i) are joint for both inner and outer problems, and they are derived from the market clearing conditions.

In order to simplify the mathematical formulation, the equation constraints are used to substitute many variables directly into the objective functions. In this way the variables q_s, q_{ex}, $q_{c,\,i}$, p can be eliminated from the problem, nevertheless they are useful in the decision support process.

The problem is a Stackelberg game between the leader and the consumers, who decide about the curtailed quantities by playing a Nash game among them that depends on the fee parameter announced by the leader and their cost functions. The interaction among the consumers' variables observed through joint constraints (1h) and (1i) leads to a generalized Nash equilibrium [27, 28]. Based on this formulation, the followers' game is assumed to converge in a unique equilibrium. In the case where all the above mentioned functions are convex, traditional mathematical programming techniques can be used in order to find the equilibria. On the other hand, this problem is generally difficult to be solved since the production cost, the consumers' costs and the subsidy may be nonconvex or/and nondifferentiable functions. The examples presented in Sect. 5 cover both cases to assess the proposed algorithms and study the players' interactions.

3 Mathematical Programming Framework

A *mathematical program with equilibrium constraints* (*MPEC*) is an optimization problem having as constraints other optimization problems that represent equilibrium conditions. The general form of an MPEC is as follows:

$$\min f(x, y), \tag{2}$$

subject to

$$x \in \Omega, \tag{2a}$$

$$y \in S(x), \tag{2b}$$

where $\Omega \subset \mathbf{R}^n$ and $S(x)$ is the solution set of the reactions of the other decision makers, which represents an equilibrium constraint. A bilevel programming problem is a special case of an MPEC where the constraint region of the upper level problem is determined implicitly by the solution set to the lower level problem. The MPEC formulated in Sect. 2 consists of the upper level optimization problem of the leader and the lower level interrelated optimization problems of the followers, namely the consumers, creating a nested structure.

The optimal solutions of the lower level equilibrium problem should satisfy the *Karush-Kuhn-Tucker* (*KKT*) conditions, assuming they can be meaningfully formulated [29, 30]. An optimization problem in the general form is stated as follows:

$$\min_x f(x), \tag{3}$$

subject to

$$h(x) = 0, \tag{3a}$$

$$g(x) \leq 0, \tag{3b}$$

where $x \in \mathbf{R}^n$ is the optimization real variable, $f(x)$ is the objective function and $h(x)$, $g(x)$ are the equality and inequality constraints respectively, while the KKT conditions are stated as follows:

$$\nabla_x f(x) + \sum_{i=1}^{k} \lambda_i \nabla_x h_i(x) + \sum_{j=1}^{m} \mu_j \nabla_x g_j(x) = 0, \tag{4}$$

$$h_i(x) = 0, \quad \text{for } i = 1, 2, \ldots, k, \tag{5}$$

$$g_j(x) \leq 0, \quad \text{for } j = 1, 2, \ldots, m, \tag{6}$$

$$\mu_j g_j(x) = 0, \quad \text{for } j = 1, 2, \ldots, m, \tag{7}$$

$$\mu_j \geq 0, \quad \text{for } j = 1, 2, \ldots, m, \tag{8}$$

where $f: \mathbf{R}^n \to \mathbf{R}$, $h = (h_1, h_2, \ldots, h_k) : \mathbf{R}^n \to \mathbf{R}^k$, $g = (g_1, g_2, \ldots, g_m) : \mathbf{R}^n \to \mathbf{R}^m$ are continuously differentiable real functions in the feasible region of x and $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$, $\mu = (\mu_1, \mu_2, \ldots, \mu_m)$ are the equality and inequality Lagrange multiplier vectors respectively, while ∇_x denotes the gradient with respect to x. The constraints in Eqs. (6)–(8) are known as complementarity conditions and they can be also expressed as follows:

$$0 \leq \mu \perp g(x) \leq 0. \tag{9}$$

For the KKT conditions to be meaningful, therefore necessary conditions, they need to meet a constraint qualification. Several constraint qualification criteria have been stated, some of them especially for MPECs. One of the simplest, which is also satisfied in our test problem, is the *linear independence constraint qualification* (*LICQ*) [31]. Moreover, the followers' problems are convex, so the KKT conditions are also sufficient for optimality. This means that in our problem, the lower level optimization problems constraining the upper level problem can be replaced by their corresponding KKT conditions.

Even after substituting the lower level optimization problems, the resulting problem has a nonconvex feasible region and is difficult to be solved, mostly because of the equality constraints in Eq. (7). There are however some techniques that can simplify this, including, among others, the *Fortuny-Amat McCarl linearization* [32]. Using this approach, complementarity constraints stated in Eq. (9) can be replaced by the following set of linear constraints:

$$0 \leq \mu \leq Mu, \tag{10}$$

$$0 \leq g(x) \leq M(1-u), \tag{11}$$

where u is binary and M a large enough constant in order to avoid numerical ill-conditioning. This technique transforms the MPEC into a mixed-integer programming problem that is easier to be solved.

Consequently, there are algorithms and techniques for solving bilevel programming problems under certain assumptions. However, traditional mathematical programming techniques cannot be applied in the case of nonconvex or/and nondifferentiable functions either in the upper or in the lower level optimization problems. Moreover, as the number of followers increases, the solutions' search space is expanded too. Therefore, a large-scale problem with many followers' results in a set of feasible solutions that cannot be practically calculated using complete enumeration of all the possible combinations, since it would be very demanding as far as computational resources are concerned [33]. This is another reason why a heuristic algorithm that can quickly converge to a solution close to the optimal should be used.

4 Particle Swarm Optimization

Since the first variant of PSO was introduced [34], many modifications have been proposed in order to improve the standard algorithm's behavior and convergence rate. PSO algorithms have the advantages and disadvantages associated with heuristics, but they have gained wide recognition and are being used in all kind of difficult optimization problems [5].

According to the plain version of the algorithm, a population of N particles is initialized in the search space A and then they move in it iteratively. Their position shift is called velocity u_i, $i = 1, 2, \ldots, N$ and their positions x_i are candidate solutions to the problem. Each particle can store the best position p_i it has visited, and all particles are informed about the best position p_g that they have generally visited. In every iteration, the velocity of each particle is updated based on information from the previous steps of the algorithm and by taking into consideration the general best position found so far. The respective mathematical formulation of the updates in velocity and position for each particle i are given in their vectorial form by:

$$u_i(t+1) = \chi\left[u_i(t) + c_1 R_1 (p_i(t) - x_i(t)) + c_2 R_2 \left(p_g(t) - x_i(t)\right)\right], \tag{12}$$

$$x_i(t+1) = x_i(t) + u_i(t+1), \tag{13}$$

where t is the iteration counter, R_1 and R_2 are random vectors with components uniformly distributed within the interval $[0, 1]$, c_1 and c_2 are weighting factors that affect the search ability of PSO by biasing the velocity update or by changing the magnitude of the search and χ is a parameter called constriction coefficient that acts as inertia weight [35].

In the study at hand, a variant of PSO is used called *unified particle swarm optimization* (*UPSO*) [36]. This variant combines the properties of the plain PSO with a modification that takes into consideration the neighborhood of each particle. Using the neighborhood means that each particle belongs to a smaller group of the swarm and some information is exchanged exclusively between the members of this group. Thus, Eq. (12) represents the global velocity update and the local best position p_l found from the members of the neighborhood is used similarly (instead of p_g) in order to calculate the local velocity update. In this study, particle segmentation into groups is based on their initialization sequence but any other criterion can be chosen arbitrarily, thus influencing the definition of local best. It should be noted that the global and local best positions are used in the velocity update and not in the position update itself, so as to attract the particles towards the respective promising areas of the search space without forcing them to move there and trap into possible local minima or miss other promising areas. In UPSO, the influence of the local and global velocity update, L_i and G_i respectively, is controlled by a real parameter u called unification factor. The velocity update for each particle is finally given by:

$$u_i(t+1) = uG_i(t+1) + (1-u)L_i(t+1). \tag{14}$$

The global component of the velocity update controls the *exploitation* properties of the algorithm whereas the local component is responsible for its *exploration* properties. UPSO is preferred in this study since it offers an extension that improves the default PSO algorithm. It is a very promising variant although the selection of u determines its efficiency and depends on the problem [36].

In order to avoid premature termination of the algorithm or unnecessary iterations after convergence, a suitable stopping criterion needs to be inserted into the algorithm. There are several stopping criteria options for stochastic algorithms, including maximum number of iterations, the number of function evaluations and tolerance, among others [37]. In the present study, the maximum number of iterations is used so as to study the particles' iterative movement and then an adaptive stopping criterion based on the change of the objective function is added in order to improve the execution time of the algorithm.

The default PSO and UPSO algorithms tackle unconstrained optimization problems. One way to address our constrained problem is to replace the objective function with a nonstationary penalty function to avoid infeasible solutions [5, 38]. Penalty functions are one of the most usual methods for addressing constrained problems since no assumptions on the continuity and differentiability are required. For a general constrained optimization problem,

$$\min f(x), \tag{15}$$

subject to

$$g_i(x) \leq 0, \quad i = 1, 2, \ldots, k, \tag{15a}$$

the penalty function used in this study is defined as

$$F(x) = f(x) + h(t)\, H(x), \tag{16}$$

where $f(x)$ is the original objective function, $h(t)$ is a penalty value depending on the iteration and $H(x)$ is a penalty factor of the form

$$H(x) = \sum_k \left[\theta(q_i(x))\; q_i(x)^{\gamma(q_i(x))}\right], \tag{17}$$

where $q_i(x) = \max\{0, g_i(x)\}$, $\theta(q_i(x))$ is a multi-stage assignment function and $\gamma(q_i(x))$ is the power of the penalty function.

Penalty functions are widely accepted and used but also have two considerable disadvantages. Specifically, they usually have a slow convergence rate and most importantly large values of the penalty value $h(t)$ could cause ill-conditioning. That is why under convexity assumptions another variation of UPSO is tested in the present paper that combines particle swarm optimization with *Lagrange multiplier methods*, in which the penalty idea is merged with the primal-dual and Lagrangian philosophy [29]. In these methods, the penalty function is not added to the objective function but rather to the Lagrangian function resulting in the augmented Lagrangian function.

For the general optimization problem (3), using the quadratic penalty method, the *augmented Lagrangian function* L_c is given by:

$$\begin{aligned} L_c(x, \lambda, \mu) = f(x) + \lambda^T h(x) + \frac{c}{2}\|h(x)\|^2 \\ + \frac{1}{2c}\sum_j \left\{ \left(\max\left\{0, \mu_j + c g_j(x)\right\}\right)^2 - \mu_j^2 \right\}, \end{aligned} \tag{18}$$

where λ and μ are the equality and inequality Lagrange multiplier vectors respectively, j denotes the jth coordinate of μ and c is a positive penalty parameter.

The method consists of solving a sequence of problems of the form:

$$\min L_{c^k}\left(x, \lambda^k, \mu^k\right), \tag{19}$$

subject to:

$$x \in X, \tag{19a}$$

where initial values are given to c^0 and vectors λ^0, μ^0 and the sequences are updated according to

$$0 < c^k < c^{k+1}, \forall k, c^k \rightarrow \infty, \tag{19b}$$

$$\lambda^{k+1} = \lambda^k + c^k h \left[x\left(\lambda^k, \mu^k, c^k\right) \right], \tag{19c}$$

$$\mu_j^{k+1} = \max\left\{ 0, \mu_j^k + c^k g_j \left[x\left(\lambda^k, \mu^k, c^k\right) \right] \right\}. \tag{19d}$$

The selected initial values determine the efficiency of the algorithm. Also, it is not necessary to increase c^k to infinity for the method to be converged. In this way ill-conditioning can be avoided and by taking also into account the improved convergence rate, this method can be superior to simple penalty functions. For solving these sequential optimization problems, the UPSO algorithm is used. The advantage as already mentioned is that the particle swarm algorithms can deal with any objective function even though the multiplier method imposes certain assumptions in order to be effective.

The plain PSO algorithm and subsequently the UPSO variant can solve a single optimization problem thus they need to be extended in order to address problems that seek equilibria of Nash and Stackelberg games with many players. In the case of Nash games, players decide simultaneously based on other players' strategies. Moreover, in our problem all the players' optimization problems are interdependent since their decisions affect the aggregated energy demand and subsequently the price. Therefore, we assume that each consumer is informed about the aggregated demand of all consumers and then the inner optimization problems described by Eq. (1g) are solved simultaneously by different groups of swarms given the fee parameter. Each consumer knows his own expected demand and decisions so, given the aggregated demand, he can calculate the sum of other players' decisions and optimize his own objective function, then, each iteration yields an updated aggregated energy demand and price so that he can take it again into consideration and adjusts his decision. This process reminds of a dynamic game and is repeated until all the consumers' decisions converge to Nash equilibrium. The hypothesis of informing a consumer about the aggregated demand instead of each other player's decision separately is also consistent with the operation of an energy market since such information are available to market participants.

This approach could also be used effectively in Stackelberg games. The inner optimization problems can be solved with multiple interdependent swarms as it is already described, and the results will be available to the leader who can then decide to change the offered fee. This process will be continued iteratively until the leader is satisfied from the outcome. However, this would be more of a trial and error approach instead of an optimization algorithm. Our purpose is to compare the proposed methodology with the mathematical programming framework for bilevel problems; thus, the inner optimization problems were substituted by their KKT conditions as they are described in Sect. 3 and the resulting constrained optimization problem was solved using the UPSO algorithm. As in mathematical programming, the resulting problem is difficult to be solved especially due to constraints of Eq. (7). To simplify this issue, some conditions were inserted in order to exploit the structure of the specific problem by calculating some Lagrange multipliers and variables that are easy to be decided without the need for swarm optimization (e.g., quantity produced q_{pr} is cheaper than q_{ex}; therefore, it is equal either to the aggregated consumers' energy demand or to $q_{\max}$ depending on whether the constraint (1a) is binding or not).

In summary, in Fig. 1, the high-level flowchart of the proposed method is exhibited that presents the sequence of the steps of the algorithm.

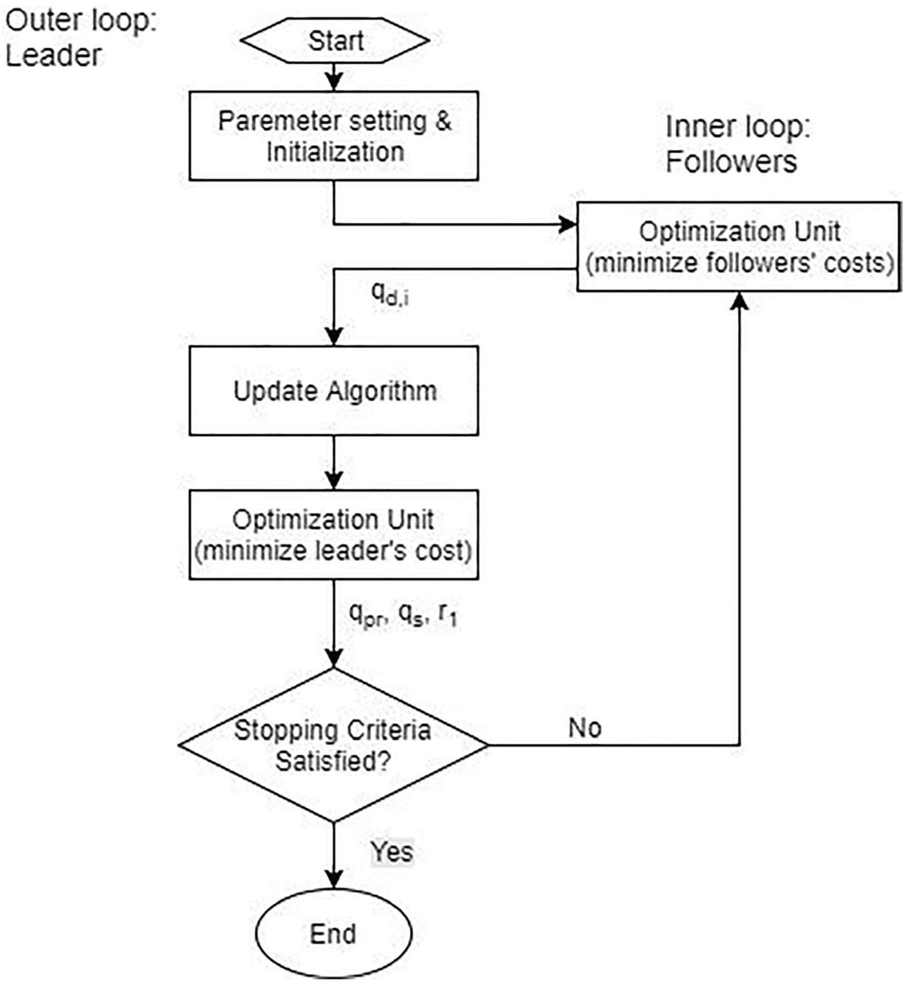

Fig. 1 High-level flowchart of the proposed algorithm

5 Results

In this section, we study the effectiveness of the proposed algorithm by solving some numerical examples and then we compare these solutions with the optimal solutions obtained by the mathematical programming techniques presented in Sect. 2 (expressed to three decimal places). Moreover, by studying the solutions we can deduce if a VLC program would be effective in an energy market and how each player is affected by it.

The parameters of the UPSO are selected as in the contemporary standard PSO [36], and the execution is performed with the unification factor $u = 0.5$ so that the algorithm is balanced between its local and global components. The neighborhood radius is assumed to be 1 and the swarm size equal to 10. Additional examples with different parameters can be performed in order to study how they affect the efficiency and effectiveness of the algorithm for this particular problem. In the examples, the modified penalty UPSO resulting from Eqs. (16)–(17) was used, while the values of the penalty parameters were based on results of other studies [5, 38, 39]. However, the penalty value $h(t)$ used was increased through trial and error, trying to avoid both ill-conditioning and constraint violations that were initially observed. In the examples that were solved using the modified multiplier UPSO of Eq. (19), the quadratic penalty function was used as already mentioned in Sect. 4. The maximum number of iterations is set to 800 and the added stopping criterion terminates the execution of the algorithm if the change of the objective function is less than 10^{-5} for 100 consecutive iterations. However, the stopping parameters can be modified according to the quality of the obtained results.

Firstly, some examples for the case of Nash equilibria are presented. We assume that the leader just wants to study the interactions among the followers without optimizing his objective function. In this case he can experiment with various fee schemes that will lead to different demand response results from the consumers. Therefore, we need to solve the game of the followers by seeking Nash equilibrium. In these examples, we present only the results for the decision variables $q_{d,\ i}$ since the rest can be easily calculated afterwards. The algorithm is stochastic and therefore the resulting values correspond to the average solutions of the considered 20 experiments.

Next, two examples of a Stackelberg game are presented, where one leader seeks to optimize his fee offer based on the reactions of two consumers. In these examples, the decision variables are the fee parameter, each consumer's demand and the amount of the generated energy. Based on these, the optimal or near optimal cost J of the leader can be deduced. Finally, a more complex problem with one leader, fifteen followers and two different fees is solved. Markets have a lot of participants that can be categorized according to their characteristics [40]. In energy markets, demand response programs are more usually offered to large consumers such as industries and commercial buildings, which are fewer but accountable for a large percentage of total consumption. These types of customers can be grouped in a few categories based on their sector and activity. Therefore, the last example is more realistic and useful in studying the players' interactions. However, as already mentioned the massive roll-out of smart meters in low voltage customers will render possible the implementation of such programs to residential customers. In this case, segmentation results in more categories since residential energy usage depends on location, building, and equipment characteristics, as well as demographic data.

The GAMS solver was used in order to compare the results with those of the mathematical programming framework (in the case that it was possible). Specifically, the DICOPT was used that is a program for solving *mixed-integer nonlinear programming* (*MINLP*) problems. The MINLP algorithm inside DICOPT solves a series of *nonlinear programming* (*NLP*) and *mixed-integer programming* (*MIP*) sub-problems using other solvers like CONOPT and CPLEX, respectively [41].

5.1 Nash Equilibrium with Two Consumers

In this case, it is assumed that there are two followers, the expected total demand is 12 (6 for each consumer) and that $b_1 = 10$. Moreover, it is assumed that the fee is given by a linear function; therefore, $m = 1$. The parameters that were used in *Example 1* are $r_1 = 11$, $c_{1,1} = 4$, $c_{1,2} = 2.5$, $n_1 = n_2 = 1$, $q_{\min,1} = 4.1$, $q_{\min,2} = 3.7$. Respectively in *Example 2* the parameters are: $r_1 = 10$, $c_{1,1} = 7$, $c_{1,2} = 5$, $n_1 = 1$, $n_2 = 2$, $q_{\min,1} = 3$, $q_{\min,2} = 3.2$, while in *Example 3*, they are $r_1 = 6$, $c_{1,1} = 3.5$, $c_{1,2} = 4$, $n_1 = n_2 = 2$, $q_{\min,1} = q_{\min,2} = 2.5$. In these simple examples, linear and nonlinear consumer cost functions are tested using both the UPSO penalty variant and the UPSO multiplier variant proposed since all assumptions hold, and the results of the algorithm are compared with those obtained using GAMS. For both UPSO algorithms, the same penalty value was used. The initial values of λ^0, μ^0 were given arbitrarily.

In Table 1, the resulting values of demand variables $q_{d,i}$ are presented. We observe that both proposed algorithms converge very close to the optimal solutions obtained from GAMS and the differences can be considered insignificant especially for the multiplier variant. Using the same hardware, the specialized software solved the problem in 2.013 s, the penalty UPSO algorithm needed 4.168 s and the multiplier UPSO algorithm 3.839 s respectively to perform the considered 20 experiments.

The standard deviation of the results obtained from the first variant is very small, meaning that we can use this algorithm in order to obtain a solution very close to the optimal with less experiments or even with only one execution of the algorithm. The second variant converges always to the same solutions setting tolerance to 0.001; therefore, only one execution is needed to converge in only 0.192 s. The UPSO algorithms' execution time is reduced by circa 70% after the implementation of the adaptive stopping criterion, meaning that most iterations were not necessary since the algorithm converges very quickly to an almost optimal solution. These observations

Table 1 Results of Nash game with two consumers

Method	Example 1	Example 2	Example 3
Modified penalty	$q_{d,1} = 4.108$	$q_{d,1} = 2.998$	$q_{d,1} = 4.289$
UPSO	$q_{d,2} = 3.710$	$q_{d,2} = 4.007$	$q_{d,2} = 4.486$
Modified multiplier	$q_{d,1} = 4.101$	$q_{d,1} = 3.001$	$q_{d,1} = 4.287$
UPSO	$q_{d,2} = 3.701$	$q_{d,2} = 4.001$	$q_{d,2} = 4.501$
GAMS	$q_{d,1} = 4.100$	$q_{d,1} = 3.000$	$q_{d,1} = 4.286$
solver	$q_{d,2} = 3.700$	$q_{d,2} = 4.000$	$q_{d,2} = 4.500$

indicate that the proposed algorithms are effective and very efficient and they could even be used to seek Nash equilibria in real-time problems.

5.2 Nash Equilibrium with Five Consumers

The main advantage of the developed penalty UPSO algorithm is that it can solve nonconvex, large-scale, complex problems without the need for continuity or differentiability assumptions as far as the objective functions are concerned. In the next example, we seek the Nash equilibrium among five consumers, each one expecting to demand 6 units of energy. Therefore, in *Example 4*, the expected total demand is 30 and b_1 is considered to be 10 again. The rest of the parameters are: $r_1 = 22$, $q_{\min,\,1} = 2,7$, $q_{\min,\,2} = 3$, $q_{\min,\,3} = 3.3$, $q_{\min,\,4} = 3.6$, $q_{\min,\,5} = 3.9$ and the comfort cost C_i for each consumer i is given by:

$$C_i = \begin{cases} 4.5q_{c,1}, & q_{c,1} < 1, \\ 5, & 1 \leq q_{c,1} \leq 2, \\ 8q_{c,i}^2, & q_{c,1} > 2. \end{cases}$$

The average along with the most frequent solutions out of the 20 experiments is presented in Table 2.

In this example, the consumers have the same expected demand and comfort cost and they differ only as far as their minimum needs are concerned. Observing the most frequent solutions for $q_{d,\,1}$ and $q_{d,\,2}$, it is implied that the optimal solution for consumers 1 and 2 of this example is 3.250 since $q_{\min,\,\iota}$ is not a binding constraint. For consumers 3 and 4 the optimal solutions are equal to their basic needs since $q_{\min,\,\iota}$'s become binding for them, whereas consumer 5 has increased energy needs and therefore the third branch of his cost function becomes an unprofitable choice.

It is also observed that the average solutions are greater compared to the respective most frequent ones. This happens probably because of the stricter penalty applied in order to avoid constraint violations; however, the differences are not significant and the algorithm generally converges very close to the optimal solutions.

5.3 Stackelberg Equilibrium with One Leader and Two Followers

In these examples, we seek to find the Stackelberg equilibrium optimizing the leader's and the followers' objective functions at the same time. Therefore, except for the

Table 2 Results of Nash game with five consumers

Example	Average solutions	Most frequent solutions
Example 4	$q_{d,\,1} = 3.314$	$q_{d,\,1} = 3.250$
	$q_{d,\,2} = 3.336$	$q_{d,\,2} = 3.250$
	$q_{d,\,3} = 3.427$	$q_{d,\,3} = 3.300$
	$q_{d,\,4} = 3.647$	$q_{d,\,4} = 3.600$
	$q_{d,\,5} = 4.005$	$q_{d,\,5} = 4.000$

consumers' demand, the leader needs to decide the quantity produced and the fee offered. We assume again that the expected total demand is 12 (6 for each consumer), $b_1 = 10$ and $m = 1$. The penalty UPSO algorithm is executed once and the results are compared with those obtained from GAMS using the mathematical programming techniques presented in Sect. 3.

The parameters that were used in *Example 5* are $c_{pr} = 8$, $c_{ex} = 50$, $q_{\max} = 8$, $c_{1,1} = 5.5$, $c_{1,2} = 6.5$, $n_1 = n_2 = 2$, $q_{\min,1} = 3$, $q_{\min,2} = 3.5$. Respectively, in *Example 6*, they are: $c_{pr} = 10$, $c_{ex} = 40$, $q_{\max} = 7$, $c_{1,1} = 2.5$, $c_{1,2} = 4$, $n_1 = n_2 = 2$, $q_{\min,1} = 4$, $q_{\min,2} = 2$. The results as far as the players' decisions and the leader's respective cost J are presented in Table 3.

We observe that the results for q_{pr} coincide since this decision can be derived easily from a condition taking into account the constraint (1a). In Example 5, the consumers' decisions and the leader's resulting cost have minor differences in both solutions but the fee parameter r_1 differs by almost 10%. On the other hand, in Example 6, the two solutions can be considered identical. In Fig. 2, it is presented how the swarm of Example 6 converges to the final value of r_1. The swarm particles that correspond to different colors initiate from random positions in the search space and they move closer to the optimal value after every iteration of the algorithm until they converge.

The results suggest that the proposed algorithm is effective also in Stackelberg games with one leader, but in some cases, there could be errors as far as the values of some decision variables are concerned. This could be due to local minima that are very close to the global one, since the problem is very complex and includes nonlinearities and nonconvexities. However, even in these cases, the leader would have a cost very close to the optimal one. The algorithm's maximum number of iterations in these examples was doubled since the problem is more complex, and the execution time is circa 9 s while GAMS needed 2.42 s, verifying that Stackelberg games are more difficult to be solved than Nash games. The insertion of the adaptive stopping criterion did not improve the execution time because the particles seem to oscillate around the best position. The variations are minor, so by relaxing the stopping criterion to a change of 10^{-3} leads to convergence in circa 3.5 s. Therefore, in this type of games, the algorithm is not as efficient as in previous examples, but its main advantage is that it

Table 3 Results of Stackelberg game with one leader and two followers

Method	Example 5	Example 6
Modified penalty UPSO	$q_{pr} = 8$	$q_{pr} = 7$
	$q_{d,1} = 5.088$	$q_{d,1} = 4.439$
	$q_{d,2} = 5.228$	$q_{d,2} = 5.024$
	$r_1 = 5.017$	$r_1 = 3.903$
	$J = -723.650$	$J = -572.927$
GAMS solver	$q_{pr} = 8$	$q_{pr} = 7$
	$q_{d,1} = 4.992$	$q_{d,1} = 4.439$
	$q_{d,2} = 5.147$	$q_{d,2} = 5.024$
	$r_1 = 5.545$	$r_1 = 3.902$
	$J = -724.909$	$J = -572.927$

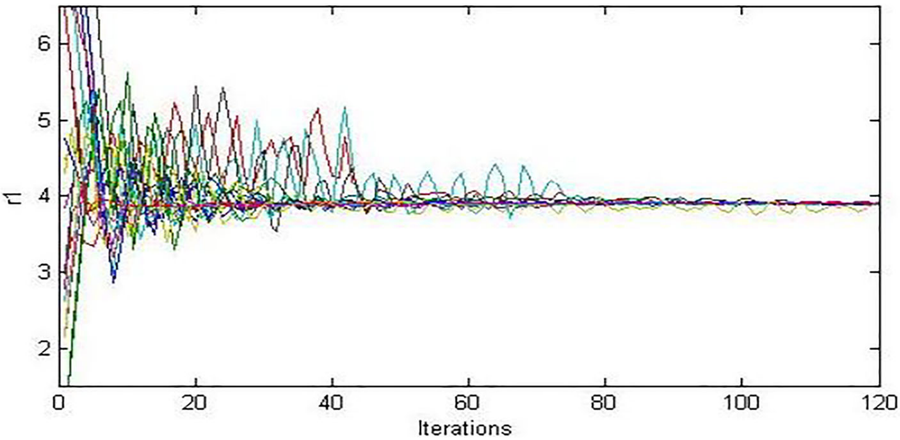

Fig. 2 Swarm convergence regarding r_1 in Example 6

can be used even if the leader's objective function is nonconvex or nondifferentiable like that in Sect. 5.2.

The UPSO and the penalty function parameters used were obtained from general tests in various optimization problems so proper configuration of the algorithm for this type of problems could improve it furthermore. The costs of each player depending on whether the VLC program is implemented or not, are presented in Table 4 where the cost J corresponds to the leader and the costs J_1, J_2 correspond to the two followers respectively. No price elasticity was assumed. Therefore, the two consumers without a VLC program would have the same costs since they are expected to have the same demand. In this case, leader's cost depends on parameters c_{pr} and c_{ex}. However, if a VLC program is implemented, two observations can be made. Firstly, consumers are motivated to reduce their demand depending on the fee offered. This reduction depends on their basic needs and their cost function, thus differentiating the consumers and this is something the leader should take into consideration. Secondly, the leader is benefited too by the curtailment since he will not have to supply very expensive energy. In case the leader's production can satisfy all the expected demand, he does not need to implement a VLC program and the result would be an equilibrium with less demand and at a lower price.

Table 4 Players' costs with and without a VLC program

Method	Example 5	Example 6
Without VLC	$J_1 = 720$	$J_1 = 720$
	$J_2 = 720$	$J_2 = 720$
	$J = -576$	$J = -370$
With VLC	$J_1 = 506.096$	$J_1 = 420.083$
	$J_2 = 521.823$	$J_2 = 475.479$
	$J = -724.909$	$J = -572.927$

5.4 Stackelberg Equilibrium with One Leader and Many Followers

In practice, the number of followers participating in a VLC program may be quite large. Additionally, the leader may prefer to offer different fees according to the participants' characteristics. The scalability of the proposed methodology makes it possible to solve also more complex problems with a lot of followers. In the following example, we assume that there are fifteen followers some of them with quadratic and some others with linear cost functions. There are also two fee parameters: r_1 for followers with small expected demand and r_2 for those who have large expected demand and therefore they can be considered as more important for the success of the VLC program. The parameters that were used in *Example 7* are $c_{pr} = 8$, $c_{ex} = 50$, $q_{\max} = 40$ the inputs and outputs for each follower are presented in Table 5.

As far as the leader is concerned, $q_{pr} = 40$, $q_{ex} = 36.725$ and his cost after implementing the VLC program is equal to 9034.800 instead of 96,960 that would be otherwise. Moreover, the market price and the total energy demand also dropped by more than 26% because of the VLC program. As far as the fees are concerned $r_1 = 6$ and $r_2 = 5$, implying that followers with large expected demand could be satisfied with a smaller fee since they are compensated for larger curtailed quantities; however, more examples need to be studied in order to conclude about the followers' behavior. One additional observation is that all the players (both the leader and the followers) can be benefited by the implementation of a suitable VLC program.

Table 5 Example of a Stackelberg game with one leader and 15 followers

Player i	Player's characteristics				Results		
	Cost C_i	Fee F_i	$q_{\min,\, i}$	q_i^*	$q_{d,\, i}$	J_i without VLC	J_i with VLC
Follower 1	$0.1q_{c,1}^2$	$r_1q_{c,\,1}$	0.5	2	0.500	2080	374.850
Follower 2	$0.1q_{c,\,2}$	$r_1q_{c,\,2}$	1	2	1	2080	761.350
Follower 3	$0.5q_{c,3}^2$	$r_1q_{c,\,3}$	0.1	1	0.100	1040	71.730
Follower 4	$1.5q_{c,4}^2$	$r_1q_{c,\,4}$	1	1	1	1040	767.250
Follower 5	$1.5q_{c,5}^2$	$r_1q_{c,\,5}$	2	4	2	4160	1528.500
Follower 6	$3q_{c,6}^2$	$r_1q_{c,\,6}$	1	3	1	3120	767.250
Follower 7	$3q_{c,\,7}$	$r_1q_{c,\,7}$	1	3	1	3120	767.250
Follower 8	$q_{c,8}^2$	$r_2q_{c,\,8}$	3	9	4	9360	3069.000
Follower 9	$q_{c,\,9}$	$r_2q_{c,\,9}$	3	9	3	9360	2307.750
Follower 10	$2.5q_{c,10}^2$	$r_2q_{c,\,10}$	4	10	8	10,400	6138.000
Follower 11	$2.5q_{c,11}^2$	$r_2q_{c,\,11}$	8	10	8	10,400	6138.000
Follower 12	$4q_{c,12}^2$	$r_2q_{c,\,12}$	6	12	10.750	12,480	8247.937
Follower 13	$5q_{c,13}^2$	$r_2q_{c,\,13}$	6	12	11	12,480	8439.750
Follower 14	$8q_{c,14}^2$	$r_2q_{c,\,14}$	5	13	12.375	13,520	9494.719
Follower 15	$8q_{c,\,15}$	$r_2q_{c,\,15}$	5	13	13	13,520	9974.250

6 Synopsis and Concluding Remarks

Market deregulation is a field where interactions among many stakeholders can be studied. These interactions lead to complex optimization problems that usually include game theory notions since the corresponding equilibria among the players need to be found. The interdependent optimization problems are very difficult to be solved using traditional optimization techniques, rendering metaheuristic algorithms more appealing.

In this study, the problem of subsidizing the consumers through a VLC program to reduce their energy demand was described and formulated. An extension of UPSO algorithm was used in order to solve this MPEC and calculate the corresponding Nash and Stackelberg equilibria. Also, the algorithm can be used in other bilevel problems as well. The proposed algorithm includes proper reformulation of the problem and multiple swarms that solve the players' optimization problems simultaneously, converging to an equilibrium without continuity or differentiability assumptions. A modification of UPSO algorithm was also proposed and used to calculate simple Nash equilibria resulting in even more promising results. On the other hand, further study is needed as to how the necessary assumptions for multiplier method restrict the main advantages of metaheuristic algorithms.

The numerical results presented suggest that the algorithm converges to a Nash equilibrium very efficiently and can also converge to a Stackelberg equilibrium. Suitable configuration of the algorithm for this type of problems by experimenting with the unification factor of UPSO as well as with the PSO algorithm's and penalty function's parameters could improve the results. Additionally, the results could be improved by properly configuring and applying other related computational intelligence methods (see, e.g., [42, 43]).

Moreover, based on the results both the leader and the consumers can be benefited from a VLC program, thus load curtailment can be considered an effective incentive mechanism in order to reduce demand peaks in energy markets. The developed decision support tool can be used in designing suitable VLC programs according to the number and type of consumers that are expected to participate. Complex large-scale demand response programs need proper planning, but simple VLC programs could be used even in real-time scheduling judging by the high rate of convergence.

Further research could address the uniqueness of the equilibria or to study the case whether the proposed algorithm can also be extended to address games with many leaders and many followers (equilibrium problems with equilibrium constraints (EPECs)). In these cases, the interactions among the players will be useful in studying large-scale and realistic games in energy markets with many players in each level.

Acknowledgments The authors would like to thank the anonymous reviewers for their helpful comments.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Gabriel SA, Conejo AJ, Fuller JD, Hobbs BF, Ruiz C (2013) Complementarity modeling in energy markets. International Series in Operations Research and Management Science 180. Springer Science & Business Media, New York
2. Marzband M, Azarinejadian F, Savaghebi M, Guerrero JM (2017) An optimal energy management system for islanded microgrids based on multi-period artificial bee colony combined with Markov chain. IEEE Syst J 11(3):1712–1722
3. Marzband M, Parhizi N, Savaghebi M, Guerrero JM (2016) Distributed smart decision-making for a multimicrogrid system based on a hierarchical interactive architecture. IEEE T Energy Conver 31(2): 637–648
4. Marzband M, Yousefnejad E, Sumper A, Domínguez-García JL (2016) Real time experimental implementation of optimum energy management system in standalone microgrid by using multi-layer ant colony optimization. Int J Electr Power Energy Syst 75:265–274
5. Parsopoulos KE, Vrahatis MN (2010) Particle swarm optimization and intelligence: advances and applications. Information Science Publishing (IGI Global), Hershey
6. US Department of Energy (2006) Benefits of demand response in electricity markets and recommendations for achieving them: report to US Congress pursuant to section 1252 of the Energy Policy Act of 2005. US Department of Energy, Washington DC
7. Federal Energy Regulatory Commission (2006) Assessment of demand response and advanced metering, staff report docket number AD-06-2-00. Federal Energy Regulatory Commission, Washington DC
8. Walawalkar R, Fernands S, Thakur N, Chevva KR (2010) Evolution and current status of demand response (DR) in electricity markets: insights from PJM and NYISO. Energy 35(4):1553–1560
9. Cappers P, Goldman C, Kathan D (2010) Demand response in US electricity markets: empirical evidence. Energy 35(4):1526–1535
10. Marzband M, Ghadimi M, Sumper A, Domínguez-García JL (2014) Experimental validation of a real-time energy management system using multi-period gravitational search algorithm for microgrids in islanded mode. Appl Energy 128:164–174
11. Marzband M, Parhizi N, Adabi J (2016) Optimal energy management for stand-alone microgrids based on multi-period imperialist competition algorithm considering uncertainties: experimental validation. Int T Electr Energy 26(6):1358–1372
12. Marzband M, Sumper A, Domínguez-García JL, Gumara-Ferret R (2013) Experimental validation of a real time energy management system for microgrids in islanded mode using a local day-ahead electricity market and MINLP. Energy Convers Manag 76:314–322
13. Marzband M, Moghaddam MM, Akorede MF, Khomeyrani G (2016) Adaptive load shedding scheme for frequency stability enhancement in microgrids. Electr Power Syst Res 140:78–86
14. Larsen ER, Osorio S, van Ackere A (2017) A framework to evaluate security of supply in the electricity sector. Renew Sust Energ Rev 79:646–655
15. Soliman HM, Leon-Garcia A (2014) Game-theoretic demand-side management with storage devices for the future smart grid. IEEE T Smart Grid 5(3):1475–1485
16. Su W-C, Huang AQ (2014) A game theoretic framework for a next-generation retail electricity market with high penetration of distributed residential electricity suppliers. Appl Energy 119:341–350
17. Papavassilopoulos GP (1980) Algorithms for leader-follower games. In: Proceedings of the 18th Annual Allerton Conference on Communication Control and Computing, pp 851–859
18. Papavassilopoulos GP (1982) Algorithms for static Stackelberg games with linear costs and polyhedra constraints. In: Proceedings of the 21st IEEE Conference on Decision and Control, vol 21, pp 647–652
19. Bialas WF, Karwan MH (1980) Multilevel optimization: a mathematical programming perspective. In: Proceedings of the 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes, vol 19, pp 761–765
20. Luo Z-Q, Pang J-S, Ralph D (1996) Mathematical programs with equilibrium constraints. Cambridge University Press
21. Marzband M, Ardeshiri RR, Moafi M, Uppal H (2017) Distributed generation for economic benefit maximization through coalition formation–based game theory concept. Int T Electr Energy 27:e2313
22. Andreou GT, Bouhouras AS, Milioudis AN, Labridis DP (2017) Energy efficiency in urban electrical grids through consumer networking. In: Computers and Operations Research Series, vol 8, Network Design and Optimization for Smart Cities, Gakis K, Pardalos PM (eds), World Scientific, Chapter 2, pp. 32–52

23. Marzband M, Javadi M, Domínguez-García JL, Moghaddam MM (2016) Non-cooperative game theory based energy management systems for energy district in the retail market considering DER uncertainties. IET Gener Transm Distrib 10(12):2999–3009
24. Bard JF (2013) Practical bilevel optimization: algorithms and applications. In: Nonconvex optimization and its applications series, vol 30, Springer
25. Dempe S (2010) Foundations of bilevel programming. In: Nonconvex optimization and its applications series, vol 61, Springer
26. McCarl BA et al (2013) McCarl expanded GAMS user guide, GAMS release 24.2.1. GAMS development corporation, Washington, DC, USA
27. Harker PT (1991) Generalized Nash games and quasi-variational inequalities. Eur J Oper Res 54(1):81–94
28. Facchinei F, Kanzow C (2007) Generalized Nash equilibrium problems. 4OR 5(3):173–210
29. Bertsekas DP (1999) Nonlinear programming. Athena scientific, Belmont
30. Luenberger DG, Ye Y-Y (2016) Linear and nonlinear programming. In: International series in operations research and management science series, vol 228, Springer
31. Wachsmuth G (2013) On LICQ and the uniqueness of Lagrange multipliers. Oper Res Lett 41(1):78–80
32. Fortuny-Amat J, McCarl B (1981) A representation and economic interpretation of a two-level programming problem. J Oper Res Soc 32(9):783–792
33. Schaefer TJ (1978) The complexity of satisfiability problems. In: Proceedings of the tenth annual ACM Symposium on Theory of Computing (STOC), pp 216–226
34. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks IV, pp1942–1948
35. Clerc M, Kennedy J (2002) The particle swarm – explosion, stability, and convergence in a multidimensional complex space. IEEE T Evol Comput 6(1):58–73
36. Parsopoulos KE, Vrahatis MN (2007) Parameter selection and adaptation in unified particle swarm optimization. Math Comput Model 46(1–2):198–213
37. Selvakumar AI, Thanushkodi K (2007) A new particle swarm optimization solution to nonconvex economic dispatch problems. IEEE T Power Syst 22:42–51
38. Yang JM, Chen YP, Horng JT, Kao CY (1997) Applying family competition to evolution strategies for constrained optimization. In: Angeline PJ, Reynolds RG. McDonnell JR, Eberhart R (eds), Evolutionary Programming VI. (EP 1997). Lecture notes in computer science 1213:201–211, Springer
39. Parsopoulos KE, Vrahatis MN (2002) Recent approaches to global optimization problems through particle swarm optimization. Nat Comput 1(2–3):235–306
40. Başar T, Srikant R (2002) A Stackelberg network game with a large number of followers. J Optim Theory Appl 115(3):479–490
41. GAMS–The Solver Manuals (2020) GAMS Release 30.1.10. GAMS Development Corporation, Washington, DC, USA
42. Parsopoulos KE, Vrahatis MN (2004) On the computation of all global minimizers through particle swarm optimization. IEEE T Evol Comput 8(3):211–224
43. Pavlidis NG, Parsopoulos KE, Vrahatis MN (2005) Computing Nash equilibria through computational intelligence methods. J Comput Appl Math 175(1):113–136

Affiliations

Michael N. Vrahatis[1] · **Panagiotis Kontogiorgos**[2] · **George P. Papavassilopoulos**[2]

[1] Computational Intelligence Laboratory (CILab), Department of Mathematics, University of Patras, GR-26110 Patras, Greece

[2] School of Electrical and Computer Engineering, National Technical University of Athens, GR-15780 Athens, Greece