EVOLUTIONARY ALGORITHMS AND EVOLUTIONARY PARTICLE SWARMS (EPSO) IN MODELING EVOLVING ENERGY RETAILERS

Vladimiro Miranda
INESC Porto and FEUP
Porto, Portugal
vmiranda@inescporto.pt

Naing Win Oo
INESC Porto
Porto, Portugal
nwo@inescporto.pt

Abstract – This paper provides evidence that Evolutionary Particle Swarm Algorithms outperform Genetic Algorithms in deriving optimal strategic decisions for an Energy Retailer, in the framework of a complex simulation of a multiple energy market, based on an Intelligent Agent FIPA-compliant open source platform.

Keywords:* *Agents, Markets, Evolutionary Computation, Genetic Algorithms, Particle Swarm Optimization

1 INTRODUCTION

This paper presents a comparison among three evolutionary algorithms in the context of a multiple energy market simulation: two types of genetic algorithms and one EPSO – evolutionary particle swarm optimization algorithm. The simulation is carried out on a platform where intelligent autonomous agents evolve and interact.

The description of the Intelligent Agent platform is not the main purpose of the paper and therefore only the necessary description will be included. The context is of market competition between different types of energy, namely electricity, gas and heat distributed in a territory through physical independent network systems supplying the same potential consumers.

In electrical power distribution, even in systems that have evolved to regulated markets, the physical distribution function has been seen as a natural monopoly because competition among companies building networks in the same territory seemed unfeasible. In fact, this led to the unbundling of the distribution and the retailer functions, decoupling the physical network from the business.

However, technological evolution opened new conceptual avenues. In fact, with the emergence of gas mini and micro-turbines, as well as fuel cells, suddenly a consumer needing electricity could buy gas and make the conversion in house. This simple fact immediately generates the potential for a gas-electricity complex market and not simply two parallel yet separate markets of gas and electricity – a gas supplier could now be competitive in the electricity demand.

Furthermore, the development of efficient small conversion units allows the spread of distributed generation with co-generation, adding to the equation the demand for heat, either for industrial needs or for commercial and even residential buildings. On top of this, district heating networks selling heat are a long known reality and a gain gas, electricity and heat enter as variables or factors in a unified model: the energy market.

The consequence is that we may witness three parallel network systems growing in parallel in the same territory and fighting for the same consumers, and therefore we can no longer talk of natural monopoly for energy distribution – even if each distribution subsystem is still a natural monopoly for its type of energy.

This reality is being acknowledged in the world. It's no surprise that power companies have taken interest in gas companies in many countries. In those operating with regulated markets, the tendency is to have a single regulatory entity supervising simultaneously the gas and the electricity markets. The complexity of interactions is of high level and it is a daunting task to try to develop any mathematical model to describe the behavior in detail of such types of markets.

Simulation is the answer. This paper is based on the work done to develop a simulator for the activity along time of a multiple energy market based on Intelligent Agent technology. The simulator has been built on JADE [1], an Open Source Intelligent Agent platform that is FIPA [2] compliant. Twelve basic agents have been modeled, namely Residential, Commercial and Industrial consumer groups, Electricity Retail Supplier, Gas Retail Supplier, Heat Retail Supplier, Distributors for Gas, Energy and Heat, Regulator, Economy and Information Environment. Each agent has its own objectives, internal processes of decision and form of communication with other agents. It is from the interaction of individual agents that a complex behavior emerges [3][4], and this collective behavior mirrors the market behavior even in conditions difficult to define by mathematical models. The plug-in capacity of agent technology allows one to simulate a diversity of agents and to insert them in the platform with a minimum effort.

The paper concentrates on Retailers. A retailer Agent has an internal process of simulation of the market in order to evolve and adapt and formulate optimized decisions in a competitive context. It is a simulation inside the simulation. This simulating ability of Retailer

Agents is what gives them adaptive power and response capacity both to the economic environment changing conditions and to the progress or evolution of the strategies developed by competitors.

This simulation is performed using evolutionary computing models – and the objective of the paper is to present the comparison of the effectiveness of three algorithms to perform the duty of market simulation and decision making for a Retailer Agent.

This comparison is very meaningful and most appropriate for meta-heuristic approaches. In fact, the problem dealt with is extremely complex, it is dynamic in time, it is non-linear, it includes in its formulation discrete decisions, complex constraints and is governed by the application of rules, i.e., it is an impossible task to try to develop a mathematical model for its representation.

In such a complex context, the power of evolutionary models emerges. Therefore, it provides a good test for the performance comparison of several distinct evolutionary algorithms.

The optimization purpose inside a Retailer Agent is, at a given moment in time, to decide on the next steps to take: change tariffs/prices of service/energy, invest in conversion equipment, invest in new lines, give incentives to customers such as discounts, apply capital in advertising or change maintenance policy). This complex vector of decisions is supposed to maximize the expected profit of the company in the future, taking in account not only constraints but also the moves that the competition may be ready to make to counteract the Retailer strategy. A Retailer Agent, therefore, has inside itself not only a model for the behavior of the competitors but it has the ability to learn and evolve the models it has for the competitors, based on the evolution of their observed behavior.

If the internal simulation is good, i.e., finds the good tactical moves in each moment, at the end of the simulation period the Retailer Agent will display a winning or, at least, a successful strategy. In this paper, we take as a measure of success the net present value of accumulated profits during the simulation period.

We have run three times the same market simulation, with the same starting conditions, and kept all Retailers constant with a fixed strategy, except one, our probe, that is successively equipped with one of three Evolutionary Algorithms to derive the desired winning strategy in face of competition. As we shall show, the result is clear: the winning algorithm is one belonging to the family of EPSO – Evolutionary Particle Swarm Optimization. It provides a greater profit to its retailer than the two other algorithms, tested in the same conditions.

2 THE MARKET AND AGENTS

The experiments were made by simulating a territory represented in a GIS platform (Geographic Information System). The land is divided in squares or blocks, each block having consumers of different types (typically, residential, commercial and industrial). Crossing the territory there are networks serving the clients that buy different types of energy: electricity, gas or heat.

Grids for gas, electricity and district heating develop in the same territory, competing for clients. There are areas already developed and areas under development, not yet served by all the networks. There is, therefore, room for network expansion and conquer of new clients.

The paper reports results from the interaction of twelve market agents engaging in energy market simulation; among them, we find Residential, Commercial and Industrial consumer groups, Electricity Retail Supplier, Gas Retail Supplier, Heat Retail Supplier, Regulator, Economy and Information Environment.

Interaction among market actors is introduced with indirect communication through the Information Environment, as a way in which interactions between environment and individual market actors take place rotationally. Simple market actors such as Economy and Regulator perform simple duties such as a) requesting information from the Information Environment, b) performing their typical duty and c) returning new information back to the Information Environment. But the process is more complicated when evaluation and optimization tasks are added for complex actors such as consumer actors and delivery actors, who must analyze information for better understanding of other actors' motives and goals, forecast future market behavior, make decisions based on predictions and learn and evolve from experience. A detailed explanation and mathematical models inside fundamental entities will not be provided in the paper and the reader must refer to future publications.

The basic functions in each agent are:

A – Economy – This agent translates into energy demand variables basic data such as economy, season of the year, weather conditions. These demand values are passed to the Information Environment Agent.

B – Consumer – Agents of this type represent not individual consumers, but rather classes of consumers such as residential, commercial or industrial. Each agent purchases a mix of energies and changes market shares of these energies according to prices, needs, elasticity of demand and adjustment delays to price changes. Energy efficiency is also taken in account as well as costs of capacity increments do increase purchases of a given type of energy.

C – Information Environment – This agent acts as a blackboard in which all available information from market players is presented and exchanged. It can be seen as an intermediate in which market actors post information regarding with their current actions and request information for evaluating new actions. Apart from communication purposes, it also performs compilation on the data obtained from market participants for providing more clear and transparent information.

D – Energy Retail – Every agent of this type has internal functions a) monitoring its performance in terms of profitability as well as market share movement, b) finding optimal decision combinations for performance improvement, and c) improving management efficiency. Achieving maximum profit while providing reliable service to consumer is the ultimate goal for a profit-oriented energy retail supplier. However, the level of reliability may, in some implementations, depend on the actions of the regulator and reactions of clients.

One important function inside a Retail agent is strategic planning. An agent of this type may use neural networks to predict consumptions and prices and uses evolutionary computing simulation to plan ahead and derive an optimal strategy both for expansion of the business and for price determination.

E – Delivery – these agents perform duties such as extending networks over the territory to supply new consumers, under request from the Retail agents. Network expansion is performed using functions optimizing paths and profits, which are also available in GIS platforms. An agent of this type has a logic of its own and also seeks to maximize profit while guaranteeing contracts of supply. The action of these agents puts energies in competition, because they allow consumers to have choices among forms of energy.

F – Regulator – In our simulations, the regulatory agent imposes simple restrictions such as limiting duration between successive product price movements and imposing price-cap over price of energy products.

3 EVOLUTIONARY META-HEURISTICS

One will compare the performance of a GADC – Genetic Algorithm using Deterministic Crowding as selection method to preserve diversity, a GAMP – Genetic Algorithm with multiple populations also to preserve diversity and allow good space exploration, and EPSO – Evolutionary Particle Swarm Optimization [5].

Comparisons are made in terms of outcome (profitable strategies discovered for the Retailer) and in terms of computing effort and precision, for a diversity of experiments including different elasticity indices for consumer reaction to price changes in the market. The simulations demonstrate the superiority of EPSO in all cases, suggesting that this is a more efficient method to be adopted in these kinds of complex problems.

We will not describe the basics of Genetic Algorithms, assuming that the principles governing their action are now well known – be we will refer to the special characteristics of the algorithms experimented. We will male a longer explanation about EPSO because it is a less known technique.

3.1 GADC

The first essay uses GADC – a sort of standard Genetic Algorithm where selection is performed with a technique called *deterministic crowding* (DC). This technique has been designed in order to try to avoid the loss of genetic diversity appearing when one uses the classical roulette selection method. In DC, the following steps are typically done:

a) two individuals are selected randomly
b) by crossover, two new individuals are generated
c) applied to the set of two parent and two descendent individuals to group them in two pairs, maximizing the similarity inside each pair; for instance, in chromosomes coding the individuals in bits, one may use Hamming distance (no. of different bits) to classify similarity
d) an elitist selection is applied for each group of similar individuals, finally selecting two individuals to form the next generation
e) this process is repeated until the following generation has the desired number of individuals.

3.2 GAMP

Another way to preserve diversity and allow good space exploration is a technique where multiple populations (MP) are kept isolated, and only from time to time selected individuals are allowed to migrate from one sub-population to another and inject their genetic material there. This technique also avoids the inconvenient phenomenon of genetic drift present in the simple Genetic Algorithm, where evolution may get stuck because all individuals prematurely become alike and no more space exploration is possible.

In the work reported, we have used only two sub-populations; at each step in time we have exchanged two individuals, randomly selected, from one of the sub-populations to the other, before crossover is applied.

3.3 EPSO

EPSO can be seen as a hybrid method of Evolution Strategies (ES) [6] and Particle Swarm Optimization (PSO) techniques [7]. As an Evolution Strategy, an EPSO algorithm may be described as follows. At a given iteration, consider a set of solutions or alternatives that we will now call particles (in the PSO tradition). The general scheme of EPSO is the following:

<u>REPLICATION</u> - each particle is replicated r times

<u>MUTATION</u> - each particle has its strategic parameters mutated

<u>REPRODUCTION</u> - each mutated particle generates an offspring through recombination, according to the particle movement rule, described below

<u>EVALUATION</u> - each offspring has its fitness evaluated

<u>SELECTION</u> - by stochastic tournament or other selection procedure, the best particles survive to form a new generation

3.3.1 Recombination and movement rule

The reproduction rule for EPSO is the following: given a particle $\mathbf{X}_i$, a new particle $\mathbf{X}_i^{new}$ results from

$$\mathbf{X}_i^{(k+1)} = \mathbf{X}_i^{(k)} + \mathbf{V}_i^{(k+1)}$$

$$\mathbf{V}_i^{(k+1)} = w_{i1}^* \mathbf{V}_i^{(k)} + w_{i2}^* (\mathbf{b}_i - \mathbf{X}_i) + w_{i3}^* (\mathbf{b}_g^* - \mathbf{X}_i)$$

where

$\mathbf{b}_i$ – best point found by particle i in its past life up to the current generation

$\mathbf{b}_g$ – best overall point found by the swarm of particles in their past life up to the current generation

$\mathbf{X}_i^{(k)}$ – location of particle i at generation k

$\mathbf{V}_i^{(k)} = \mathbf{X}_i^{(k)} - \mathbf{X}_i^{(k-1)}$ – is the velocity of particle i at generation k

w_{i1} – weight conditioning the *inertia* term (the particle tends to move in the same direction as the previous movement)

w_{i2} – weight conditioning the *memory* term (the particle is attracted to its previous best position)

w_{i3} – weight conditioning the *cooperation* or *information exchange* term (the particle is attracted to the overall best-so-far found by the swarm).

The symbol * indicates that these parameters will undergo evolution under a mutation process to be explained. This is called "the movement rule" in PSO; it is illustrated in **Figure 1**. In fact, it is a form of a recombination operator called intermediary recombination, where the value of any variable in the offspring receives a contribution from all parents. What is special in EPSO is the choice of parents: the global best, the best particle ancestor and the direct parent. And what is unusual in EPSO is that, in this method, the recombination operator is adaptive and evolving, instead of being fixed.

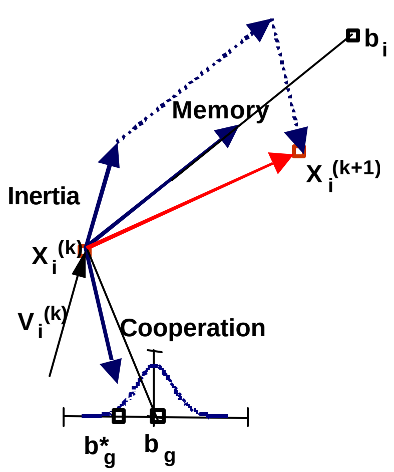

Figure 1 – Illustration of EPSO particle reproduction: a particle Xi generates an offspring at a location commanded by the movement rule.

3.3.2 Mutating strategic parameters

As in a σSA-Evolution Strategy, we distinguish, in each particle or solution representation, object parameters and strategic parameters.

Object parameters are those giving the *phenotypic* description of a solution (its natural variables). Strategic parameters are those that condition the evolution of a given solution.

The basic mutation rule for the strategic parameters is the following:

$$w_{ik}^* = w_{ik} \left[\log N(0,1)\right]^{\tau}$$

where

logN(0,1) is a random variable with lognormal distribution derived from the Gaussian distribution N(0,1) of 0 mean and variance 1;

τ is a learning parameter, fixed externally, controlling the amplitude of the mutations – smaller values of τ lead to higher probability of having values close to 1.

The logN distribution is classically adopted for strategic parameters because, in this multiplicative form of mutation, the probability of having a new value multiplied by m is the same as having a value multiplied by 1/m.

Approximations to this scheme are sometimes used by other researchers, such as

$$w_{ik}^* = w_{ik} \left[1 + \tau N(0,1)\right]$$

and they are equivalent provided that τ is small and the outcome is controlled so that negative weights are ruled out. This scheme is preferable to additive mutations like

$$w_{ik}^* = w_{ik} + \tau N(0,1)$$

because in this case the absolute value of the mutation is insensitive to the value of w.

As for the global best $\mathbf{b}_g$, it is randomly disturbed to give

$$\mathbf{b}_g^* = \mathbf{b}_g + w_{i4}^* N(0,1)$$

where w_{i4} is the forth strategic parameter associated with particle i. It controls the "size" of the neighborhood of $\mathbf{b}_g$ where it is more likely to find the real global best solution (assumed not found so far during the process) or, at least, a solution better that the current $\mathbf{b}_g$. This weight w_{i4} is mutated (signaled by *) according to the general mutation rule of strategic parameters, allowing the search to focus on a given point, if convenient.

3.3.3 The merits of EPSO

There is a solid theoretical background giving insight on why ES achieve convergence and how a near optimal progress rate is achieved [6]. In any ES, the generation of offspring is regulated by operations of mutation and crossover. However, these reproduction mechanisms do not provide a positive push towards the optimum – this is the role of the operator selection.

On the other hand, in Classical PSO there is a reproduction scheme but selection is trivial – each parent has one child and each of these survives to its parent, in a sort of parallel nx(1,1)ES. However, the movement (reproduction) rule, by itself, assures the progress to the optimum, meaning that, on average, each generation will be better that the preceding one.

In EPSO we have two mechanisms acting in sequence, each one with its own probability of producing not only better individuals but an average better group. Selection acts on a generation that is already on average better that the preceding, so the effects are additive.

The fact that EPSO is self-adaptive adds another interest to the method: it avoids in a large scale the need for fine tuning the parameters of the algorithm, because the procedure will hopefully learn (in the evolutionary sense) the characteristics of the search space and will self-tune the weights in order to produce an adequate rate of progress towards the optimum.

4 SIMULATION CONDITIONS

A Gas Retailer Agent has been equipped, in turns, either with a GADC, GAMP or EPSO, while the remaining retailers were kept acting with a pre-determined fixed strategy. A strategy optimization via internal simulation, performed by the Gas Retailer Agent, used individuals (particles) with the following decision variables:

- energy price for residential consumers
- energy price for commercial consumers
- energy price for industrial consumers
- incentive to residential consumers
- incentive to commercial consumers
- incentive to industrial consumer
- advertisement cost
- service cost
- quality improvement investment
- investment on management efficiency improvement

All these variables are represented as real numbers.

In all cases, the population has been set to 20 individuals. An initial population has been randomly initialized. In the case of GAMP, we have split the population into two sub-populations of equal size of 10.

The market simulation has been run for a period of 720 days (24 months), every day. The Gas Retailer activates its internal simulation at the beginning of every month, to optimize strategic decisions.

The objective is to maximize the accumulated profit evaluated in a period of n days and the objective function (or fitness function), the same for all algorithms, is described by

$$\text{Maximize } OBJ = \sum_{d=1}^{n} \left[Eco_d - Pen_d \right] \quad (1)$$

subject to:

$$Price_a^{min} < Price_a < Price_a^{max}$$
$$Dev\,Price_a < Limit$$
$$Incentive_a^{min} < Incentive_a < Incentive_a^{max}$$
$$Advertise_a^{min} < Advertise_a < Advertise_a^{max}$$
$$Service_a^{min} < Service_a < Service_a^{max}$$
$$Quality_a^{min} < Quality_a < Quality_a^{max}$$
$$Management_a^{min} < Management_a < Management_a^{max}$$

where

n = number of days of an internal simulation

Eco_d = Economical performance value at day d

Pen_d = Penalty assigned at day d

$Price_d$ = Energy prices applied to residential, commercial and industrial consumers

$Dev\,Price_a$ = Deviation in energy prices

$Incentive_d$ = Incentives given to residential, commercial and industrial consumers

$Advertise_d$ = Spending on advertisement

$Service_d$ = Spending on customer service improvement

$Quality_d$ = Spending on quality improvement of the product

$Management_d$ = Spending on management efficiency improvement

These latter variables are mentioned because the model causes consumers to react differently to each of them. This aspect of the implementation is to be improved in later versions.

The objective function is actually a combination of two parts: economical performance and penalties assigned for breaking rules. The economic performance part of a retailer can is evaluated using

$$Eco_d = P_d \times UP_d + S_d \times US_d \quad (2)$$

where,

UP_d = Unit profit of retail at day d

US_d = Unit market share at day d

P_d = Profit weight factor at day d

S_d = Share weight factor at day d

The economic performance of the retailer is judged by two components, its profit received and market share holding. The energy retailer is designed to take different action in different situation. When the action foreseen is of the aggressive type in which attention is only fixed on profiting, economical performance is evaluated as

$$Eco_d = \begin{cases} P_d^1 \times UP_d + S_d^0 \times US_d & \text{if } \delta P_d \geq 0 \\ P_d^2 \times UP_d + S_d^0 \times US_d & \text{if } 0 > \delta P_d \geq \delta P_{limit}^{lower} \\ P_d^3 \times UP_d + S_d^0 \times US_d & \text{if } \delta P_d < \delta P_{limit}^{lower} \end{cases} \quad (3)$$

subject to:

$$P_d^0 < P_d^1 < P_d^2 < P_d^3$$

where δP_d represents the derivative of current profit with respect to reference profit. The value of the profit weight factor changes with respect to the condition of δP_d. The predetermined value δP^{lower}_{limit} is set as the lower limit for the profit derivative.

When action is defensive, with attention more on market share gaining, the economical performance is evaluated as

$$Eco_d = \left\{ \begin{array}{c} P_d^0 \times UP_d + S_d^1 \times US_d \quad \text{if } \delta S_d \geq 0 \\ P_d^0 \times UP_d + S_d^2 \times US_d \quad \text{if } 0 > \delta S_d \geq \delta S^{lower}_{limit} \\ P_d^0 \times UP_d + S_d^3 \times US_d \quad \text{if } \delta S_d < \delta S^{lower}_{limit} \end{array} \right\} \qquad (4)$$

subject to:

$$S_d^0 < S_d^1 < S_d^2 < S_d^3$$

where δS_d is the derivative of the current profit relative to reference profit. The predetermined value δS^{lower}_{limit} is also set as the lower limit for share derivative.

Penalties are evaluated as follows, when an aggressive move is foreseen:

$$Pen_d = C + \delta S^2 \times F \qquad (5)$$

and it are evaluated as follows when action prepared is defensive

$$Pen_d = C + \delta P^2 \times F \qquad (6)$$

where,

C = constant

F = penalty factor

In our experiments, we have fixed n = 60 days for all internal simulations. In all cases, the stopping criterion has been the same: in the first month, when performing the first internal simulation, the evolutionary process would be stopped if after 50 consecutive generations there were no improvement in the fitness function; in all the following internal simulations, during the market simulation of 24 months, we have used the threshold of 10 generations instead of 50.

In GADC and in GAMP we have used uniform crossover and Gaussian mutations. In EPSO, we have used Lognormal mutations as described. In order to compensate for the influence of random events, we have run every simulation 5 times.

5 RESULTS

Figure 2 shows the evolution of profits during the market simulation, when a Retailer Agent is equipped with one of the three models GADC, GAMP or EPSO, in an average of 5 runs. Figure 3 represents the same but for the best run.

We see that EPSO shows much better values than GADC or GAMP. This suggests that, during the 24 months simulated, the strategic decisions produced by the EPSO internal simulator outperformed the strategic decisions produced by the competing algorithms.

The graphic in Figure 4 presents the present worth value for the 2 year period of market simulation. Figure 5 indicates the number of evaluations of the fitness function performed by each algorithm; they depend on the activation of the stopping criterion. We see that EPSO used much more computing effort, but as a compensation produced the best results.

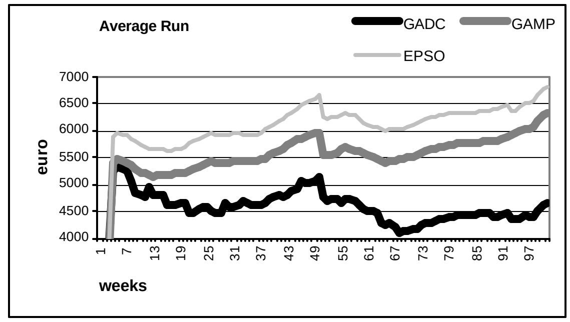

Figure 2 – Profits per week gained by each algorithm during 24 month simulation – average of 5 simulations

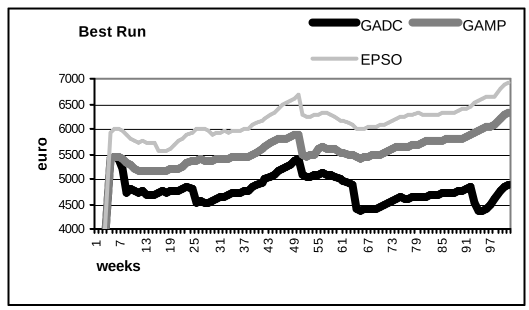

Figure 3 – Profits per week gained by each algorithm during 24 month simulation – best run

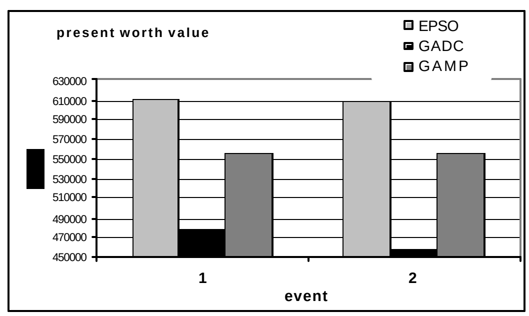

Figure 4 – Present worth value accumulated in 24 months of simulation. 1 – best run; 2 – average of 5 runs

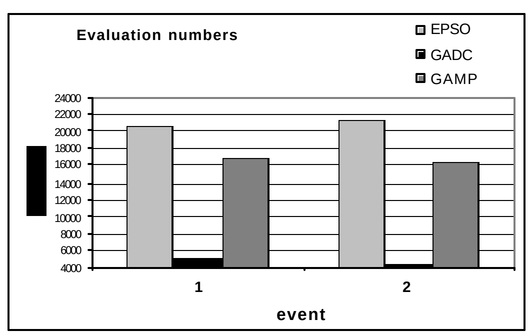

Figure 5 – Number of evaluations by each algorithm. 1 – best run; 2 – average of 5 runs

This suggests that EPSO may be a more flexible algorithm and does not get stuck so easily in local optima. The superior results obtained may be also a consequence of this characteristic. One could ask if, by increasing the size of the population in the GA approaches, these algorithms could have shown better performance. For instance, GADC shows poor results, when compared with the other options, but it is also the algorithm that employs less effort, measured in terms of number of fitness function evaluations. We have confirmed that an enlarged population in GADC and GAMP improves the results, but not significantly and not changing the general conclusion of the superiority of EPSO in flexibility: EPSO seems to offer a stronger guarantee of not getting trapped in local optima.

6 CONCLUSIONS

The research reported in this paper seems to clearly suggest that an algorithm belonging to the family of Evolutionary Particle Swarm Optimization outperforms evolutionary algorithms of the family of Genetic Algorithms in defining successful strategies for Retail Agents in an energy market simulation.

This simulation has been conducted under an Intelligent Agent paradigm, using a FIPA compliant opens source platform.

The market simulation has almost unpredictable outcomes due to the complex behavior of Agents, which emerges from their interaction. This is an extremely challenging problem and constitutes a hard test for algorithms trying to optimize the action of market agents.

Therefore, the results must be considered as meaningful and give evidence that the use, as a recombination operator in evolutionary models, of the particle movement rule borrowed from PSO methods, adds efficiency to the optimization procedures. This effect is increased in EPSO by its self-adapting characteristic.

In fact, EPSO may be seen as the first evolutionary algorithm with self-adaptive characteristics of the recombination operator. This feature proved remarkably well in the complex problem of interaction of Intelligent Agents.

REFERENCES

[1] FIPA – Foundation for Intelligent Physical Agents – http://www.fipa.org/

[2] JADE – Java Agent Development framework – http://jade.tilab.com/

[3] L. Tesfatsion, “Agent-based Computational Economics: Growing Economies from the Bottom Up”, Artificial Life, 8:55--82, 2002

[4] G. B. Sheblé, Agent Based Economics, Power Systems Restructuring: Engineering and Economics, Kluwer, 1998.

[5] V. Miranda, N. Fonseca, "EPSO – Best-of-Two-Worlds Meta-Heuristic Applied To Power System Problems ", Proceedings of WCCI 2002 - CEC - World Congress on Computational Intelligence - Conference on Evolutionary Computing, Honolulu, Hawaii, U.S.A., May, 2002

[6] H.-G. Beyer, “Toward a Theory of Evolution Strategies: Self-Adaptation”, in Evolutionary Computation, vol. 3, no. 3, pag. 311-347, 1996

R. Eberhardt and J. Kennedy, “A new optimizer using Particle Swarm theory”, Proceedings of the 6th International Symposium on Micro Machines and Human Science, Nagoya, Japan, 1995