Proceedings of the 6th Conference on Numerical Analysis (NumAn 2014)
September 2–5, 2014, Chania, Greece, pp.161–166, 2014

An Energy Market Stackelberg Game solved with Particle Swarm Optimization

Panagiotis Kontogiorgos[1], Elena Sarri[1], Michael N. Vrahatis[2], and George P. Papavassilopoulos[1]

[1] Department of Electrical and Computer Engineering, National Technical University of Athens (NTUA), Athens, Greece
[2] Department of Mathematics University of Patras, Patras, Greece

Abstract. Complex interactions between stakeholders in deregulated markets are formulated using game theory notions. This study is motivated by energy markets and addresses Stackelberg games with a leader that decides first his strategy and many followers, each one with his own characteristics. A static Stackelberg game corresponding to a Voluntary Load Curtailment (VLC) program for energy consumers is formulated. This leads to a bilevel programming problem that is generally difficult to solve, due to nonlinearities, nonconvexities that arise and the large dimensionality of the problem due to the existence of many followers. In these problems metaheuristic algorithms become attractive. In this study an algorithm for solving such problems is developed, using Particle Swarm Optimization (PSO) which is based on collective intelligent behaviors in nature and has gained wide recognition the last years. Some examples are then solved using the proposed algorithm in order to study its efficiency and the interactions between the players of this game.

Key words: Complex Systems, Energy Market, Stackelberg, Particle Swarm Optimization

1 Introduction

Market deregulation has been a priority in many sectors during the last few decades resulting in interactions among different stakeholders at all levels of the market. These interactions are modeled with use of game theory notions and solved with various optimization methods. Therefore, deregulated markets are complex systems that are generally difficult to solve.

This research has been co-financed by the program ARISTEIA, project name HEPHAISTOS and by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Thales. Investing in knowledge society through the European Social Fund.

If the modeling of a complex system contains nonlinear or nonconvex functions, the use of traditional optimization methods does not guarantee convergence to the optimal solution. In these cases, the use of metaheuristic algorithms becomes attractive. One category of these algorithms is based on swarm intelligence that studies the collective behavior of simple interacting agents in small groups. PSO belongs in swarm intelligence algorithms and is stochastic [10].

This study is motivated by deregulated energy markets and the problem addressed is the implementation of a VLC program. In a VLC program, the power producer announces a fee that will be paid to the consumers if they don't use an amount of energy for a certain period. According to their response, he chooses the best VLC scheme for him. This is a Stackelberg game where the power producer is the leader and the consumers are the followers. There are algorithms for solving these problems under specific assumptions [1,3]. However as the number of players increases and these assumptions don't hold, the problem becomes difficult to solve.

In this paper, a PSO algorithm is developed for solving a VLC program model. The solutions to some simple examples are compared with these obtained from GAMS and another example is solved in order to demonstrate the efficiency of the algorithm in complex problems. The paper is structured as follows: In Section 2, the VLC model and the mathematical formulation of the problem are presented and the difficulties in solving it are described. In Section 3, a standard PSO algorithm is described along with the modifications made to it so as to tackle this problem. In Section 4 the results are presented and compared. Finally, in section 5, we present the conclusions and future extensions of the research.

2 Mathematical Formulation

A Stackelberg game is formulated as a bilevel programming problem when the leader has an optimization problem and the followers' optimization problems are included in it as constraints. The problem to be solved can be formulated mathematically as:

$$min_{q_{pr},q_A,q_s,q_{c,i}\forall i}\left(c_3 q_{pr} + M\left(q_{max} - q_{pr}\right) + c_4 q_A^2 - p q_s + \sum_i r_1 q_{c,i}^m\right) \tag{1}$$

subject to:

$$q, q_A, q_s \geq 0 \qquad q_{pr} \leq q_{max} \qquad q_s = q_{pr} + q_A$$
$$q_s + q_c = q^* \qquad q_d = \textstyle\sum_i q_{d,i} \qquad q_c = \textstyle\sum_i q_{c,i}$$

and $\forall i, i = 1, ..., n$:

$$min^i_{q_{d,i},q_{c,i}}\left(p q_{d,i} + c_{1,i} q_{c,i}^{n_i} - r_1 q_{c,i}^m\right) \tag{2}$$

subject to:

$$c_{1,i} q_{c,i}^{n_i} < r_1 q_{c,i}^m \qquad q_{d,i} \geq q_{min,i} \qquad q_{c,i} = q_i^* - q_{d,i}$$

with $q_s = q_d$ and $p = b_1 q_d$ as joint constraints of the outer and inner problems

Positive cost parameters c_3 and c_4 correspond to the quantity produced q_{pr} and the quantity otherwise acquired expensively q_A in case of emergency or failure. The power producer has a capacity limit q_{max} and M is sufficiently large so as to prevent him from finding energy expensively when he can produce it on his own. The total demanded and supplied energy is q_d and q_s respectively and its price is p. The energy curtailed to a consumer i is $q_{c,i}$ and r_1 is the fee parameter defined by the producer. The energy curtailed along with the parameters r_1 and m form the fee function for each consumer. The producer can also calculate the total expected demand q^* based on historical data. For every consumer i, there is a comfort cost parameter $c_{1,i}$ that together with $q_{c,i}$ and n_i give his comfort cost function. The expected consumer's demand is q_i^*, whereas his final actual demand is $q_{d,i}$. Each consumer has some basic needs that are expressed with the lower limit $q_{min,i}$ for his demand. Finally, b_1 is the slope of the market's supply function used to calculate the energy price.

In order to simplify the mathematical formulation, the equations in the constraints can be substituted directly into the objective functions. Nevertheless solving this problem is difficult because a producer's cost, a consumer's comfort cost and the fee offered can be nonlinear, nonconvex or/and discontinuous functions. Moreover, the followers play a generalized Nash game and they all seek to minimize their objective functions taking into consideration the other consumers' decisions. The main difficulty lies in solving the followers' problem. Being able to estimate their decisions, the leader can easily optimize his objective function and redefine his strategy if necessary.

3 Algorithm Description

Since the first variant of PSO was first introduced [4,6], there have been developed many modifications to the standard algorithm in order to improve its behavior and convergence rate. PSO algorithms have gained wide recognition and are being used in all kind of difficult optimization problems [10]. In PSO, a population of N particles is initialized in the search space and then they move in it iteratively. Their position shift is called velocity v_i and their positions x_i are candidate solutions to the problem. In this study, a variant of PSO called Unified Particle Swarm Optimization (UPSO) [8] is used. This variant modifies the standard PSO so that the neighborhood of each particle is also taken into consideration. The local and global components of the velocity update, L_i and G_i respectively, are given in their vectorial form by:

$$G_i\left(t+1\right) = \chi\left[v_i\left(t\right) + c_1 R_1\left(p_i\left(t\right) - x_i\left(t\right)\right) + c_2 R_2\left(p_g\left(t\right) - x_i\left(t\right)\right)\right] \quad (3)$$

$$L_i\left(t+1\right) = \chi\left[v_i\left(t\right) + c_1 R_1\left(p_i\left(t\right) - x_i\left(t\right)\right) + c_2 R_2\left(p_l\left(t\right) - x_i\left(t\right)\right)\right] \quad (4)$$

where $i = 1, 2, ..., N$, t the iteration counter, p_g the overall best position found so far, p_l the local best position for each neighborhood, R_1 and R_2 random vectors with components uniformly distributed within [0,1], c_1 and c_2 weighting factors and χ is a parameter called constriction coefficient.

The influence in UPSO of the local and global velocity update, is controlled by a parameter u called unification factor. The mathematical formulation of the updates in velocity and position are:

$$v_i (t+1) = uG_i (t+1) + (1-u) L_i (t+1) \tag{5}$$

$$x_i (t+1) = x_i (t) + v_i (t+1) \tag{6}$$

UPSO is a very promising variant although the selection of u influences its efficiency and depends on the problem [9].

The default PSO and UPSO algorithms optimize unconstrained problems. In order to address this constrained problem a penalty function is used so as to avoid infeasible solutions [7,11]. This is one of the most usual methods for solving constrained problems since it does not require assumptions on the continuity and differentiability of the objective function. If the constrained problem is: min $f(x)$, subject to $C_j (x) \leq 0$, $j = 1, 2, ..., k$ the penalty function is defined as:

$$F(x) = f(x) + h(t) H(x) \tag{7}$$

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

As mentioned, the followers play a Nash game where they all decide simultaneously, each one based on the strategies of the rest. Each consumer's optimization problem is solved with the UPSO algorithm by adding loop iterations so as to solve all these optimization problems simultaneously. The consumers' problems are interdependent since their decisions affect the demanded quantity of energy and therefore its price. For this reason, the algorithm has been modified so that after every iteration each consumer is informed about the total requested energy that corresponds to the others' decisions so that he can take it into consideration in his next iteration. This process, which reminds of a dynamic game, is repeated until the system becomes stable and the players' decisions do not change significantly.

4 Results

In all the examples we assume that the swarm size is 10. Since the algorithm is stochastic, we present the average results for 20 executions. The parameters of the UPSO are considered to be $c_1 = c_2 = 2.05$ and $\chi = 0.729$ [2] as in the contemporary standard PSO and the execution is performed with $u = 0.5$ so that the algorithm is balanced between the global and the local component. The neighborhood radius for each particle is assumed to be 1. The main variable is the final energy demand $q_{d,i}$ for each consumer i. We present only the results for these variables since the rest of the variables and the objective functions of all players are then easy to calculate.

As far as the penalty parameters are concerned, suitable values that are tested in various other experiments [7,11] are used.

4.1 Examples with two consumers

It is assumed for all examples that 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 so $m = 1$. In Example 1 the parameters used are: $r_1 = 3$, $c_{1,1} = 2$, $c_{1,2} = 1.5$, $n_1 = n_2 = 1$, $q_{min,1} = 4$, $q_{min,2} = 3.5$, in Example 2 they are: $r_1 = 8$, $c_{1,1} = 6$, $c_{1,2} = 5$, $n_1 = 1$, $n_2 = 2$, $q_{min,1} = 4$, $q_{min,2} = 3.5$ and in Example 3 they are: $r_1 = 5$, $c_{1,1} = 3$, $c_{1,2} = 4$, $n_1 = n_2 = 2$, $q_{min,1} = q_{min,2} = 3$. In these simple examples linear and nonlinear comfort cost functions are tested and the results of the algorithm are compared with these obtained from GAMS:

	Example 1	Example 2	Example 3
modified UPSO	$q_{d,1} = 4.0058$	$q_{d,1} = 4.02535$	$q_{d,1} = 4.32571$
	$q_{d,2} = 3,48812$	$q_{d,2} = 4.42601$	$q_{d,2} = 4.73722$
GAMS	$q_{d,1} = 4$	$q_{d,1} = 4$	$q_{d,1} = 4.333$
	$q_{d,2} = 3,5$	$q_{d,2} = 4.4$	$q_{d,2} = 4.75$

We observe that the algorithm converges almost to the optimal solution at about the same time a specialized software needs to find it. More precisely, the differences are between 0.1 and 0.6 percent. Moreover, the standard deviation of the results for 20 experiments is small; therefore we can find a satisfactory solution within 2% of the optimal with only one execution of the algorithm which is much faster.

4.2 Example with five consumers

The main advantage of the algorithm is that it can solve nonconvex, large scale, complex systems without continuity or differentiability assumptions for the objective functions. We now assume that there are five consumers, each one expecting to demand 5 units of energy. Therefore, the expected total demand is 25 and b_1 is considered to be 10 again.

If $r_1 = 8$, $q_{min,1} = 4.5$, $q_{min,2} = 4$, $q_{min,3} = 3.5$, $q_{min,4} = 3$, $q_{min,5} = 2.5$ and the comfort cost C of each consumer is given by: $C = \begin{cases} 4q_{c,i} & q_{c,i} < 0.6 \\ 6 & 0.6 \leq q_{c,i} \leq 1.1 \\ 7q_{c,i}^2 & q_{c,i} > 1.1 \end{cases}$, the average solutions are:

$$q_{d,1} = 4.52521 \; q_{d,2} = 4.02953 \; q_{d,3} = 3.84558 \; q_{d,4} = 3.89112 \; q_{d,5} = 3.89725$$

In this case, the first two consumers are constrained by their basic needs and the other three that are not, reduce their demand but don't accept large curtailment because the comfort cost becomes too high in the third branch of the comfort cost function. Similarly, cases with a lot of players can be modeled and solved. As the number of consumers and the complexity of the problem increases, standard deviation increases too. Therefore we can't rely on just one execution of the algorithm. Moreover, the penalty for violation of the constraints must be increased because in some cases they are underestimated.

5 Conclusions

A PSO algorithm was developed for solving complex optimization problems that result from interactions among various stakeholders in deregulated markets. The algorithm was based on UPSO and is able to address constrained optimization problems with many players that need to decide their strategies based on the other players' decisions. The study was motivated by energy markets and the test model used is a static Stackelberg game with a power producer as the leader and many consumers as followers.

Some examples were solved and the results show that the algorithm is effective and can generally solve complex problems that cannot be addressed with the traditional optimization methods. Further research could include the solution of more realistic models, so as to study the interactions between the stakeholders and also the configuration of the algorithm so that the parameters' values correspond to the specific structure of this problem rendering the algorithm even more effective.

References

1. Bard, Jonathan F. Practical bilevel optimization: algorithms and applications. Vol. 30. Springer, 1998.
2. Clerc, Maurice, and James Kennedy. "The particle swarm-explosion, stability, and convergence in a multidimensional complex space." Evolutionary Computation, IEEE Transactions on 6.1 (2002): 58-73.
3. Dempe, Stephan. Foundations of bilevel programming. Springer, 2002.
4. Eberhart, Russ C., and James Kennedy. "A new optimizer using particle swarm theory." Proceedings of the sixth international symposium on micro machine and human science. Vol. 1. 1995.
5. Homaifar, Abdollah, Charlene X. Qi, and Steven H. Lai. "Constrained optimization via genetic algorithms." Simulation 62.4 (1994): 242-253.
6. Kennedy James, and Eberhart Russell. "Particle swarm optimization." Proceedings of 1995 IEEE International Conference on Neural Networks. 1995.
7. Parsopoulos, Konstantinos E., and Michael N. Vrahatis. "Particle swarm optimization method for constrained optimization problems." Intelligent Technologies–Theory and Application: New Trends in Intelligent Technologies 76 (2002): 214-220.
8. Parsopoulos, Konstantinos E., and Michael N. Vrahatis. "UPSO: A unified particle swarm optimization scheme." Lecture series on computer and computational sciences 1 (2004): 868-873.
9. Parsopoulos, Konstantinos E., and Michael N. Vrahatis. "Parameter selection and adaptation in unified particle swarm optimization." Mathematical and Computer Modelling 46.1 (2007): 198-213.
10. Parsopoulos, Konstantinos E., and Michael N. Vrahatis. Particle swarm optimization and intelligence: advances and applications. Hershey: Information Science Reference, 2010.
11. Yang, Jinn-Moon, et al. "Applying family competition to evolution strategies for constrained optimization." Evolutionary Programming VI. Springer Berlin Heidelberg, 1997.