Hybrid Swarm and Agent-Based Evolutionary Optimization

Leszek Placzkiewicz, Marcin Sendera, Adam Szlachta, Mateusz Paciorek,
Aleksander Byrski(✉), Marek Kisiel-Dorohinicki, and Mateusz Godzik

Department of Computer Science, Faculty of Computer Science,
Electronics and Telecommunications, AGH University of Science and Technology,
Al. Mickiewicza 30, 30-059 Krakow, Poland
leszekplaczkiewicz@gmail.com, marcin.sendera@gmail.com,
adam.szlachta@gmail.com, godzik.mateusz@gmail.com,
{mpaciorek,olekb,doroh}@agh.edu.pl

Abstract. In this paper a novel hybridization of agent-based evolutionary system (EMAS, a metaheuristic putting together agency and evolutionary paradigms) is presented. This method assumes utilization of particle swarm optimization (PSO) for upgrading certain agents used in the EMAS population, based on agent-related condition. This may be perceived as a method similar to local-search already used in EMAS (and many memetic algorithms). The obtained and presented in the end of the paper results show the applicability of this hybrid based on a selection of a number of 500 dimensional benchmark functions, when compared to non-hybrid, classic EMAS version.

1 Introduction

Solving difficult search problems requires turning to unconventional methods. Metaheuristics are often called "methods of last resort" and are successfully applied to solving different problems that cannot be solved with deterministic means in a reasonable time. Moreover, metaheuristics do not assume any knowledge about the intrinsic features of the search space, that helps a lot in solving complex problems such as combinatorial ones. It has also been proven that there is always need for searching for novel metaheuristics, as there is no Holy Grail of metaheuristics computing, and there is no one method that could solve all the possible problems with the same accuracy (cf. Wolpert and MacReady [21]). One has however to retain common sense and not produce the metaheuristics only for the sake of using another inspiration (cf. Sorensen [18]).

In 1996, Krzysztof Cetnarowicz proposed the concept of an Evolutionary Multi-Agent System (EMAS) [7]. The basis of this agent-based metaheuristic are agents—entities that bear appearances of intelligence and are able to make decisions autonomously. Following the idea of population decomposition and evolution decentralization, the main problem is decomposed into sub-tasks, each of which is entrusted to an agent. One of the most-important features of EMAS is

Y. Shi et al. (Eds.): ICCS 2018, LNCS 10861, pp. 89–102, 2018.
https://doi.org/10.1007/978-3-319-93701-4_7

the lack of global control—agents co-evolve independently of any superior management. Another remarkable advantage of EMAS over classic population-based algorithms is the parallel ontogenesis—agents may die, reproduce, or act at the same time. EMAS was successfully applied to solving many discrete and continuous problems, and was thoroughly theoretically analyzed, along with preparing of formal model proving its potential applicability to any possible problem (capability of being an universal optimizer, based on Markov-chain analysis and ergodicity feature) [3].

Particle swarm optimization [11] is an iterative algorithm commonly used for mathematical optimization of certain problems. Particle swarm optimization was originally proposed for simulating social behavior, and was used for simulating the group movement of fish schools, bird flocks, and so on. But the algorithm was also found to be useful for performing mathematical optimization after some simplification. The algorithm considers a number of particles moving in the search space, utilizing the available knowledge (generated by a certain particle and its neighbors) regarding the current optimal solutions, providing the user with an attractive technique retaining both exploitation and exploration features.

Memetic algorithms originate from Richard Dawkins' theory of memes. Meme is understood as a "unit of culture" that carries ideas, behaviors, and styles. This unit spreads among people by being passed from person to person within a culture by speech, writing, and other means of direct and indirect communication. The actual implementation of memetic algorithms proposed by Pablo Moscato is based on coupling local-search technique with evolutionary process, either on the reproduction level (e.g. during mutation: lamarckian memetization) or on the evaluation level (baldwinian memetization).

The hybrid method presented in this paper is based on coupling two metaheuristics, namely EMAS and PSO, using the memetic approach, i.e. allowing the agents in EMAS to run PSO-based "local-search". It should be noted, that PSO is a global optimization technique, and thus its synergy with EMAS seems to be even more attractive than e.g. introducing of a certain steepest-descent method that we have already done in the past [13].

The paper is organized as follows. After this introduction a number of hybrid PSO and evolutionary methods are referenced, leading the reader to the short recalling of EMAS basics and later presenting the PSO and its hybridization with EMAS. Next the experimental results comparing the base model of EMAS with the PSO-memetic one are shown, and finally the paper is concluded with some remarks.

2 Hybrid Particle Swarm Optimization

There exist many methods which can be used to hybridize Genetic Algorithms (GA) with Particle Swarm Optimization (PSO). One of them, called GA-PSO, has been presented by Kao and Zahara [10]. Their algorithm starts with generating a population of individuals of a fixed size $4N$ where N is a dimension of the solution space. The fitness function is calculated for each individual, the population is sorted by the fitness value and divided into two $2N$ subpopulations. The top $2N$ individuals are further processed using standard real-coded GA operators: crossover and mutation. Crossover is defined as a random linear combination of two vectors and happens with 100% probability. The probability of mutation is fixed at 20%. The obtained subpopulation of $2N$ individuals is used to adjust the remaining $2N$ individuals in PSO method. This operation involves the selection of the global best particle, the neighborhood and the velocity updates. The result is sorted in order to perform the next iteration. The algorithm stops when the convergence criterion is met, that is when a standard deviation of the objective function for $N + 1$ best individuals is below a predefined threshold (authors suggest 10^{-4}). The article shows the performance of the hybrid GA-PSO algorithm using a suit of 17 standard test functions and compares it to the results obtained with different methods (tabu search, simulated annealing, pure GA, and some modifications). In some cases GA-PSO performs clearly better, but in general behaves very competitive.

Similar method has been used by Li et al. [19]. Their algorithm, called PGHA (PSO GA Hybrid Algorithm), divides the initial population into two parts which then perform GA and PSO operators respectively. The subpopulations are recombined into new population which is again divided into two parts for the next population. Authors successfully used this technique for creation optimal antenna design.

Another method of hybridization of PSO and GA has been presented by Gupta and Yadav in [9] as PSO-GA hybrid. In their algorithm there are two populations, PSO and GA based, running independently and simultaneously. Occasionally, after a predefined number of iterations $N1$, certain number $P1$ of individuals from each system are designated for an exchange. The results authors obtained showed clear superiority of their PSA-GA hybrid technique over plain PSO and GA algorithms. The article compares GA, PSO and PSO-GA hybrid in the application of optimization 2nd and 3rd Order Digital Differential Operators.

There also exist GA/PSO hybrids for combinatorial problems. Borna and Khezri developed a new method to solve Traveling Salesman Problem (TSP) called MPSO [2]. Their idea is to perform the PSO procedure, but without using velocity variable. Instead, the crossover operator between *pbest* (particle's best position) and *gbest* (global best position) is used to calculate new positions. Both *pbest* and *gbest* values are updated as in normal PSO algorithm. Authors show that their MPSO technique gives better accuracy than other methods.

A combination of GA and PSO for combinatorial vehicle routing optimization problem (VRP) has been presented by Xu et al. [22]. Their algorithm starts with parameters and population initialization. Then the step of particle encoding is performed in order to calculate fitness function of each particle for VRP problem in the following step. Then *pbest* and *gbest* values are updated as in standard PSO. After that particles positions and velocities are recalculated using special crossover formulas which use a random value from a defined range to describe crossover probability. If the fitness of offspring is lower than the fitness of parents it is discarded, otherwise it replaces the parents. The algorithm is performed in loop until the stop conditions are met. The test results show that the proposed algorithm can find the same solutions as the best-known, but has overall better performance than other algorithms.

AUC-GAPSO is a hybrid algorithm proposed by Ykhlef and Alqifari in order to solve winner determination problem in multiunit double internet auction [23]. In each iteration the chromosomes are updated using specialized for this problem crossover and mutation operators. After that a PSO step is performed and new *gbest* and *pbest* together with new positions and velocities are calculated. If *gbest* is not being changed for more than one fourth of the maximum number of generations, the algorithm stops as no further improvement is assumed. Authors showed that their method performs superior to plain AUC-GA giving higher performance and reduced time to obtain satisfactory optimization results.

Different variation of PSO-GA hybrid has been presented by Singh et al. [17]. Their technique, called HGPSTA, is similar to ordinary GA. PSO is used to enhance individuals before performing crossover and mutation operators. Once the fitness values of all individuals are calculated, the most successful first half is selected for further processing using crossover. Parents are selected by roulette wheel method. Mutation is then performed on entire population. HGPSTA (Hybrid Genetic Particle Swarm Technique Algorithm) has been used to identify the paths of software that are error prone in order to generate software test cases. Authors demonstrated that the method needs less iterations to deliver 100% test coverage than plain GA and PSO.

The performance of GA is also improved by incorporating PSO in the work of Nazir et al. [16]. Individuals are enhanced by PSO step after crossover and mutation operations are performed. There are some innovations to the basic algorithm. The first one is that the probability of taking PSO enhancement into account varies according to a special formula. The second one is that if *gbest* value remains a number of times unchanged it is updated to prevent from getting trapped in local extremum. The method has been used to select the most significant features in gender classification using facial and clothing information.

Another hybrid method has been presented by Abd-El-Wahed, Mousa and El-Shorbagy [1], who apply it to solve constrained nonlinear optimization problems. The entire procedure is based on interleaving steps of PSO and GA mechanisms. Moreover the algorithm incorporates a calculation and usage of modified dynamic constriction factor to maintain the feasibility of a particle. In GA part selection, crossover and mutation are used, as well as elitist strategy. The last step of an iteration is to repair infeasible individuals to make them feasible again. Authors show an excellent performance of the algorithm for a presented set of test problems.

Algorithm presented by Mousavi et al. in [15] is a mixture of PSO and GA steps. The PSO part is performed first (updating particles' positions and velocities), then standard selection, crossover and mutation steps follow. Before and after the GA part the boundary check is done for each particle. If a particle is out of predefined boundary then a new random particle is generated until it fits into the boundary. Authors successfully applied their GA-PSO method in multi-objective AGV (automated guided vehicles) scheduling in a FMS (flexible manufacturing system) problem. The study shows that GA-PSO outperforms single PSO and GA algorithms in this application.

Kuo and Han in [14] describe and evaluate three hybrid GA and PSO algorithms, HGAPSO-1, HGAPSO-2, HGAPSO-3. The first two are taken from other studies, whereas the last one is invented by the authors. This method follows the general PSO procedure, but if *gbest* is unchanged in given iteration, then each particle is additionally updated using mutation operator. The idea is to prevent premature convergence to a local optimum. Moreover the elitist policy is applied in the last step. Positions of particles are checked to fit into a defined range, also a velocity value is constrained by a predefined upper limit. Authors show that their version is superior to the other two described. They apply the method to solving bi-level linear programming problem.

Another overview of PSO hybridizations is presented in [20] by Thangaraj, Pant, Abraham and Bouvry. The research also include some other algorithms used in conjunction with PSO like differential evolution, evolutionary programming, ant colony optimization, sequential quadratic programming, tabu search, gradient descend, simulated annealing, k-means, simplex and others. A small subset of them is chosen for further performance comparison using a set of standard numerical problems like Rosenbrock function, DeJong function etc.

Summing up the presented state-of-the-art, one can clearly see that many approaches using Genetic Algorithm with PSO for improvement of the solutions were realized, however none of them considered hybridization in fully autonomous environment. Thus we would like to present an agent-based metaheuristic that utilizes PSO selectively, by certain agent, and its decision is fully autonomous.

3 Evolutionary Multi Agent-Systems

Evolutionary Multi Agent-System [7] can be treated as an interesting and quite efficient metaheuristic, moreover with a proper formal background proving its correctness [3]. Therefore this system has been chosen as a tool for solving the problem described in this paper.

Evolutionary processes are by nature decentralized and therefore they may be easily introduced in a multi-agent system at a population level. It means that agents are able to *reproduce* (generate new agents), which is a kind of cooperative interaction, and may *die* (be eliminated from the system), which is the result of competition (selection). A similar idea with limited autonomy of agents located in fixed positions on some lattice (like in a cellular model of parallel evolutionary algorithms) was developed by Zhong et al. [24]. The key idea of the decentralized model of evolution in EMAS [12] was to ensure full autonomy of agents.

Such a system consists of a relatively large number of rather simple (reactive), homogeneous agents, which have or work out solutions to the same problem (a common goal). Due to computational simplicity and the ability to form independent subsystems (sub-populations), these systems may be efficiently realized in distributed, large-scale environments (see, e.g. [4]).

Agents in EMAS represent solutions to a given optimization problem. They are located on islands representing distributed structure of computation. The islands constitute local environments, where direct interactions among agents may take place. In addition, agents are able to change their location, which makes it possible to exchange information and resources all over the system [12].

In EMAS, phenomena of inheritance and selection—the main components of evolutionary processes—are modeled via agent actions of *death* and *reproduction* (see Fig. 1). As in the case of classical evolutionary algorithms, inheritance is accomplished by an appropriate definition of reproduction. Core properties of the agent are encoded in its genotype and inherited from its parent(s) with the use of variation operators (mutation and recombination). Moreover, an agent may possess some knowledge acquired during its life, which is not inherited. Both inherited and acquired information (phenotype) determines the behavior of an agent. It is noteworthy that it is easy to add mechanisms of diversity enhancement, such as allotropic speciation (cf. [6]) to EMAS. It consists in introducing population decomposition and a new action of the agent based on moving from one evolutionary island to another (migration) (see Fig. 1).

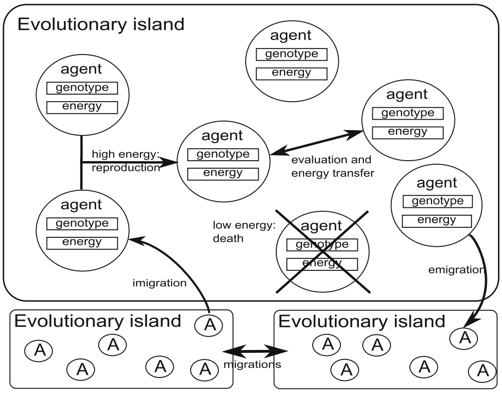


Fig. 1. Evolutionary multi-agent system (EMAS)

Assuming that no global knowledge is available, and the agents being autonomous, selection mechanism based on acquiring and exchanging non-renewable resources [7] is introduced. It means that a decisive factor of the agent's fitness is still the quality of solution it represents, but expressed by the amount of non-renewable resource it possesses. In general, the agent gains resources as a reward for "good" behavior, and looses resources as a consequence of "bad" behavior (behavior here may be understood as, e.g. acquiring sufficiently good solution). Selection is then realized in such a way that agents with a lot of resources are more likely to reproduce, while a low level of resources increases the possibility of death. So according to classical Franklin's and Graesser's taxonomy—agents of EMAS can be classified as Artificial Life Agents (a kind of Computational Agents) [8].

Many optimization tasks, which have already been solved with EMAS and its modifications, have yielded better results than certain classical approaches. They include, among others, optimization of neural network architecture, multi-objective optimization, multimodal optimization and financial optimization. EMAS has thus been proved to be a versatile optimization mechanism in practical situations. A summary of EMAS-related review has is given in [5].

EMAS may be held up as an example of a cultural algorithm, where evolution is performed at the level of relations among agents, and cultural knowledge is acquired from the energy-related information. This knowledge makes it possible to state which agent is better and which is worse, justifying the decision about reproduction. Therefore, the energy-related knowledge serves as situational knowledge. Memetic variants of EMAS may be easily introduced by modifying evaluation or variation operators (by adding an appropriate local-search method).

4 From Classic to Hybrid PSO

In the basic particle swarm optimization [11] implementation, the potential solutions are located in a subspace of D-dimensional Euclidean space R^D limited in each dimension (usually a D-dimensional hypercube). The search space is a domain of the optimized quality function $f : R^D \rightarrow R$.

A particle is a candidate solution described by three D-dimensional vectors: position $X = x_d, d \in [1 \ldots D]$; velocity $V = v_d, d \in [1 \ldots D]$; best known position $P = p_d, d \in [1 \ldots D]$. A swarm is a set of m particles. The swarm is associated with a D-dimensional vector $G = g_d, d \in [1 \ldots D]$ which is swarm's best known position (the solution with the currently highest quality).

The execution of the algorithm begins by initializing the start values. Each particle I belonging to the swarm S is initialized with the following values:

1. position X of the particle I is initialized with a random vector belonging to the search space A
2. best known position is initialized with current particle's position: $P \leftarrow X$
3. velocity V of the particle I is initialized with a random vector belonging to the search space A
4. swarm's best position is updated by the following rule: $if\ f(P) < f(G)\ then\ G \leftarrow P$

Once all the particles are initialized and uniformly distributed in the search space, the main part of the algorithm starts executing. During each iteration of the algorithm, the following steps are executed. These steps of the algorithm are executed until a termination criteria are met. The most common termination criteria for the particle swarm optimization are:

Algorithm 1

```
for each particle I in swarm S do
    update particle's velocity:
        $V \leftarrow r_g(G - X) + r_p(P - X) + \omega V; r_g, r_p \in [0, 1]$
    update particle's position:
        $X \leftarrow X + V$
        where $\omega$ is the inertia factor
    update particle's best position:
        $if\ \ f(X) < f(P)\ \ then\ \ P \leftarrow X$
    update global best position:
        $if\ \ f(P) < f(G)\ \ then\ \ G \leftarrow P$
end for
```

1. number of executed iterations reaches a specified value,
2. swarm's best position exceeds a specified value,
3. the algorithm found global optimum,
4. swarm's best positions in two subsequent iterations are the same.

The idea of hybridization of EMAS with PSO follows the cultural and memetic inspirations, by utilizing the PSO-defined movements of the solutions (agents' genotypes) as a kind of additional "local-search" algorithm for making the "worse" agents better by updating their solutions (see Fig. 2). This is not entirely a local-search algorithm, as PSO of course is a well-known global optimization technique, however the planned synergy seems to be attractive and thus not prone to early-convergence problems.

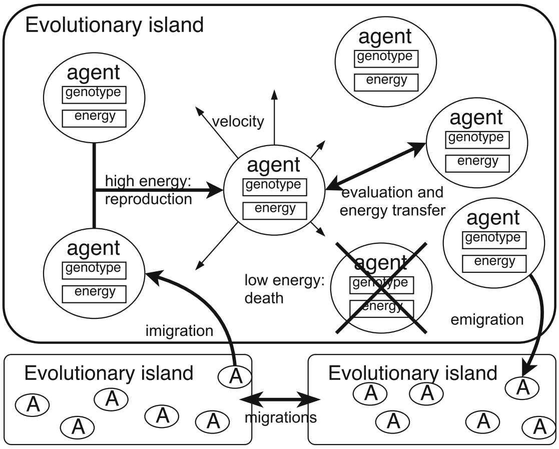


Fig. 2. Evolutionary multi-agent system with PSO modification (PSO-EMAS)

In the proposed hybrid algorithm, the agent may be treated either as regular EMAS agent—when its energy is higher than certain, fixed level, and as

PSO particle—when its energy is lower (a dedicated energy threshold, so called “move” energy is considered a parameter of the algorithm). Thus better agents are evolved using well-known evolutionary methods, while worse agents update their solutions based on PSO rules.

5 Experimental Results

The experiments were performed taking advantage of AgE 3 platform[1], which is distributed, agent-based computational platform developed by Intelligent Information Systems Group. The platform was further developed in order to combine PSO with EMAS. The tests were executed on Samsung NP550P5C with Intel CORE i5-3210M @ 2.5 GHz; 8 GB RAM; Ubuntu 14.04.5 LTS.

5.1 Experimental Setting

In the PSO aspect of the hybrid algorithm, an agent can move in the search space only when its energy value is lower than 40. The max/min velocity parameters determine the size of the move performed by an agent. Other parameters presented below relate to the formula below, which is used for updating agent’s velocity.

$$v_{i,d}^{t+1} \leftarrow \omega \cdot v_{i,d}^{t} + r_p(p_{i,d} - x_{i,d}) + r_g(g_d - x_{i,d})$$

where:

- $v_{i,d}^{t}$ describes i-th agent’s (particle) d-th component of its velocity in t-th step of algorithm;
- r_p and r_g are random numbers within (0, 1) range;
- $p_{i,d}$ is i-th agent’s local best position d-th component value;
- $x_{i,d}$ is i-th agent’s current position d-th component value;
- g_d is globally best position d-th component value;
- ω is a weight considering current velocity of particle.

The most important parameters set for the compared systems were as follows:

- EMAS parameters: Population size: 50; Initial energy: 100; Reproduction predicate: energy above 45; Death predicate: energy equal to 0; Crossover operator: discrete crossover; Mutation operator: uniform mutation; Mutation probability: 0.05; Reproduction energy transfer: proportional, 0.25; Fight energy transfer: 5.0;
- PSO parameters: Move energy threshold: 40; Maximum velocity: 0.05; ω 0.5.

For each dimensionality and algorithm variant (EMAS or PSO-EMAS hybrid) optimization tests were performed 30 times and the stopping condition was time-related, namely each experiment could last only for 200 s.

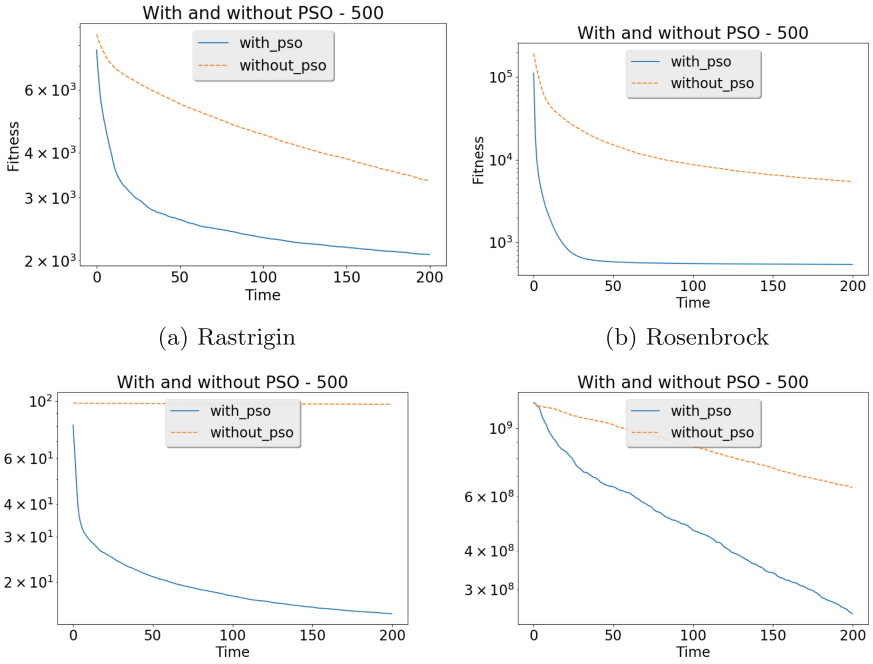
[1] http://www.age.agh.edu.pl.

5.2 Discussion of the Results

The main objective of the tests was to compare optimization results achieved for PSO-EMAS hybrid with those obtained for EMAS approach. The experiments were realized in the following sequence.

In the beginning, selected benchmark problems (Rastrigin in Fig. 3a, Rosenbrock in Fig. 3b, Schwefel in Fig. 3c and Whitley in Fig. 3d) were optimized in 500 dimensions, in order to realize preliminary checking of the compared algorithms. As shown in Fig. 3 in all the considered cases the hybrid of PSO and EMAS did significantly better, however it is to note, that in all the cases the actual global optima were not approached closely, probably because of arbitrarily chosen algorithm parameters. Thus, in order to do further examination, any of these problems could have been selected, therefore we have selected Rastrigin problem, as this is a very popular benchmark and we have already used it many times in our previous research.

Next, the parameters of the constructed hybrid (namely move energy, maximum velocity, weights of personal and global optima and weight of the previous vector in PSO update) were tested on 500 dimensional Rastrigin problem. The results of these tests are presented in Fig. 4.



(a) Rastrigin

(b) Rosenbrock

(c) Schwefel

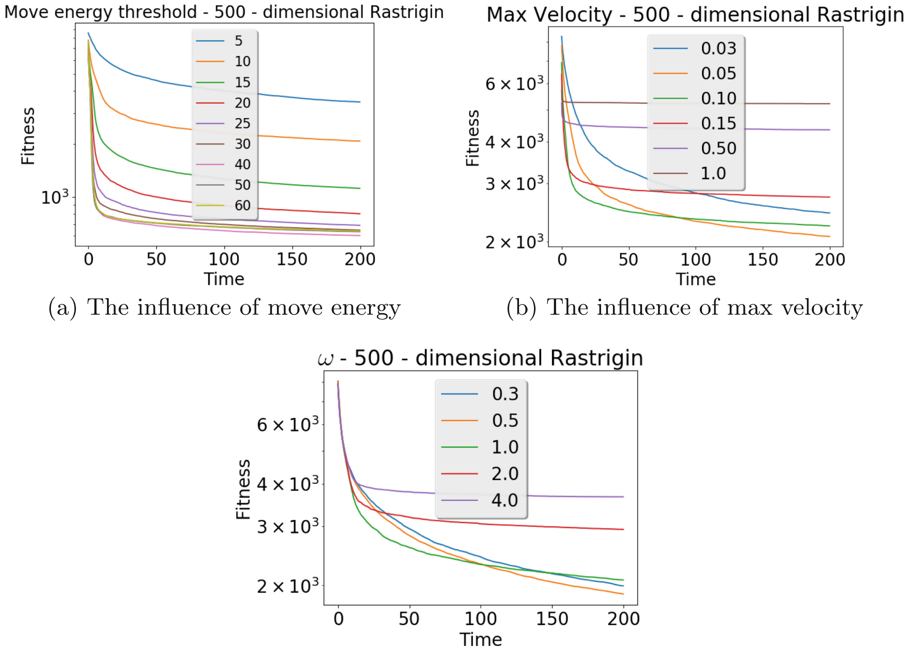
(d) Whitley

Fig. 3. Comparison of EMAS and PSO-EMAS fitness for selected 500 dimensional benchmark functions optimization

Testing the move energy (see Fig. 4a) it is easy to see that the best results were obtained for its value 40 (out of tested values between 5 and 60). It is to note, that the reproduction energy is 45, so the difference is quite small: the agents apparently participate in PSO hybrid until their energy becomes close to the reproduction threshold. Then the PSO action is suspended and the agents participate in EMAS part of the hybrid, acting towards reproduction.

Testing the maximum velocity (see Fig. 4b) can be summarized with a quite natural and predictable solution: from the values between 0.03 and 1.0 the value of 0.05 turned out to be the best in the tested case, suggesting that too high values of the velocity cap will bring the examined hybrid to a random stochastic search type algorithm, hampering the intelligent search usually realized by meta-heuristic algorithms.

The graph showing the dependency of the weight of the previous vector ω (see Fig. 4c) yielded 0.5 as the optimal value of this parameter for the tested case. Again, similar to the observation of the move energy, not too big value (considering the tested range) turned out to be the best. It is quite predictable, as almost “copying” the previous vector would stop the exploration process, while complete forgetting about this vector would lose the “metaheuristic” information turning the whole algorithm to a purely random walk technique.



(a) The influence of move energy

(b) The influence of max velocity

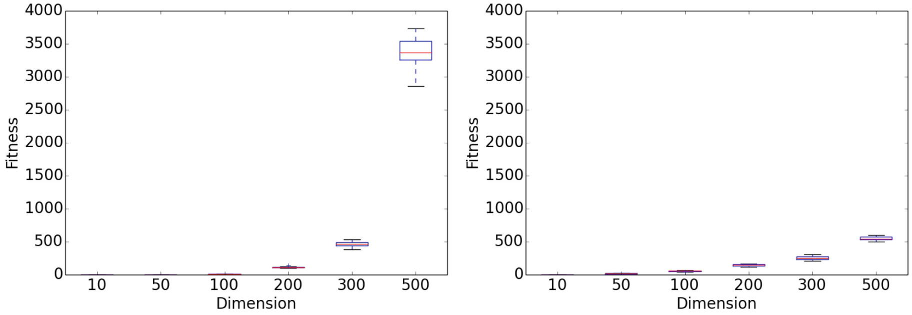
(c) The influence of ω value

Fig. 4. Optimization of 500-dimensional Rastrigin problem using various values of PSO parameters

Finally, the Rastrigin problem was tested in different dimensions (10, 50, 100, 200, 300, 500), using the best values of the hybrid parameters found in the previous step. For Rastrigin problem in less than 200-dimensional domains standard EMAS achieved better results than hybrid variant, as shown on Fig. 5 and in Table 1. However in higher dimensional problems PSO-EMAS hybrid significantly outperforms standard algorithm yielding both better fitness values and lower standard deviations. The latter highlights good reproducibility of conducted experiments, as opposed to results of EMAS in 500-dimensional Rastrigin experiments.

Table 1. Final results found by EMAS and PSO-EMAS with standard deviation for optimization of Rastrigin function in different dimensions

Dimensions	EMAS average	EMAS std. dev.	PSO-EMAS average	PSO-EMAS std. dev.
10	0.00	0.00	0.00	0.00
50	0.00	0.00	12.15	8.78
100	1.40	0.40	52.26	6.62
200	108.81	9.60	143.45	13.14
300	464.16	35.80	251.19	27.51
500	3343.55	216.58	546.88	28.50



(a) EMAS

(b) PSO-EMAS

Fig. 5. Comparison of final fitness values for EMAS and PSO-EMAS using the best parameters found during the experimentation.

6 Conclusion

In the paper a PSO and EMAS hybrid was presented and tested against several selected, popular benchmark functions. The research consisted in preliminary testing different benchmark functions using arbitrarily chosen parameters, then a detailed study on the best values for the PSO parameters based on Rastrigin function in 500 dimensions was realized, and finally the efficacy of EMAS and PSO-EMAS was tested for the Rastrigin function in different dimensions, using the above-mentioned parameter values.

The results show that the hybrid version is significantly better than the original one in some of the considered cases. Moreover, not only final fitness values were similar or better (obtained in the assumed time of 200 s) but also in most of the tested cases better fitness was significantly earlier obtained by the hybrid version of the algorithm. In the future we plan to propose new PSO and EMAS hybrid algorithms, as well as do broader experimentation with the presented PSO-EMAS metaheuristic.

Acknowlegment. The research presented in this paper was partially supported by the Grant of the Dean of Faculty of Computer Science, Electronics and Telecommunications, AGH University of Science and Technology, for Ph.D. Students.

References

1. Abd-El-Wahed, W.F., Mousa, A.A., El-Shorbagy, M.A.: Integrating particle swarm optimization with genetic algorithms for solving nonlinear optimization problems. J. Comput. Appl. Math. **235**(5), 1446–1453 (2011)
2. Borna, K., Khezri, R.: A combination of genetic algorithm and particle swarm optimization method for solving traveling salesman problem. Cogent Math. **2**(1) (2015)
3. Byrski, A., Schaefer, R., Smołka, M., Cotta, C.: Asymptotic guarantee of success for multi-agent memetic systems. Bull. Pol. Acad. Sci.-Tech. Sci. **61**(1), 257–278 (2013)
4. Byrski, A., Debski, R., Kisiel-Dorohinicki, M.: Agent-based computing in an augmented cloud environment. Comput. Syst. Sci. Eng. **27**(1), 7–18 (2012)
5. Byrski, A., Dreżewski, R., Siwik, L., Kisiel-Dorohinicki, M.: Evolutionary multi-agent systems. Knowl. Eng. Rev. **30**(2), 171–186 (2015)
6. Cantú-Paz, E.: A summary of research on parallel genetic algorithms. IlliGAL Report No. 95007. University of Illinois (1995)
7. Cetnarowicz, K., Kisiel-Dorohinicki, M., Nawarecki, E.: The application of evolution process in multi-agent world (MAW) to the prediction system. In: Tokoro, M. (ed.) Proceedings of the 2nd International Conference on Multi-Agent Systems (ICMAS 1996), pp. 26–32. AAAI Press (1996)
8. Franklin, S., Graesser, A.: Is it an agent, or just a program?: a taxonomy for autonomous agents. In: Müller, J.P., Wooldridge, M.J., Jennings, N.R. (eds.) ATAL 1996. LNCS, vol. 1193, pp. 21–35. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0013570
9. Gupta, M., Yadav, R.: New improved fractional order differentiator models based on optimized digital differentiators. Sci. World J. **2014**, Article ID 741395 (2014)
10. Kao, Y.-T., Zahara, E.: A hybrid genetic algorithm and particle swarm optimization for multimodal functions. Appl. Soft Comput. **8**(2), 849–857 (2008)
11. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of International Conference on Neural Networks, vol. 4, pp. 1942–1948, November 1995

12. Kisiel-Dorohinicki, M.: Agent-oriented model of simulated evolution. In: Grosky, W.I., Plášil, F. (eds.) SOFSEM 2002. LNCS, vol. 2540, pp. 253–261. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36137-5_19
13. Korczynski, W., Byrski, A., Kisiel-Dorohinicki, M.: Buffered local search for efficient memetic agent-based continuous optimization. J. Comput. Sci. **20**(Suppl. C), 112–117 (2017)
14. Kuo, R.J., Han, Y.S.: A hybrid of genetic algorithm and particle swarm optimization for solving bi-level linear programming problem - a case study on supply chain model. Appl. Math. Model. **35**(8), 3905–3917 (2011)
15. Mousavi, M., Yap, H.J., Musa, S.N., Tahriri, F., Md Dawal, S.Z.: Multi-objective AGV scheduling in an FMS using a hybrid of genetic algorithm and particle swarm optimization. PLOS ONE **12**(3), 1–24 (2017)
16. Nazir, M., Majid-Mirza, A., Ali-Khan, S.: PSO-GA based optimized feature selection using facial and clothing information for gender classification. J. Appl. Res. Technol. **12**(1), 145–152 (2014)
17. Singh, A., Garg, N., Saini, T.: A hybrid approach of genetic algorithm and particle swarm technique to software test case generation. Int. J. Innov. Eng. Technol. **3**, 208–214 (2014)
18. Sörensen, K.: Metaheuristics—the metaphor exposed. Int. Trans. Oper. Res. **22**(1), 3–18 (2015)
19. Li, W.T., Xu, L., Shi, X.W.: A hybrid of genetic algorithm and particle swarm optimization for antenna design. In: Progress in Electromagnetics Research Symposium, vol. 2 (2008)
20. Thangaraj, R., Pant, M., Abraham, A., Bouvry, P.: Particle swarm optimization: hybridization perspectives and experimental illustrations. Appl. Math. Comput. **217**(12), 5208–5226 (2011)
21. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Trans. Evol. Comput. **67**(1), 67–82 (1997)
22. Xu, S.-H., Liu, J.-P., Zhang, F.-H., Wang, L., Sun, L.-J.: A combination of genetic algorithm and particle swarm optimization for vehicle routing problem with time windows. Sensors **15**(9), 21033–21053 (2015)
23. Ykhlef, M., Alqifari, R.: A new hybrid algorithm to solve winner determination problem in multiunit double internet auction. **2015**, 1–10 (2015)
24. Zhong, W., Liu, J., Xue, M., Jiao, L.: A multiagent genetic algorithm for global numerical optimization. IEEE Trans. Syst. Man Cybern. Part B: Cybern. **34**(2), 1128–1141 (2004)