sciendo

JAISCR, 2019, Vol. 9, No. 2, pp. 81 – 97
10.2478/jaiscr-2018-0026

FINE TUNING OF AGENT-BASED EVOLUTIONARY COMPUTING

Michal Mizera, Pawel Nowotarski, Aleksander Byrski*, Marek Kisiel-Dorohinicki

Department of Computer Science, Faculty of Computer Science, Electronics and Telecommunications, AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Krakow, Poland

**E-mail: olekb@agh.edu.pl*

Submitted: 18th January 2018; Accepted: 10th May 2018

Abstract

Evolutionary Multi-agent System introduced by late Krzysztof Cetnarowicz and developed further at the AGH University of Science and Technology became a reliable optimization system, both proven experimentally and theoretically. This paper follows a work of Byrski further testing and analyzing the efficacy of this metaheuristic based on popular, high-dimensional benchmark functions. The contents of this paper will be useful for anybody willing to apply this computing algorithm to continuous and not only optimization.

Keywords: multi-agent systems, metaheuristics, evolutionary computing

1 Introduction

Novel metaheuristics are always needed in order to tackle difficult problems that cannot be solved by the deterministic means (cf. [31]). At the same time these metaheuristics cannot be proposed only to explore a new metaphor, but their existence should be properly motivated and proven (cf. [28]). In this paper we try to delve into intrinsic features of an agent-based evolutionary metaheuristic, i.e. EMAS (Evolutionary Multi-Agent System), and explore the parameters of the search, looking for relations between their values and their efficacy.

EMAS has been proposed in 1996 by Krzysztof Cetnarowicz [16] and since then gathered many successful followers who worked on a significant number of applications (continuous, discrete optimization), single-criteria, multi-criteria, multimodal [13]. Moreover a number of software frameworks were created keeping in mind that they should be used i.a. to flexibly support EMAS computing, following formal deliberations on their feasibility [12, 24]. Following preliminary results presented in [6] (summarized by rather qualitative conclusion), we would like to provide a more tangible and measurable results trying to assess the quality of the EMAS-based search in the continuous optimization domain.

A detailed insight into parametrization of EMAS and immunological EMAS [10, 11] (e.g. efficiency, energy levels) for a selected benchmark function was given in [6]. This paper may be treated as its direct follow-up, showing the influence of different parameters of variation operators, different configurations of the agent population, different dimensions of the tackled problems and a broader selection of benchmark functions. This paper brings makes possible for a potential user of EMAS to gather more preliminary experience before setting and tuning properly the algorithm for a particular use.

The paper is organized as follows. In the beginning the EMAS basics are recalled, along with

motivation for constructing of such computing systems, then the extensive experimental results section begins, where EMAS applied to solving popular benchmark problems like Rosenbrock, Schwefel, Rastrigin etc. [17] is tested systematically using different configurations, looking for optimal ones for particular tested problems. Finally, the paper is concluded.

2 Evolutionary Multi-Agent System

Evolutionary Multi Agent-System [16] can be treated as an interesting and quite efficient metaheuristic, moreover with a proper formal background proving its correctness [5]. 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 evolutionary processes in a multi-agent system at a population level may be easily introduced. 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 a 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. [33]. The key idea of the decentralized model of evolution in EMAS [23] was to ensure the full autonomy of agents.

Such a system consists of a relatively large number of rather simple (reactive), homogeneous agents [32], 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. [8]). One has to refer here to the work of Hanna and Cagan [22] who also propose evolutionary multi-agent systems, however they work rather on the level of more technical processing by employing the agent-paradigm, contrary to the idea of integrating the agency and evolution in the EMAS described here.

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 [23].

In EMAS, phenomena of inheritance and selection – the main components of evolutionary processes [21, 2] – are modeled via agent actions of *death* and *reproduction* (see Figure 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. [14, 1]) 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 Figure 1).

Assuming that no global knowledge is available, and the agents being autonomous, selection mechanism based on acquiring and exchanging non-renewable resources [16] 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 a 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) [20].

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 [3], multi-

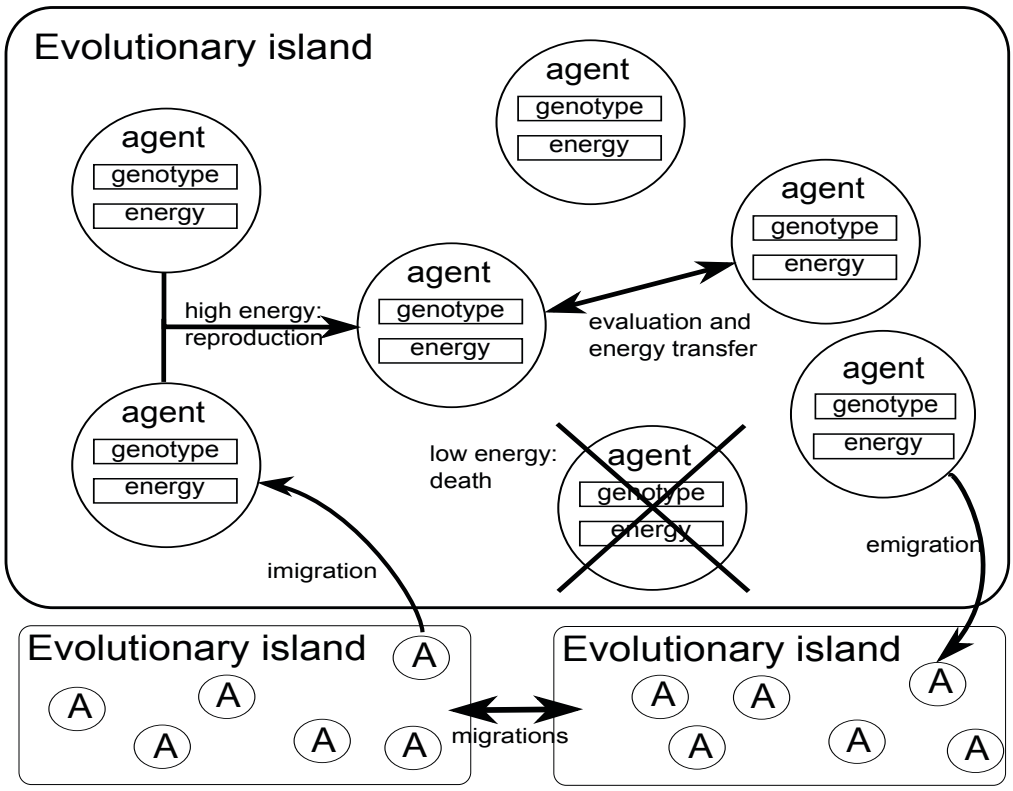


Figure 1. Evolutionary multi-agent system (EMAS)

objective optimization [27], multi-modal optimization [18] and financial optimization [19]. EMAS has thus been proved to be a versatile optimization mechanism in practical situations. A summary of EMAS-related review has is given in [9].

EMAS may be held up as an example of a cultural algorithms, 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).

3 Experimental results

In this Section, various experimental results connected with tuning the EMAS will be presented, mostly focusing on mutation, as this component of the computing system is mostly connected with finding a balance between exploration and exploitation. Other aspects will also be tackled, e.g. the number of the evolutionary islands. The presented experimental results are supposed to be a continuation and extension of the research presented in [6]. All these experiments were realized using Zeus supercomputer (HP BL2x220c, Intel Xeon, 23 TB, 169 TFlops) made available by ACC Cyfronet AGH.

The Base configuration of EMAS looks as follows:

- Migration Minimum Threshold = 120,
- Newborn energy = 100,
- Fight transferred energy = 40,
- Reproduction minimum threshold = 120,
- Initial energy = 100,
- Mutation probability = 0.01,
- Mutation radius = 0.1,
- Global mutation probability = 0,
- Torus size = 10x10,

– Number of islands = 3,

– Initial population size (per island) = 25.

3.1 Initial experiments

Aiming at extending the research presented in [6], we tried to test broader ranges of a selected configuration parameters. For example, in the beginning, in order to assess the capability of solving problems of different difficulty, the Rastrigin benchmark was used in different dimensions (see Figure 2), running the computation for different periods of time. Namely completing 1 hour and 3 hours were assumed as stopping conditions and the observed results were compared.

In Figure 2a the best fitness observed during 1 hour is shown and in Figure 2b, the same for 3 hours. It is quite obvious to see, that EMAS behaves predictably in this test, namely assuming longer computing time, the results become significantly closer to global optimum (note different scale used on Y-axis). Moreover, the tests for the instances of lower and equal to 200 finish with suboptimal results very close to the global optimum in the case of 3 hours.

3.2 Energy thresholds

Next, again in Figure 2, dependencies of best fitness on time for different reproduction thresholds are shown. Namely, graphs showing these dependencies for minimum reproduction equal to 20 (see Figure 2c), 100 (see Figure 2d) and 200 (see Figure 2e) are presented. Moreover, below each graph the number of individuals in the system is shown, in order to check the stability of the population (note that the number of individuals in EMAS is not constant but varies, caused by the reproduction and death actions performed by the agents).

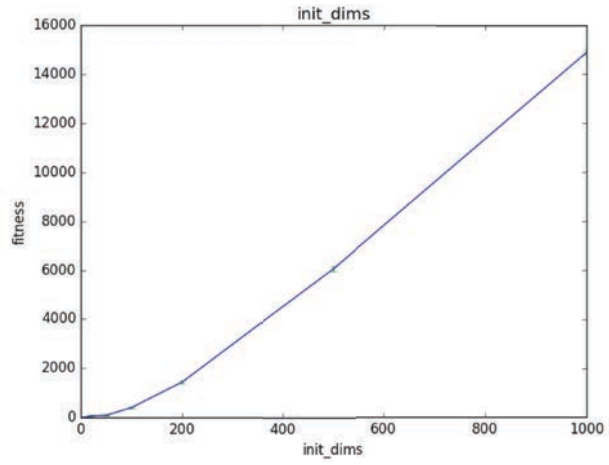
Regarding the fitness, it is easy to see that all the experiments yield a similar result (note these are only single runs in order to observe actual shapes of the fitness curves, so this conclusion cannot be drawn as a general one).

However, the most important observation based on these graphs is relevant to the dynamics of the population: in the first case (reproduction 20) the curve showing the number of agents in the population is smooth, while in the two next cases (100 and 200) the number of individuals fluctuates. Moreover, in the first case the number of individual rises, while in the two latter: falls down.

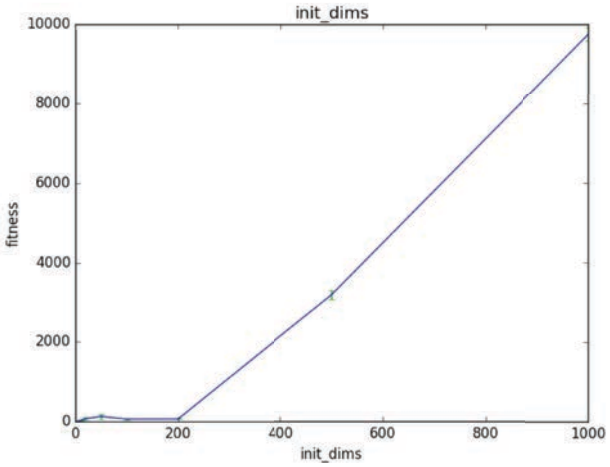
Of course it can be explained when looking at the actual values of the parameters: in all the cases the initial energy is 100, while in the first case the reproduction energy is lower than this threshold and most of the individuals start with reproduction. Later the average energy falls below the reproduction threshold, and the agents must gather energy in order to reproduce. However, not too many meetings are needed (note that the reproduction threshold is only 20, while the 40 units of energy is transferred when the agent wins a fight. As the transfer energy is close to the death energy, usually after winning the fight by one agent, the other dies: so the balance remains more or less stable, thus the curve is smoother than in the latter cases.

At the same time, more “predictable” situation arises for the two latter experiments, when the reproduction energy is significantly higher than the initial energy. In those cases, the number of agents falls quickly down first (as many of them die rather than reproduce) and later, when some of the agents gather enough energy, reproduction actions are realized. As there is significantly more energy to be gathered than in the first case, to reproduce, the fluctuation of the number of agents can be observed – in the population, different numbers of agents reproduce and die at the same time, thus showing clearly the parallel ontogenesis happening in the population.

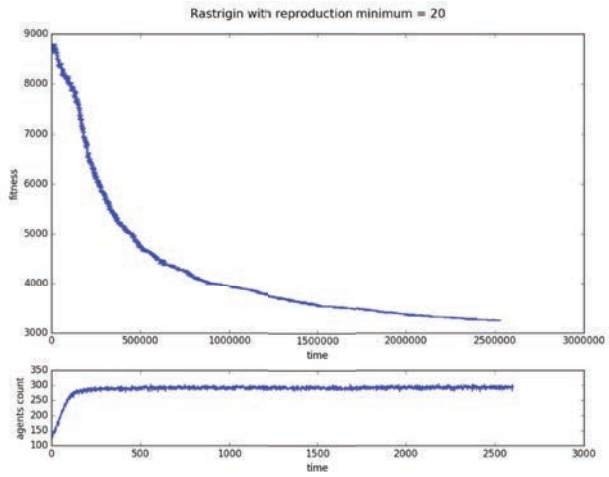
Finally, in Figure 2f the dependency of the final fitness on the reproduction, minimum parameter is presented. It seems that the optimal value of this parameter for the tested problem in the current setting is 100. For lower values of the reproduction thresholds, many more agents were present in the population, requiring much more computing power and finally bringing worse results. For the reproduction threshold equal to 100 the number of agents fluctuates around 30, proving the observations found in the literature (cf. [15]) that the subpopulation should not be very big, in order to do evolutionary computation in an efficacious and efficient way.



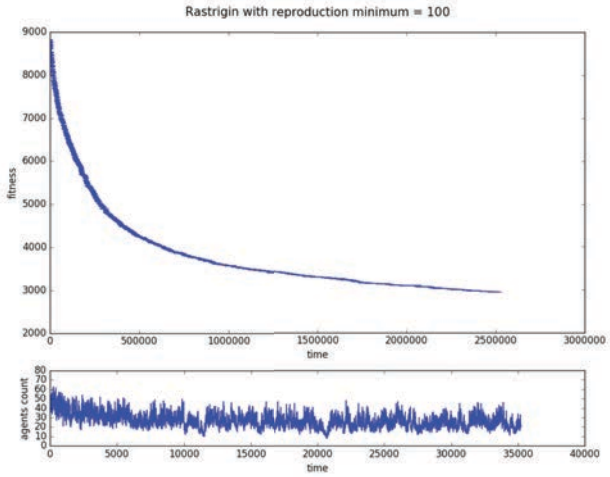
(a) 1 hour



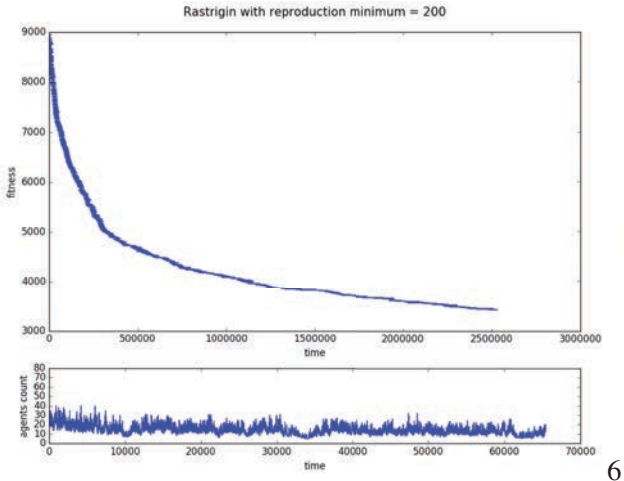
(b) 3 hours



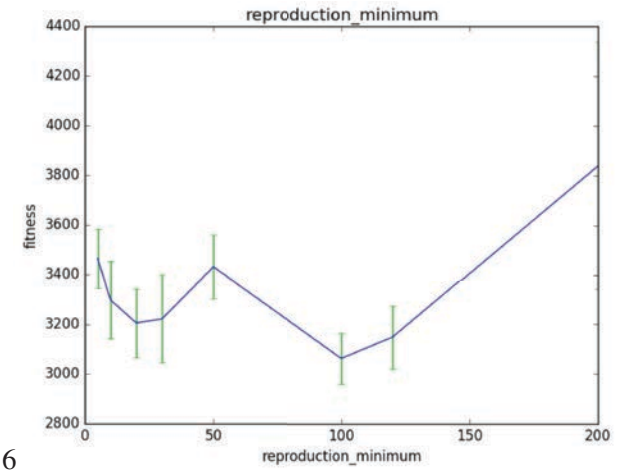
(c) Reproduction threshold 20



(d) Reproduction threshold 100



(e) Reproduction threshold 200



(f) Final fitness depending on the reproduction minimum

Figure 2. Final fitness after different time of computation, depending on the number of dimensions of the tackled Rastrigin problem. Single runs of computation with different reproduction thresholds, below each graph, the number of individuals in population is displayed. The final graph depicts the final fitness depending on the reproduction threshold.

3.3 Mutation parameters

For the mutation-related research the following configuration was used:

- Migration Minimum Threshold = 120,
- Newborn energy = 100,
- Fight transferred energy = 40,
- Reproduction minimum threshold = 90,
- Initial energy = 100,
- Mutation probability = [0.005, 0.01, 0.02, 0.03, 0.05, 0.1. 0.2, 0.3, 0.5. 0.7. 0.9],
- Mutation radius = [0.1, 0.20, 0.5, 0.75, 1.0, 1.5, 2.0, 2.5, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0] / [0.05, 0.07, 0.1, 0.15, 0.20, 0.25, 0.35, 0.5, 0.75, 1.0],
- Dimensionality of the problems = Schwefel - 20, Rosenbrock - 100, Rastrigin - 100, Griewank - 100,
- Torus size = 10x10,
- Number of islands = 3,
- Initial population size (per island) = 25,
- Number of iterations = 2501.

In Figure 3 mutation parameters were initially tested for the Rastrigin benchmark in order to find their optimal values for the considered problem. It is very clear, when looking at the two first graphs showing the dependency of the final fitness for the value of mutation probability (see Figures 3a and 3b) that the value of 0.01 seems to be optimal, and this is quite predictable as too small value of mutation probability will not introduce enough diversity into the genetic material.

Note that the ontogenesis in EMAS does not have such high rate as in classic evolutionary algorithms, there are no generations that tend to replace all the population with new individuals, therefore a proper setting of ontogenesis-related parameters will help in wise exploration and exploitation of the search space.

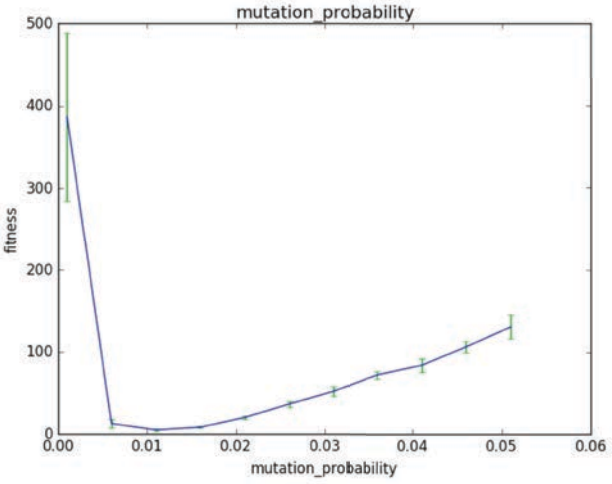
Initial research of mutation parameters gave significant improvement of results. We decided to analyze in details effects of changing probability and radius on results. For each problem series of experiments with different values were conducted.

Regarding the mutation radius (see Figures 3c and 3d, the optimal results are located in a small plateau between 0.7 and 1.4. The values below this range clearly hampered the exploration capability and the results were very bad. At the same time the values above this range were not so bad, however the quality of the obtained fitness slowly decreased, that is quite natural, as having too high mutation step would bring the search very quickly to oscillate between the sub-optimal results.

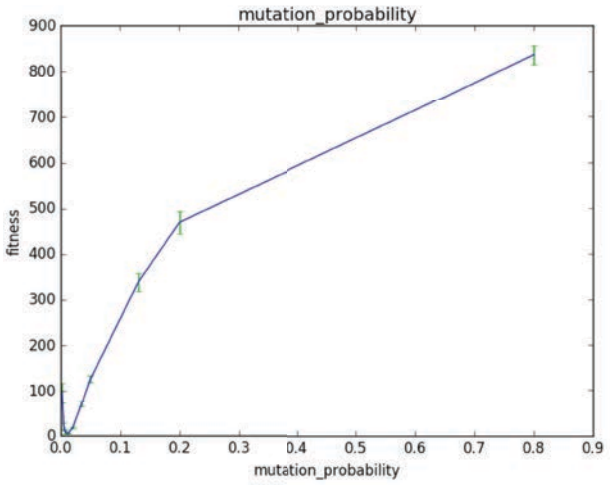
Finally, using the optimal mutation parameters found, we have tackled the Rastrigin problem in different dimensions (between 2 and 500), see Figures 3e, 3f. In the first graph, the best results are observed up to dimension equal to 80, however when the perspective changes, one can see a significant improvement comparing to the results presented in Figures 2a and 2b. Comparing the stopping condition, the results close to the global optimum were attained up to 200 dimensions after 1 or 3 hours in the previous experiments, and only after 14 mins. in the case of the experiments using optimal mutation parameters.

Of course, using the optimal mutation parameters found for Rastrigin problem in the case of solving other problems cannot bring the same or sometimes even similar efficacy results. We tried to apply the discussed parameters to optimization of Schwefel problem in different dimensions (see Figure 4a) and the result was more than discouraging, namely increasing the dimensionality of the problem the quality of results decreased dramatically.

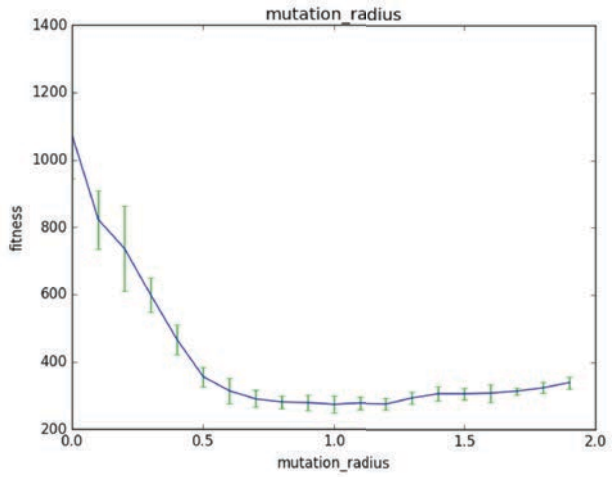
Thus, we decided to pursue the further examination of different efficacy-related effects observed when experimenting with several selected benchmark functions and mutation parameters. First the Griewank function in 200 dimensions was tested. In Figure 4b it is easy to see that starting from the value of mutation radius around 3, the final fitness obtained is reasonably close to the optimum. Further exploration of the mutation radius range would probably bring stabilization of the curve, but such experiments will be realized in the future. In this paper, we tended to retain the assumed symmetry of the results thus we did not cross the maximum value of the mutation radius: 10.



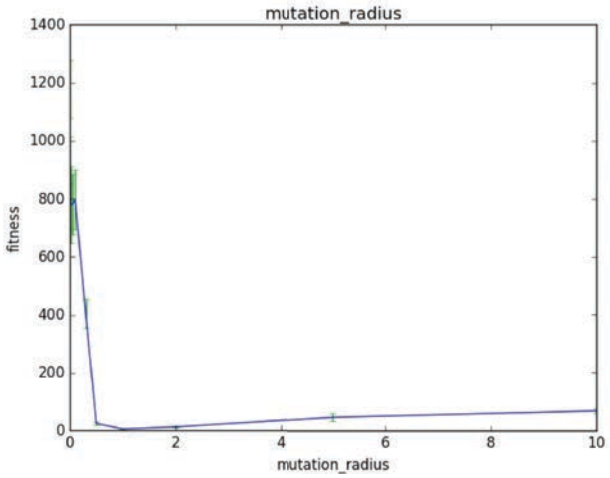
(a) Mutation probability (zoomed)



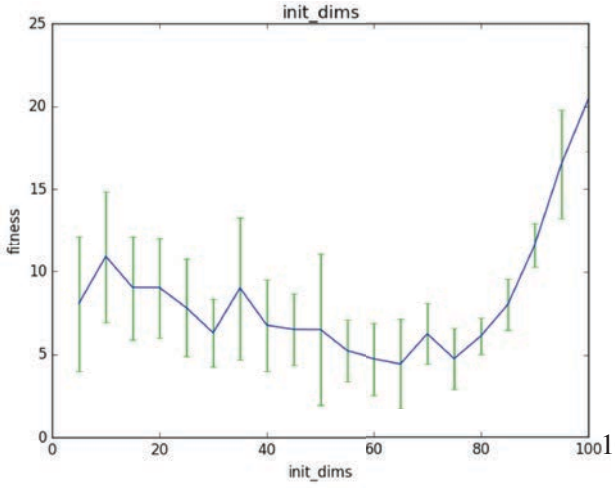
(b) Mutation probability



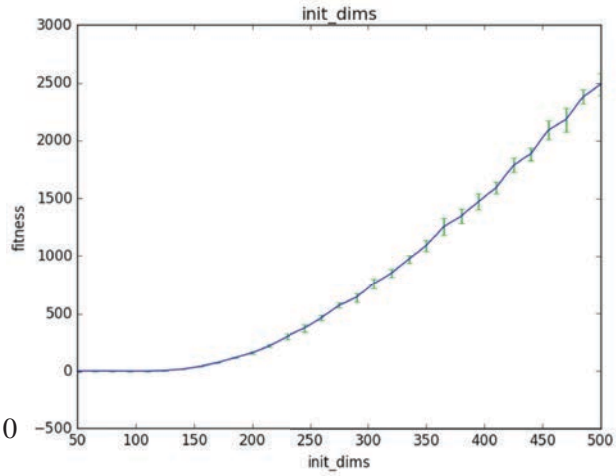
(c) Mutation radius (zoomed)



(d) Mutation radius

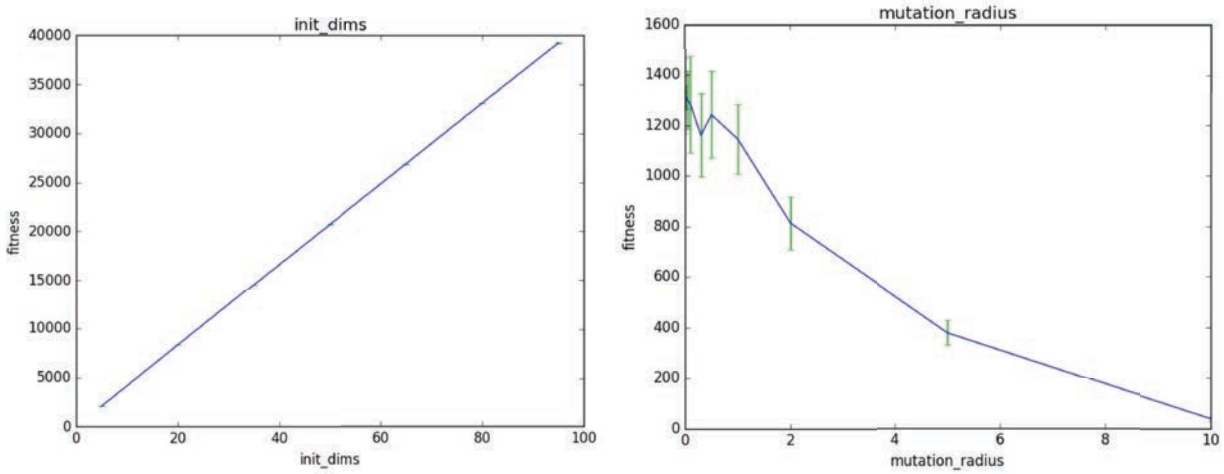


(e) Optimal mutation parameters depending on dimensions (zoomed)



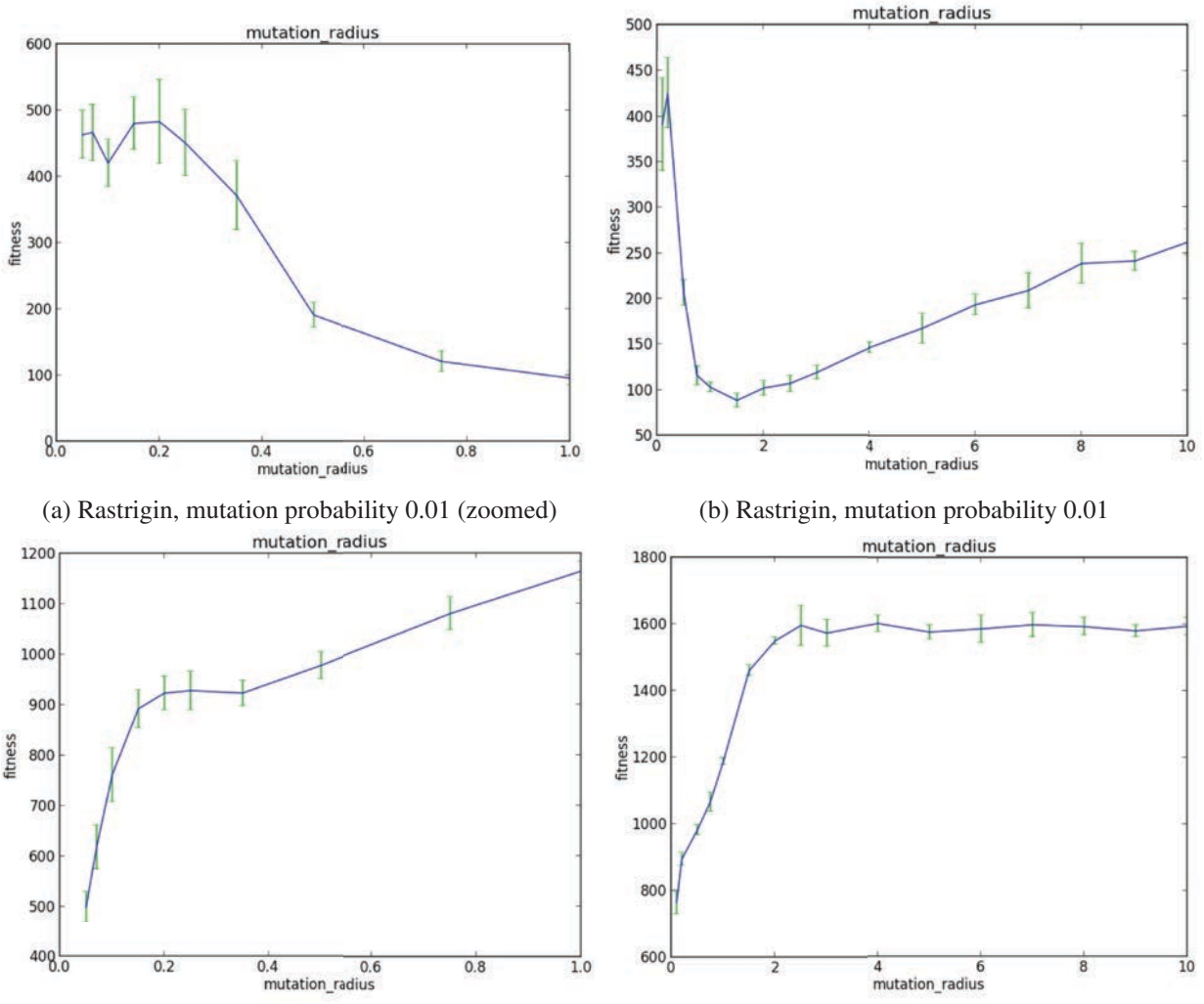
(f) Optimal mutation parameters depending on dimensions

Figure 3. Final fitness depending on mutation probability and mutation radius, finally optimal mutation parameters depending on dimensions for the tackled Rastrigin problem.



(a) Schwefel (b) Griewank

Figure 4. Final fitness for the tackled Schwefel and Griewank problems using optimal mutation parameters found for Rastrigin depending on the number of dimensions.



(a) Rastrigin, mutation probability 0.01 (zoomed) (b) Rastrigin, mutation probability 0.01

(c) Rastrigin, mutation probability 0.7 (zoomed) (d) Rastrigin, mutation probability 0.7

Figure 5. Final fitness of the tackled Rastrigin problem for different probabilities depending on the mutation radius.

Next, we have returned to the Rastrigin problem for a moment, in order to check the efficacy for different values of mutation radius, assuming the mutation probability to 0.01 and later to 0.7, in order to test two quite contrary values. The results are presented in Figure 5. One can easily see that the best efficacy was observed for the mutation probability equal to 0.01 and the mutation radius between 0.5 and 1.5 – in these cases the obtained results were relatively close to the optimum.

When testing higher mutation rate, namely 0.7, the obtained results for the smallest value of mutation radius are quite worse than the one obtained for mutation probability 0.01, and their quality decreases even more, when the mutation radius becomes higher.

In the case of optimization of Rosenbrock function, the obtained results had apparently lower efficacy – see Figure 6. The starting parameters turned out to be the best ones. Rosenbrock function belongs to especially difficult problems, therefore much more effort would be required here than only testing two mutation probabilities: 0.01 and 0.3.

Considering Schwefel problem while testing two different probabilities of mutation, namely 0.01 and 0.3 for different mutation radii (see Figure 7) no apparent success was encountered, the efficacy for all the observed cases stayed quite diverse and more or less stable.

Finally, considering Griewank problem while testing four different probabilities of mutation, namely 0.01, 0.1, 0.3 and 0.7 for different mutation radii (see Figure 8), it is easy to see that increasing the mutation probability in the assumed range increased the efficacy of the problem solution.

3.4 Population structure

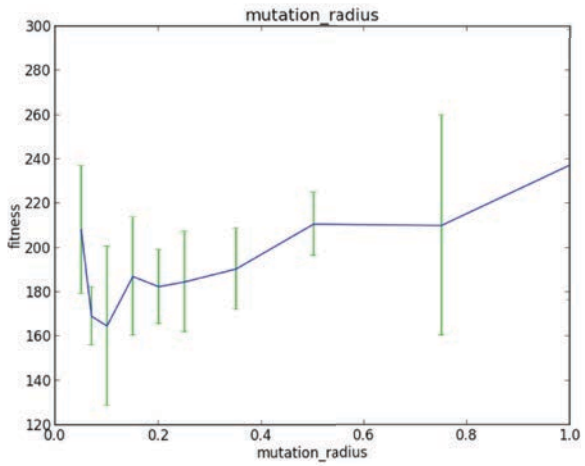
For the population structure examination the following configuration was used:

- Migration Minimum Threshold = 120,
- Newborn energy = 100,
- Fight transferred energy = 40,
- Reproduction minimum threshold = 90,
- Initial energy = 100,
- Mutation probability = 0.01,
- Mutation radius = 1,
- Dimensionality of the problems = Schwefel - 20, Rosenbrock - 100, Rastrigin - 75, Dejong - 250, Greiwang - 75,
- Torus size = 6x6, 8x8, 10x10,
- Number of islands = [1,2,3,5,7,10,15,20,25,30,50],
- Initial population size (per island) = 25,
- Number of iterations = 30000/(Number of islands) + 1.

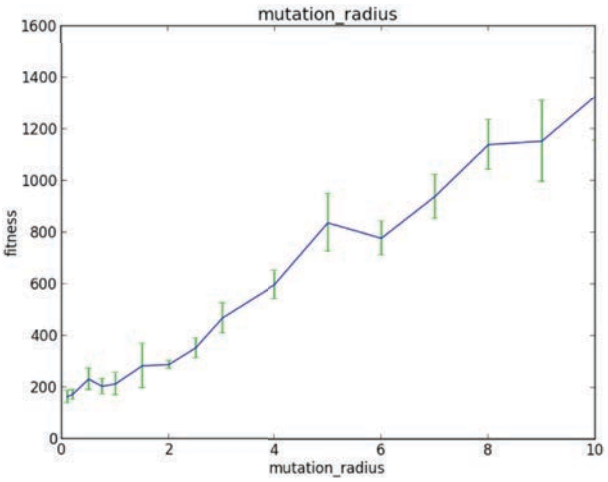
An important parameter affecting the efficiency and efficacy (and diversity of the search) of the evolutionary metaheuristic is the number of subpopulations. We have tackled four selected benchmark problems, namely Rastrigin, Schwefel, Rosenbrock and Griewank, changing the number of the evolutionary islands involved. In Figure 9 the final fitness obtained for these four problems run in different configurations concerning the number of islands are shown. It is quite predictable that increasing the number of islands makes possible more accurate localization of the suboptimal solutions with relation to the global optimum. Testing Rosenbrock, Rastrigin and Griewank problems the final fitness was of range close to 100-200. In the case of Schwefel function, the approaching to the global optimum was also observed, yet the final result was close to 5000, far from the global optimum.

The subpopulations present on each of evolutionary islands are located on tori of a certain size. In Figure 10 the final fitness depending on the number of evolutionary islands for different sizes of tori, considering Rastrigin and Schwefel problems. It is clear that the number of evolutionary islands again affects the final result to a large extent. At the same time the size of torus seems to be less significant, although apparently, the best results were obtained for the size 6 in the case of Schwefel function, and for the size 10 in the case of Rastrigin function.

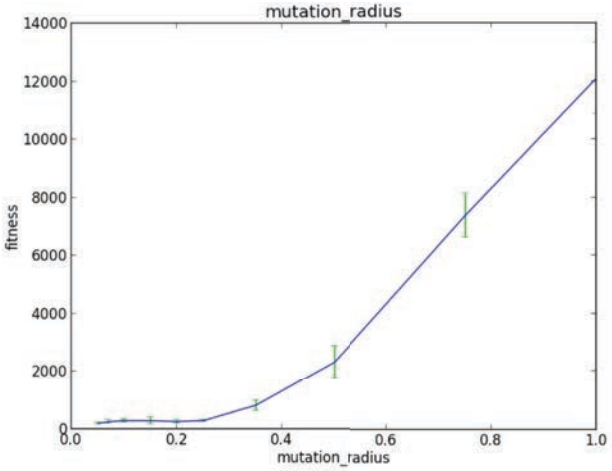
It is to note that while increasing the number of islands, the time of computation was decreased (as the iteration number was decreased), so the total sum of operations realized in the computing system was practically constant.



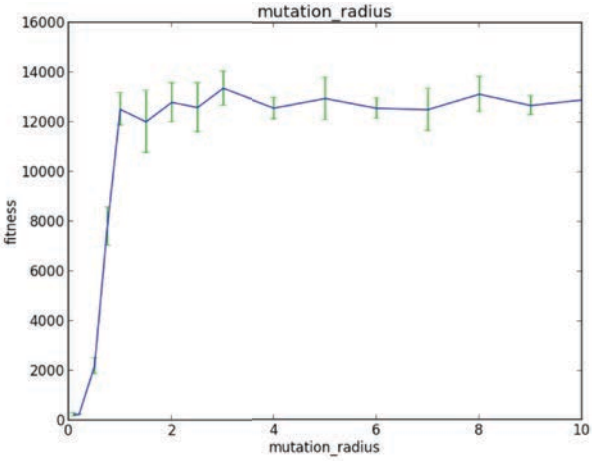
(a) Rosenbrock, mutation probability 0.01 (zoomed)



(b) Rosenbrock, mutation probability 0.01

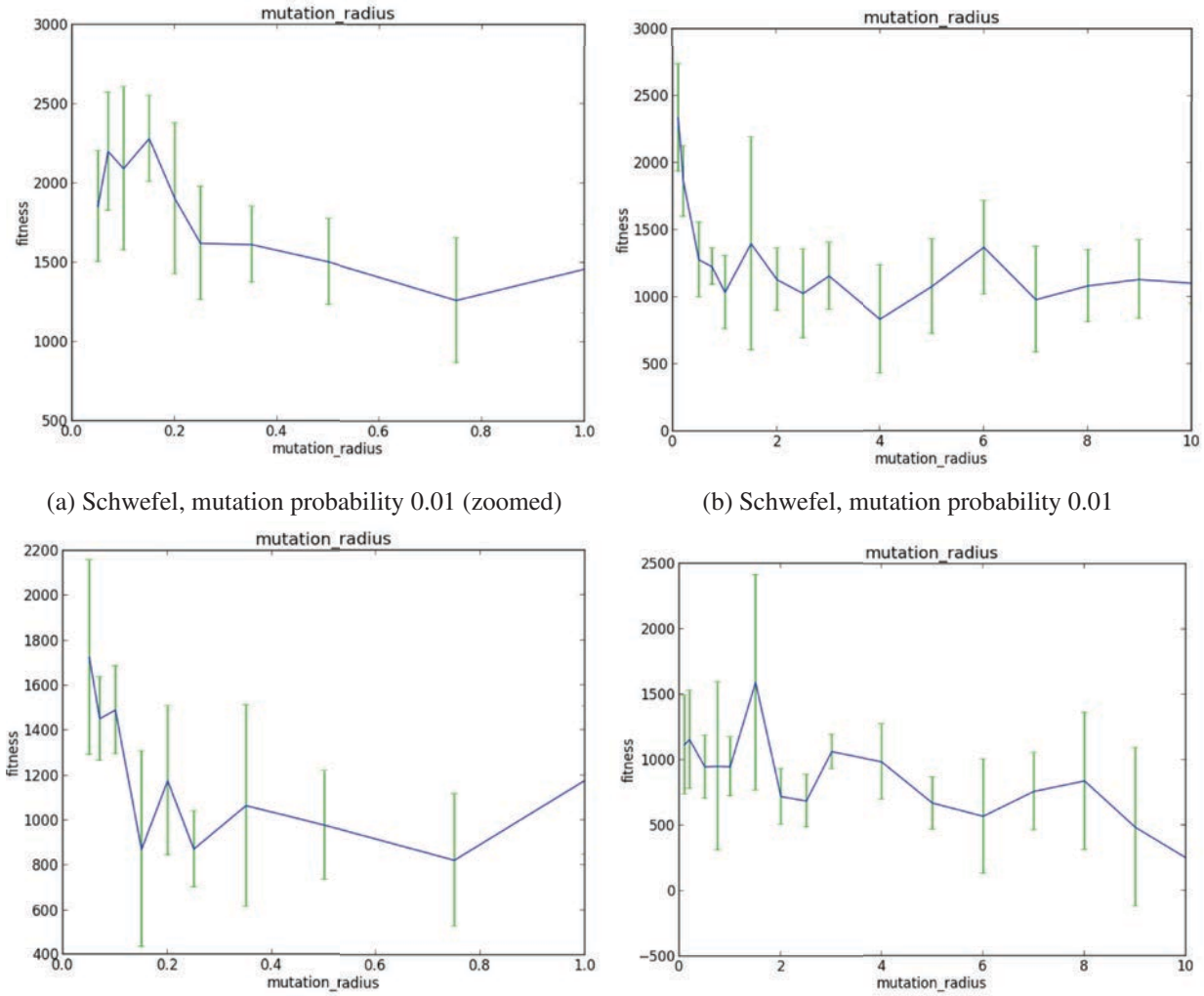


(c) Rosenbrock, mutation probability 0.3 (zoomed)



(d) Rosenbrock, mutation probability 0.3

Figure 6. Final fitness of the tackled Rosenbrock problem for different probabilities depending on the mutation radius.



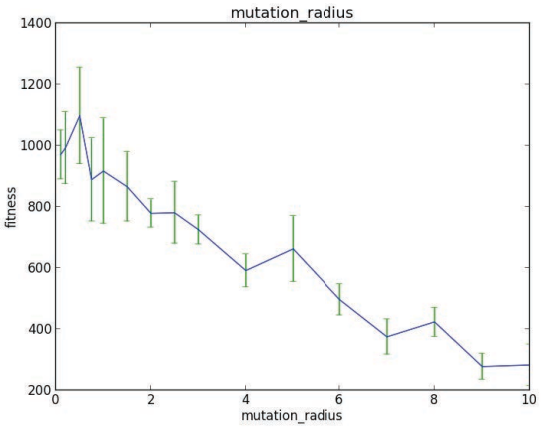
(a) Schwefel, mutation probability 0.01 (zoomed)

(b) Schwefel, mutation probability 0.01

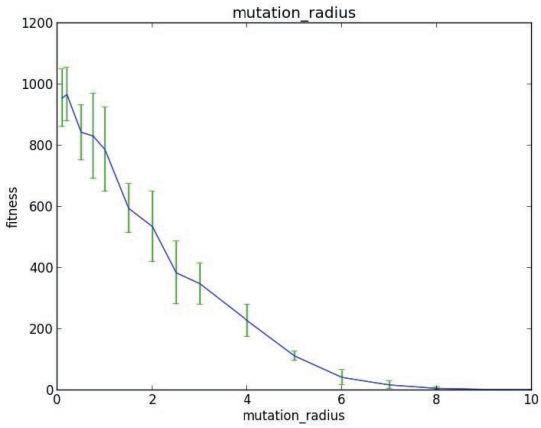
(c) Schwefel, mutation probability 0.3 (zoomed)

(d) Schwefel, mutation probability 0.3

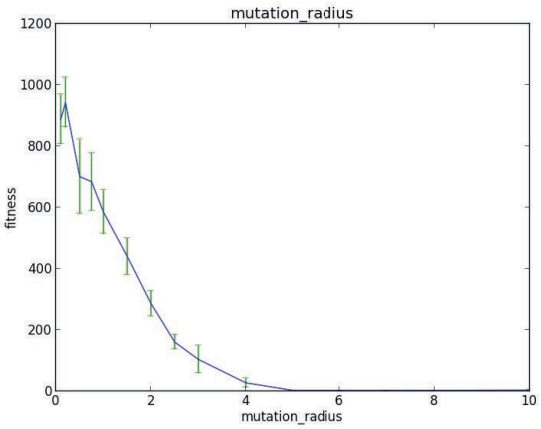
Figure 7. Final fitness of the tackled Schwefel problem for different probabilities depending on the mutation radius.



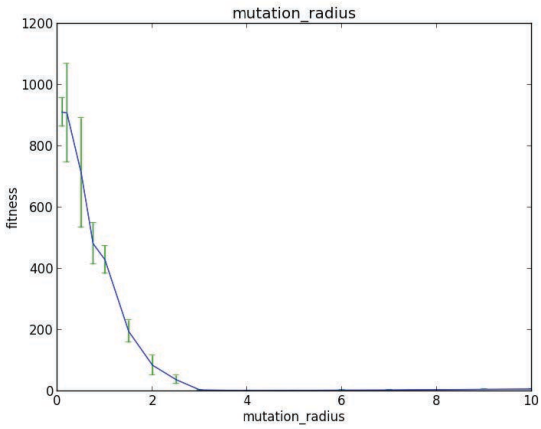
(a) Griewank, mutation probability 0.01



(b) Griewank, mutation probability 0.1

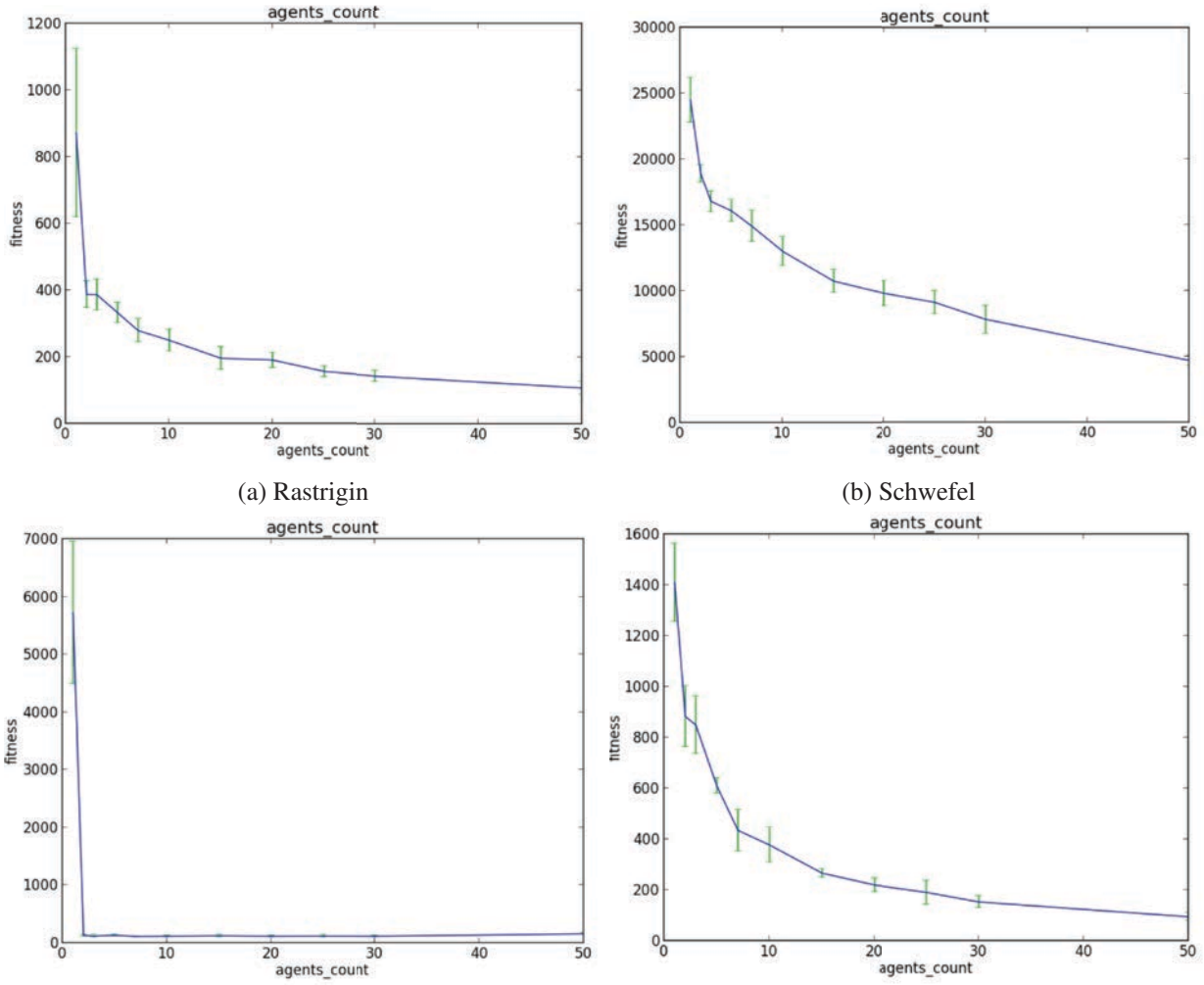


(c) Griewank, mutation probability 0.3



(d) Griewank, mutation probability 0.7

Figure 8. Final fitness of the tackled Griewank problem for different probabilities depending on the mutation radius.



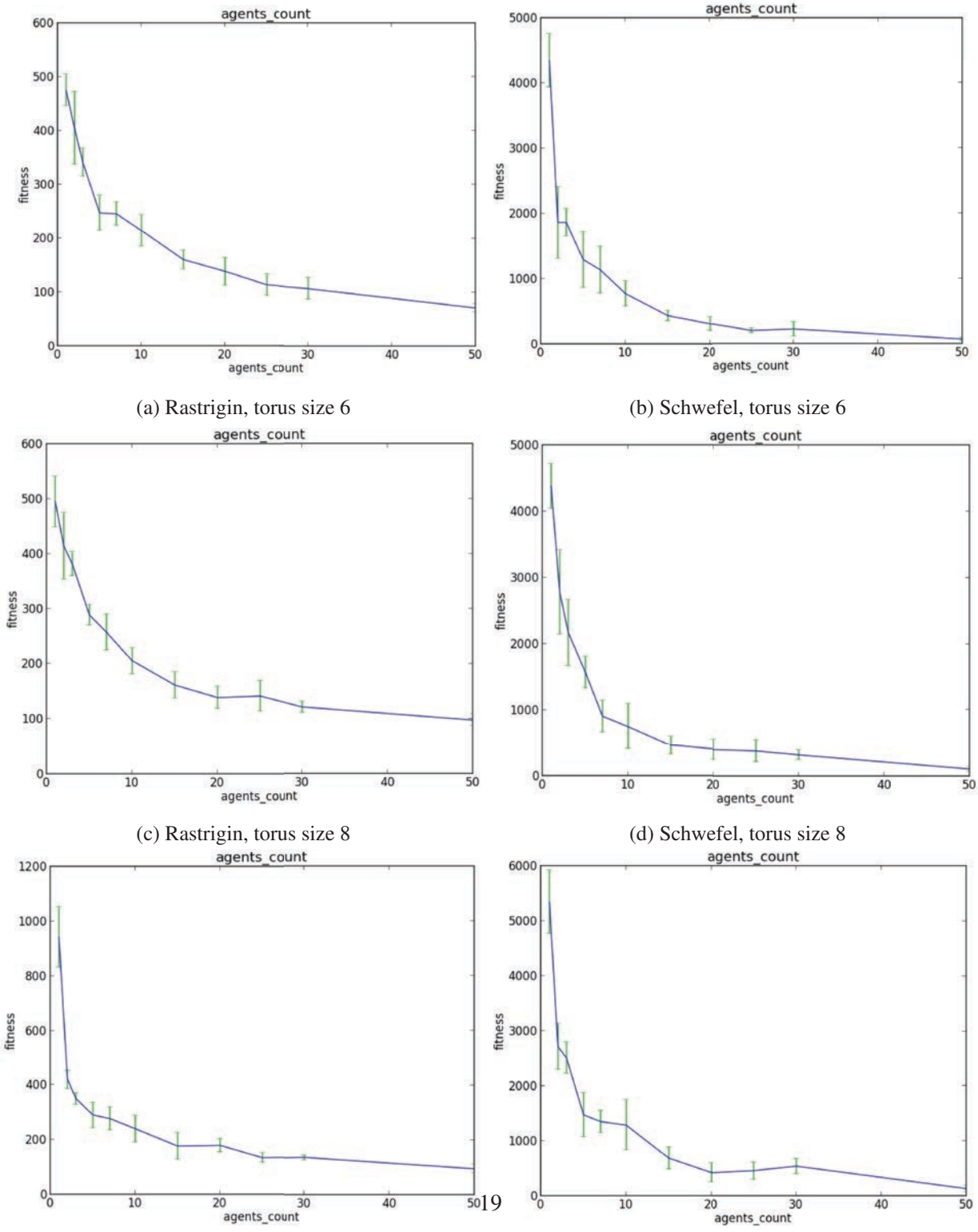
(a) Rastrigin

(b) Schwefel

(c) Rosenbrock

(d) Griewank

Figure 9. Final fitness of the tackled different problems for different number of evolutionary islands.



(a) Rastrigin, torus size 6

(b) Schwefel, torus size 6

(c) Rastrigin, torus size 8

(d) Schwefel, torus size 8

(e) Rastrigin, torus size 10

(f) Schwefel, torus size 10

Figure 10. Final fitness of the tackled Rastrigin and Schwefel for different sizes of torus depending on the number of islands.

Conclusion

Summing up and recalling again the well-known "no free lunch theorem" by Wolpert and MacReady [31], one can state that there will always be a place in the computational sciences for another metaheuristic. At the same time one has to keep in mind quite surprising, yet very true findings of scientists like Sorensen [28] and propose new metaheuristics carefully, with proper theoretical background and when they are really needed.

In the opinion of authors, EMAS belongs to such metaheuristics, bringing together evolution and theagency, and being able to solve many problems for the last 20 years advantageously over other metaheuristics. Theoretical background was provided by Byrski and Schaefer [5] proving that EMAS is an universal optimization algorithm.

Byrski also gave broad experimental evidence on the applicability of EMAS to solving high-dimensional benchmark functions in [7]. The results presented in this paper can be treated as a follow-up and extension of this work, tackling more problems in significantly broader range of parameters, and can be used as a reference for all interested in utilizing this efficient and efficacious metaheuristic in continuous optimization problems.

The research on EMAS is and will be continued, focusing on different hybridization of this metaheuristic algorithm, e.g. [25, 4] and different ways of implementation, e.g. in order to utilize properly the available supercomputing infrastructure e.g. [29, 30]. Currently, two hybridization of EMAS are researched, namely the one with Differential Evolution (in progress, not published yet) and with Particle Swarm Optimization (preliminary results already published in [26]).

Acknowledgment

The research presented in this paper received partial support from AGH University of Science and Technology Statutory Project. The authors used the PlGrid infrastructure in order to realize the research presented in this paper.

References

[1] P. Adamidis. Parallel evolutionary algorithms: A review. In Proceedings of the 4th Hellenic-European Conference on Computer Mathematics and its Applications (HERCMA 1998), Athens, Greece, 1998.

[2] T. Bäck and H.-P. Schwefel. Evolutionary computation: An overview. In T. Fukuda and T. Furuhashi, editors, Proceedings of the Third IEEE Conference on Evolutionary Computation. IEEE Press, 1996.

[3] A. Byrski, M. Kisiel-Dorohinicki, and E. Nawarecki. Agent-based evolution of neural network architecture. In M. Hamza, editor, Proc. of the IASTED Int. Symp.: Applied Informatics. IASTED/ACTA Press, 2002.

[4] A. Byrski, M. Kisiel-Dorohinicki, and N. Tusinski. Extending estimation of distribution algorithms with agent-based computing inspirations. Transactions on Computational Collective Intelligence, XXVII, 2017.

[5] A. Byrski, R. Schaefer, M. Smołka, and C. Cotta. Asymptotic guarantee of success for multi-agent memetic systems. Bulletin of the Polish Academy of Sciences – Technical Sciences, 61(1), 2013.

[6] Aleksander Byrski. Tuning of agent-based computing. Computer Science, 14(3):491, 2013.

[7] Aleksander Byrski. Tuning of agent-based computing. Computer Science (accepted), 2013.

[8] Aleksander Byrski, Roman Debski, and Marek Kisiel-Dorohinicki. Agent-based computing in an augmented cloud environment. Computer Systems Science and Engineering, 27(1), 2012.

[9] Aleksander Byrski, Rafal Drezewski, Leszek Siwik, and Marek Kisiel-Dorohinicki. Evolutionary multi-agent systems. Knowledge Eng. Review, 30(2):171–186, 2015.

[10] Aleksander Byrski and Marek Kisiel-Dorohinicki. Immune-based optimization of predicting neural networks. In Vaidy S. Sunderam, Geert Dick van Albada, Peter M. A. Sloot, and Jack Dongarra, editors, Computational Science – ICCS 2005, pages 703–710, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[11] Aleksander Byrski and Marek Kisiel-Dorohinicki. Agent-based evolutionary and immunological optimization. In Yong Shi, Geert Dick van Albada, Jack Dongarra, and Peter M. A. Sloot, editors, Computational Science – ICCS 2007, pages 928–935, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[12] Aleksander Byrski and Marek Kisiel-Dorohinicki. Agent-based model and computing environment facilitating the development of distributed computational intelligence systems. In Gabrielle Allen, Jarosław Nabrzyski, Edward Seidel, Geert Dick van Albada, Jack Dongarra, and Peter M. A. Sloot, editors, Computational Science – ICCS 2009, pages 865–874, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[13] Aleksander Byrski and Marek Kisiel-Dorohinicki. Evolutionary Multi-agent Systems: From inspirations to applications, volume 680 of Studies in Computational Intelligence. Springer, 2017.

[14] E. Cantú-Paz. A summary of research on parallel genetic algorithms. IlliGAL Report No. 95007. University of Illinois, 1995.

[15] E. Cantú-Paz. A survey of parallel genetic algorithms. Calculateurs Paralleles, Reseaux et Systems Repartis, 10(2):141–171, 1998.

[16] K. Cetnarowicz, M. Kisiel-Dorohinicki, and E. Nawarecki. The application of evolution process in multi-agent world (MAW) to the prediction system. In M. Tokoro, editor, Proc. of the 2nd Int. Conf. on Multi-Agent Systems (ICMAS'96). AAAI Press, 1996.

[17] J. Digalakis and K. Margaritis. An experimental study of benchmarking functions for evolutionary algorithms. International Journal of Computer Mathemathics, 79(4):403–416, April 2002.

[18] Rafał Dreżewski. Co-evolutionary multi-agent system with speciation and resource sharing mechanisms. Computing and Informatics, 25(4):305–331, 2006.

[19] Rafał Dreżewski, Jan Sepielak, and Leszek Siwik. Classical and agent-based evolutionary algorithms for investment strategies generation. In Anthony Brabazon and Michael O'Neill, editors, Natural Computing in Computational Finance, volume 185 of Studies in Computational Intelligence, pages 181–205. Springer-Verlag, 2009.

[20] Stan Franklin and Art Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. In Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages, ECAI '96, pages 21–35, London, UK, UK, 1997. Springer-Verlag.

[21] D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In J. J. Grefenstette, editor, Proceedings of the 2nd International Conference on Genetic Algorithms, pages 41–49. Lawrence Erlbaum Associates, 1987.

[22] L. Hanna and J. Cagan. Evolutionary multi-agent systems: An adaptive and dynamic approach to optimization. ASME Journal of Mechanical Design, 131(1), 2009.

[23] M. Kisiel-Dorohinicki. Agent-oriented model of simulated evolution. In William I. Grosky and Frantisek Plasil, editors, SofSem 2002: Theory and Practice of Informatics, volume 2540 of LNCS. Springer-Verlag, 2002.

[24] Marek Kisiel-Dorohinicki. Agent-based models and platforms for parallel evolutionary algorithms. In Marian Bubak, Geert Dick van Albada, Peter M. A. Sloot, and Jack Dongarra, editors, Computational Science - ICCS 2004, pages 646–653, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[25] Wojciech Korczynski, Aleksander Byrski, and Marek Kisiel-Dorohinicki. Buffered local search for efficient memetic agent-based continuous optimization. Journal of Computational Science, 20:112 – 117, 2017.

[26] L. Placzkiewicz, M. Sendera, A. Szlachta, M. Paciorek, A. Byrski, M. Kisiel-Dorohinicki, and M. Godzik. Hybrid swarm and agent-based evolutionary optimization. In Proc. of International Conference on Computational Science, Wuxi, China (accepted). 2018.

[27] Leszek Siwik and Rafał Dreżewski. Agent-based multi-objective evolutionary algorithms with cultural and immunological mechanisms. In Wellington Pinheiro dos Santos, editor, Evolutionary computation, pages 541–556. In-Teh, 2009.

[28] Kenneth Sörensen. Metaheuristicsthe metaphor exposed. International Transactions in Operational Research, 22(1):3–18, 2015.

[29] Jan Stypka, Wojciech Turek, Aleksander Byrski, Marek Kisiel-Dorohinicki, Adam D. Barwell, Christopher Brown, Kevin Hammond, and Vladimir Janjic. The missing link! a new skeleton for evolutionary multi-agent systems in erlang. International Journal of Parallel Programming, 46(1):4–22, Feb 2018.

[30] Wojciech Turek, Jan Stypka, Daniel Krzywicki, Piotr Anielski, Kamil Pietak, Aleksander Byrski, and Marek Kisiel-Dorohinicki. Highly scalable erlang framework for agent-based metaheuristic computing. J. Comput. Science, 17:234–248, 2016.

[31] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, 67(1), 1997.

[32] M.J. Wooldridge. An Introduction to Multiagent Systems. John Wiley & Sons, 2009.

[33] Weicai Zhong, Jing Liu, Mingzhi Xue, and Licheng Jiao. A multiagent genetic algorithm for global numerical optimization. IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics, 34(2):1128–1141, 2004.

Michał Mizera obtained M.Sc. in 2017 at the Department of Computer Science of the AGH University of Science and Technology in Krakow, Poland. His interests cover artificial intelligence (including genetic algorithms) and systems architecture. Currently he works as Senior Software Developer at company providing solutions for travel industry.

Pawel Nowotarski obtained M.Sc. in 2017 at the Department of Computer Science of the AGH University of Science and Technology in Krakow, Poland. His main domains of interest are: artificial intelligence, evolutionary algorithms and multi-agent systems. Pawel currently works as Software Developer at international company providing high performance solutions for travel industry.

Aleksander Byrski obtained Ph.D. in 2007 and D.Sc. in 2013 at the Department of Computer Science of the AGH University of Science and Technology in Krakow, Poland. His main research interests are metaheuristics, agent-based systems, high performance computing and simulation. He works as associate professor at the Department of Computer Science AGH.

Marek Kisiel-Dorohinicki obtained Ph.D. in 2001 and D.Sc. in 2013 at the Department of Computer Science of the AGH University of Science and Technology in Krakow, Poland.
His main research interests are metaheuristics, agent-based systems and criminal analysis. He works as associate professor at the Department of Computer Science AGH.