MEMETIC COMPUTING IN SELECTED AGENT-BASED EVOLUTIONARY SYSTEMS

Aleksander Byrski, Marek Kisiel Dorohinicki
Department of Computer Science
AGH University of Science and Technology
Al. Mickiewicza 30, 30-059 Krakow, Poland
Email: {olekb,doroh}@agh.edu.pl

KEYWORDS

evolutionary algorithms; continuous optimisation; multi-agent computing systems; memetic computation

ABSTRACT

In the paper an application of selected agent-based evolutionary computing models, such as flock-based multi agent system (FLOCK) and evolutionary multi-agent system (EMAS), to the problem of continuous optimisation is presented. It turns out, that hybridizing of agent-based paradigm with evolutionary computation brings a new quality to the meta-heuristic field, easily enhancing static individuals with possibilities of perception and interaction with other agents. The examination of selected benchmarks leads to the observation regarding the overall efficiency of the systems in comparison to the standard genetic algorithm (as defined by Michalewicz) and memetic versions of all the systems. The experiments confirm that the efficiency is dependent on the problem, however, the observed number of fitness function calls makes EMAS dominate over its competitors. This feature makes EMAS a promising solution for the problems with complex fitness functions, (such as inverse problems).

INTRODUCTION

Recently both software agents and evolutionary computation have been gaining more and more applications in various domains. The key concept in multi-agent systems (MAS) constitute intelligent interactions. Evolutionary computation can be perceived as a universal technique for solving optimisation problems. This paper concerns a hybrid evolutionary-agent approach. In contrary to typical approaches reported in literature (see e.g. [17] or [8] for a review) we assume that evolutionary processes are incorporated into a multi-agent system at a population level [10]. The advantages of agents autonomy in this case appear in the possibility of enhancing evolutionary processes with agents interactions, e.g., making possible undertaking autonomous decisions regarding the reproduction by choosing the partner agents.

The paper aims to present selected results of the experiments regarding the selected evolutionary agent-based computing systems. The stress is put on evolutionary multi-agent systems (EMAS), which over the years proved useful in different optimisation problems (e.g., single-criteria, multi-criteria, discrete, continuous) [3].

In this paper, one of the most important features of EMAS is presented—a relatively low computational cost measured as a number of fitness function calls. This makes the system appear well-suited for the problems utilising complex fitness function, requiring e.g., running a simulation to compute the value of the fitness (see inverse problems [1]). This conclusion is based on premise of the presented experimental results concerning popular continuous optimisation benchmarks in comparison to two selected algorithms, popular simple genetic algorithm operating in real-value space [13] and flock-based evolutionary system [11] being another agent-based computational technique proposed by the authors. All the presented algorithms are examined in memetic and standard versions (i.e. with local-search technique enabled or disabled).

In the course of paper, after recalling the basics of evolutionary, memetic and agent-based computation and presenting the concepts of the examined systems, the experimental results are given and discussed, and in the end, the conclusions are drawn.

Agent-based computing paradigm has already been studied, and supported by a number of scientific projects. One of such notable examples is ParaPhrase[1], focusing on supplying hybrid CPU/GPU computing infrastructure via dedicated virtualisation tools. The computing experiments presented in this paper may be treated as preliminary results, planned to be adapted and ported to ParaPhrase infrastructure.

EVOLUTIONARY AND MEMETIC ALGORITHMS

In **evolutionary algorithms** [13] the problem is encoded in a special way (genotype) and random populations of potential solutions are constructed. Based on the existing fitness function (evaluating the genotype), selection is performed (so the mating pool is created) and based on the mating pool, the subsequent population is created with use of predefined variation operators (such as crossover and mutation). The process continues until some stopping condition is reached (e.g., number of generations, lack of changes in the best solution found so far).

It may be seen, that the population of potential (encoded) solutions of a given problem is decomposed into evolutionary islands (there is also a possibility of migration between them) [6]. Such algorithms are usually called "parallel evolutionary

[1] http://paraphrase-ict.eu

Proceedings 28th European Conference on Modelling and Simulation ©ECMS Flaminio Squazzoni, Fabio Baronio, Claudia Archetti, Marco Castellani (Editors)
ISBN: 978-0-9564944-8-1 / ISBN: 978-0-9564944-9-8 (CD)

algorithms" (PEA). The most important fact is that the evolutionary algorithm is common to all islands, all operators are applied one by one, during each of generations, to all parts of the population. After meeting some kind of stopping condition, the best solution so far is presented as the optimal one. One of the main drawbacks of such an approach is global (god-like) selection algorithm—possibilities of its de-globalisation will be described later.

Solving optimisation problems with evolutionary algorithms requires that the following must be defined [2]: appropriate encoding of the solutions, crossover and mutation operators appropriate for the encoding, choosing a selection mechanism, and possibly other components of specialized techniques, like configuring topology of islands and migration strategies for the island model of parallel evolutionary algorithms.

Memetic algorithms [14], [12], [15] are population-based techniques that hybridize other meta-heuristics, usually by integrating local search (LS) within the population-based search engine. One of the most important feature of memetic algorithms increased exploitation ability that must be carefully balanced with exploration power of the population heuristics, in order to retain diversity.

In the most cases, two types of memetic systems are defined [15], [12], [16]:

- Baldwinian evolutionary algorithms—in these algorithms the fitness of the individual is evaluated based not only on genotype, but rather on the genotype of one of its potential successors (after, e.g., applying some local-search technique in the course of mutation of the genotype, being a starting point for this local search) —the genotype of the invidual remains intact, in the end).
- Lamarckian evolutionary algorithms—in these algorithms, the fitness of the individual is computed after applying local search method to mutate the genotype of the individual (the genotypes is changed, so Lamarckian evolution may be perceived as applying a complex mutation operator).

One of the main advantages of these systems is usually quick attaining of the target optimum, however applying such complex mutation makes the system focused on the exploitation and because of that, additional methods for enhancing the diversity of the population (even such simple, as fitness sharing or crowding [13]) are desired to retain the balance between exploration and exploitation.

Hybridizing memetics with agent-based approaches leads also to the possibility of controlling certain parameters of e.g., memetic-based mutation, adaptation of their value depending on the observation conducted in the environment etc.

INTELLIGENT DECENTRALISATION: FROM INDIVIDUALS TO AGENTS

A **flock-based architecture** may be treated as an extension of the classical island model of evolutionary algorithm (PEA) providing additional level of organisation of the system [11]. Subpopulations on the evolutionary islands (distribution units) are divided into flocks, where independently conducted processes of evolution are managed by agents (see Fig. 1). It is possible to distinguish two levels of migration:

- exchange of individuals between flocks on one island,
- migration of flocks between islands.

Also merging of flocks containing similar individuals or dividing of flocks with large diversity allows for dynamic changes of population structure to possibly well reflect the problem to be solved.

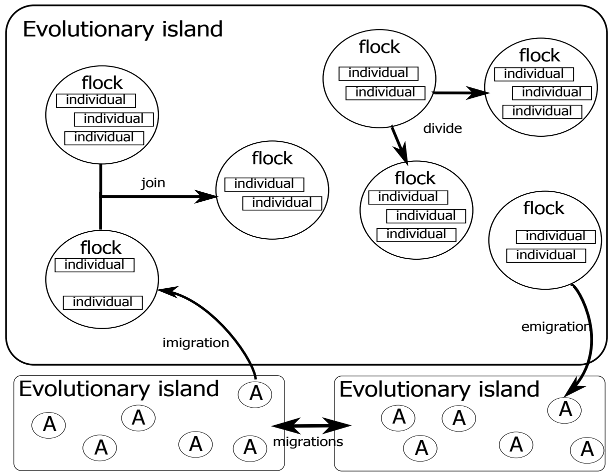


Fig. 1. Flock-based evolutionary system

Agents of an **evolutionary multi-agent system** (EMAS) represent or generate solutions for a given optimisation problem. They are located on islands, which constitute their local environment where direct interactions may take place, and represent a distributed structure of computation (see Fig. 2). Obviously, agents are able to change their location, which allows for diffusion of information and resources all over the system [10].

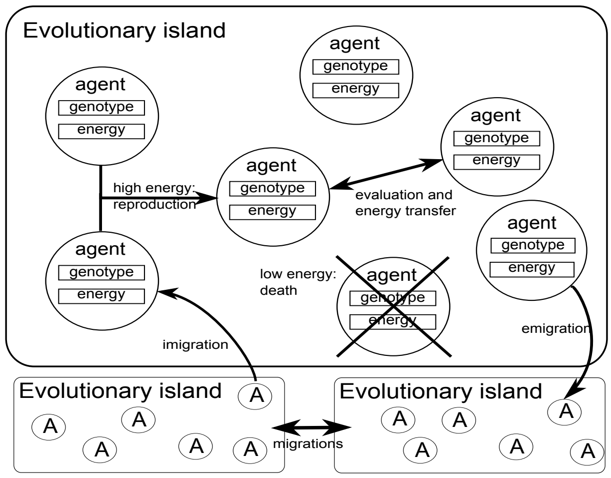


Fig. 2. Evolutionary multi-agent system

Assuming that no global knowledge is available (which makes it impossible to evaluate all individuals at the same time) and autonomy of the agents (which causes that reproduction is achieved asynchronously), selection is based on the non-renewable resources [7]. Thus a decisive factor of the agent's activity is its fitness, expressed by the amount of non-renewable resource it possesses. The agent gains resources as a reward for 'good' behaviour, and looses resources as a consequence of 'bad' behaviour. Selection is realised in such a way that agents with a lot of resources are more likely to reproduce, while low energy increases the possibility of death.

In the simplest possible model of an evolutionary multi-agent system there is one type of agents and one resource defined. Genotypes of agents represent feasible solutions to the problem.

Energy is exchanged by agents in the process of evaluation. The agent increases its energy when it finds out that one (e.g. randomly chosen) of its neighbours, has lower fitness. In this case, the agent takes part of its neighbour's energy, otherwise, it passes part of its own energy to the evaluated neighbour. The level of life energy triggers actions of death and reproduction (low energy causes death while high energy makes reproduction possible).

Summing up, EMAS agents may perform reproduction action (producing new offspring), death action (in case of low level of energy), evaluation action (in order to exchange the energy based on the fitness function value) and migration action (in order to spread the genetic information among the evolutinary islands). Each action is attempted randomly with certain probability, and it is performed only when their basic preconditions are met (e.g. an agent may attempt to perform the action of reproduction, but it will reproduce only if its energy rises above certain level and it meets an appropriate neighbour).

EXPERIMENTAL RESULTS

In order to examine the features of standard and agent-based computing systems, they were implemented using AgE computing platform (http://age.iisg.agh.edu.pl). All parameters of the the systems under consideration (SGA (Michalewicz version [13], FLOCK and EMAS both in standard and memetic versions) were chosen in such way, that the comparison between them could be possible and the perceived differences could depend only on the intrinsic features of the algorithms.

Thus, the configurations of SGA, FLOCK and EMAS were as follows:

- ALL real-value encoding, discrete recombination (offspring gets parents' genes one by one, from each parent with certain probability), normal mutation with standard deviation 0.3 and probability 0.2, stopping condition: reaching 1000th step of the computation.
- SGA 100 individuals, tournament selection.
- FLOCK 5 flocks 20 individuals each, tournament selection, the flocks join together when their populations overlap and divide, when the diversity of the population is low.
- EMAS in the beginning, there are 30 individuals, the population number stabilises at about 100 individuals, starting energy: 30 units, total energy constant: 900 units, reproduction at 15 units, during evaluation agents exchange 5 units, energy of death: 0 units.
- MEM. all memetic versions utilized a Lamarckian mutation based on steepest descent there are three attempts to mutate the genotype, each time the next proposed genotype is sampled three times in the vicinity of the individual, and the best proposition is chosen.

The considered benchmark problems were popular De Jong, Ackley, Rastrigin, Griewank and Rosenbrock functions [9] described in 10 dimensions. All the experiments were repeated 30 times and the standard deviation was computed as a measure of repeatability.

In Fig. 3 the progress of optimisation process conducted in all examined systems was presented. It was displayed as the best fitness observed in subsequent steps of the computation. In order to distinguish individual features of each process, logarithmic scale was used on ordinate axis.

Recalling "no free lunch theorem" [18] the authors were not aiming at proving that one of the examined systems proves as the best for all benchmark used. Instead, certain information about the features of each system may be discovered, when looking at the graphs in Fig. 3 and the tables later on. E.g., quick look at the graphs reveals, that almost independent on the system used, the Rosenbrock problem, being a well known deceptive function, remains the most difficult one. On the other side, De Jong problem, being a simple convex function, appears of course the easiest one to be solved.

When comparing the effectiveness of certain computing systems relatively to the problems solved, looking at the graphs presented in Figs. 3(a), 3(c), 3(e), does not let to favour any of the systems, maybe apart from EMAS doing much better in the case of De Jong function 3(e), though it is to note, that this problem is too straightforward to prove the domination of one of optimisation methods.

When comparing memetic versions of all the examined systems (see Figs. 3(a), 3(c), 3(e)) it is easy to see, that these versions are much better in solving the given problems, than their standard versions, as they reach much better results, moreover, the descent in the direction of the optimum is quicker and the curve depicting it is steeper in the beginning of the computation.

Additional information regarding the efficiency of certain systems may be found in Tables I, II. When looking for the best obtained results throughout the all experiments, it seems, that it is hard to find one algorithm dominating the others (see, [18]).

In Tables III, IV the diversity obtained in 1000th step for the all population was shown. This measure was computed as minimum standard deviation of all genes averaged over the whole population. It is easy to see, that memetic versions of all algorithms tend to process much less diverse populations than their standard versions (a well known problem of memetic computation [15]). Diversity is also quite dependent on the problem, as the problem itself influences the distribution of the populations, see, e.g., column presenting the data gathered for Rosenbrock problem: this values are one of the highest in the table, as Rosenbrock problem, visualized in 2 dimension as quite flat surface with several bumps, allows the population to be spread more than, e.g., Rastrigin or Griewank problem, where the individuals gather in local extrema throughout the whole computation.

The most interesting results however, are presented in Table V. There, approximated number of fitness function calls computed for all conducted experiments is shown. It is easy to see, that soft selection mechanism (energetic selection) used in EMAS (both in standard and memetic verions) allowed to obtain quite similar results (see, Fig. 3 and Tables I, II) at the same time reducing the number of fitness function calls (better by two-three orders of magnitude when comparing with FLOCK or SGA).

CONCLUSIONS

In the course of the paper selected agent-based computing systems were recalled (FLOCK and EMAS) and the experimental results obtained for optimisation of several benchmark functions were given. Detailed insight into the features presented in graphs depicting the best fitness in the examined population did not allow to state, that one of the tested systems prevailed. However, classical features of memetic computation were spotted: the optimum is pursued faster in the beginning, and the diversity of these systems is lower than in the case of their standard versions.

The most important conclusion is proving, that regardless the efficiency of EMAS in comparison to other systems, it prevails in the means of fitness function calls during the computation (even by two or three orders of magnitude). This feature makes EMAS a reliable means for solving problems with complex fitness functions, such as inverse problems, evolution of neural network parameters (see, e.g., [5], [4]), and others.

In the future the authors plan to enhance the testing conditions by considering continuous and discrete benchmarks as well as increasing the dimensionality of the problems to be solved.

ACKNOWLEDGMENTS

The research presented in the paper was partially supported by the European Commission FP7 through the project ParaPhrase: Parallel Patterns for Adaptive Heterogeneous Multicore Systems, under contract no.: 288570 (http://paraphrase-ict.eu).

The research presented in this paper received partial financial support from AGH University of Science and Technology statutory project.

REFERENCES

[1] R. C. Aster, B. Borchers, and C. H. Thurber. *Parameter Estimation and Inverse Problems*. Academic Press, 2005.

[2] T. Bäck, D. Fogel, and Z. Michalewicz, editors. *Handbook of Evolutionary Computation*. IOP Publishing and Oxford University Press, 1997.

[3] A. Byrski, R. Dreżewski, L. Siwik, and M. Kisiel-Dorohinicki. Evolutionary multi-agent systems. *The Knowledge Engineering Review*, Accepted for publiFcation, 2012.

[4] A. Byrski and M. Kisiel-Dorohinicki. Evolving RBF networks in a multi-agent system. *Neural Network World*, 12(5):433–440, 2002.

[5] 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. on Applied Informatics*. IASTED/ACTA Press, 2002.

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

[7] 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.

[8] S.-H. Chen, Y. Kambayashi, and H. Sato. *Multi-Agent Applications with Evolutionary Computation and Biologically Inspired Technologies*. IGI Global, 2011.

[9] 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.

[10] M. Kisiel-Dorohinicki. Agent-based models and platforms for parallel evolutionary algorithms. In M. Bubak, G. D. van Albada, P. M. A. Sloot, and J. Dongarra, editors, *Computational Science – ICCS 2004. Part III*, volume 3038 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2004.

[11] M. Kisiel-Dorohinicki. Flock-based architecture for distributed evolutionary algorithms. In L. Rutkowski, J. Siekmann, R. Tedeusiewicz, and L. Zadeh, editors, *Artificial Intelligence and Soft Computing – ICAISC 2004*, volume 3070 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2004.

[12] N. Krasnogor and J. Smith. A tutorial for competent memetic algorithms: Model, taxonomy, and design issues. *IEEE Transactions on Evolutionary Computation*, 9(5):474–488, 2005.

[13] Z. Michalewicz. *Genetic Algorithms Plus Data Structures Equals Evolution Programs*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1994.

[14] P. Moscato. Memetic algorithms: a short introduction. In *New ideas in optimization*, pages 219–234, Maidenhead, UK, England, 1999. McGraw-Hill Ltd., UK.

[15] P. Moscato and C. Cotta. A modern introduction to memetic algorithms. In M. Gendrau and J.-Y. Potvin, editors, *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research and Management Science*, pages 141–183. Springer, 2 edition, 2010.

[16] Y.-S. Ong, M.-H. Lim, and X. Che. Memetic computation – past, present & future. *IEEE Computational Intelligence Magazine*, 5(2):24–36, 2010.

[17] R. Sarker and T. Ray. *Agent-Based Evolutionary Search*. Springer, 2010.

[18] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

TABLE I. AVERAGE BEST FITNESS OBTAINED IN 1000TH STEP AND STANDARD DEVIATION (COMPUTED FOR 30 RUNS) FOR MEMETIC AND STANDARD VERSIONS OF EXAMINED COMPUTING SYSTEMS (ACKLEY, GRIEWANK AND RASTRIGIN PROBLEMS)

System	Ackley		Griewank		Rastrigin	
SGA Std	$0.057\,237\,12$	$\pm 0.007\,241\,441$	$0.085\,502\,67$	$\pm 0.061\,758\,09$	$0.340\,349$	$\pm 0.107\,659\,7$
SGA Mem	$2.148\,808 \times 10^{-5}$	$\pm 7.754\,295 \times 10^{-6}$	$0.060\,106\,21$	$\pm 0.029\,837\,69$	$6.493\,436 \times 10^{-8}$	$\pm 4.583\,01 \times 10^{-8}$
FLOCK Std	$0.076\,686\,35$	$\pm 0.014\,271\,68$	$0.541\,716\,8$	$\pm 0.456\,685\,6$	$1.785\,208$	$\pm 0.775\,546\,1$
FLOCK Mem	$4.631\,602 \times 10^{-5}$	$\pm 1.580\,519 \times 10^{-5}$	$0.082\,211\,74$	$\pm 0.058\,706\,72$	$4.006\,674 \times 10^{-7}$	$\pm 2.747\,038 \times 10^{-7}$
EMAS Std	$0.047\,309\,22$	$\pm 0.148\,428\,8$	$27.977\,86$	$\pm 18.809\,17$	$3.011\,314$	$\pm 1.498\,797$
EMAS Mem	$0.000\,431\,029\,2$	$\pm 0.000\,147\,903\,2$	$3.008\,084$	$\pm 4.320\,769$	$3.958\,766 \times 10^{-5}$	$\pm 3.085\,049 \times 10^{-5}$

TABLE II. AVERAGE BEST FITNESS OBTAINED IN 1000TH STEP AND STANDARD DEVIATION (COMPUTED FOR 30 RUNS) FOR MEMETIC AND STANDARD VERSIONS OF EXAMINED COMPUTING SYSTEMS (ROSENBROCK AND DE JONG PROBLEMS)

System	Rosenbrock		De Jong	
SGA Std	$2.706\,568$	$\pm 1.774\,557$	$0.001\,540\,284$	$\pm 0.000\,387\,838$
SGA Mem	$2.210\,916$	$\pm 1.739\,984$	$4.583\,541 \times 10^{-10}$	$\pm 3.120\,165 \times 10^{-10}$
FLOCK Std	$3.884\,875$	$\pm 2.081\,934$	$0.002\,535\,208$	$\pm 0.000\,588\,550\,1$
FLOCK Mem	$0.901\,195\,7$	$\pm 1.036\,577$	$1.871\,672 \times 10^{-9}$	$\pm 1.202\,404 \times 10^{-9}$
EMAS Std	$25.200\,49$	± 101.7666	$1.748\,485 \times 10^{-6}$	$\pm 9.186\,823 \times 10^{-7}$
EMAS Mem	$3.940\,108$	$\pm 3.054\,97$	$6.333\,364 \times 10^{-8}$	$\pm 3.935\,681 \times 10^{-8}$

TABLE III. AVERAGE DIVERSITY OBTAINED IN 1000TH STEP AND STANDARD DEVIATION (COMPUTED FOR 30 RUNS) FOR MEMETIC AND STANDARD VERSIONS OF EXAMINED COMPUTING SYSTEMS (ACKLEY, GRIEWANK AND RASTRIGIN PROBLEMS)

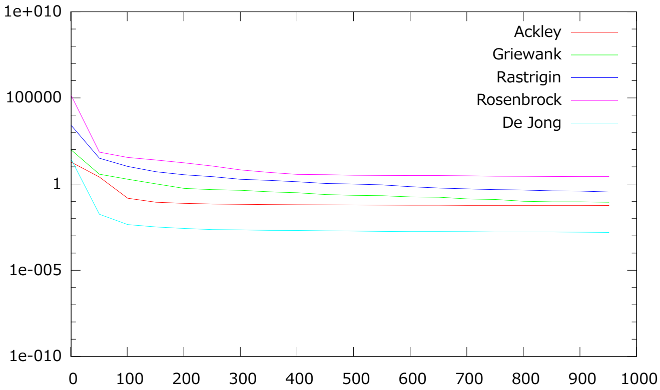
System	Ackley		Griewank		Rastrigin	
SGA Std	$0.073\,756\,12$	$\pm 0.017\,941\,05$	$0.073\,891\,11$	$\pm 0.016\,700\,04$	$0.078\,326\,87$	$\pm 0.026\,888\,11$
SGA Mem	$1.027\,91 \times 10^{-22}$	$\pm 2.331\,456 \times 10^{-22}$	$6.458\,626 \times 10^{-22}$	$\pm 1.118\,568 \times 10^{-21}$	$2.841\,09 \times 10^{-22}$	$\pm 8.269\,988 \times 10^{-22}$
FLOCK Std	$0.068\,397\,06$	$\pm 0.012\,846\,73$	$3.282\,302$	± 3.3432	$0.146\,949\,1$	$\pm 0.085\,565\,59$
FLOCK Mem	$3.269\,656 \times 10^{-6}$	$\pm 2.644\,61 \times 10^{-6}$	$1.244\,616$	$\pm 2.017\,345$	$2.957\,484 \times 10^{-6}$	$\pm 3.169\,475 \times 10^{-6}$
EMAS Std	$0.015\,511\,87$	$\pm 0.006\,212\,766$	$0.098\,006\,43$	$\pm 0.076\,427\,53$	$0.016\,464\,35$	$\pm 0.009\,102\,824$
EMAS Mem	$2.177\,581 \times 10^{-21}$	$\pm 4.648\,354 \times 10^{-21}$	$0.047\,304\,02$	$\pm 0.065\,510\,94$	$4.842\,205 \times 10^{-21}$	$\pm 1.743\,64 \times 10^{-20}$

TABLE IV. AVERAGE DIVERSITY OBTAINED IN 1000TH STEP AND STANDARD DEVIATION (COMPUTED FOR 30 RUNS) FOR MEMETIC AND STANDARD VERSIONS OF EXAMINED COMPUTING SYSTEMS (ROSENBROCK AND DE JONG PROBLEMS)

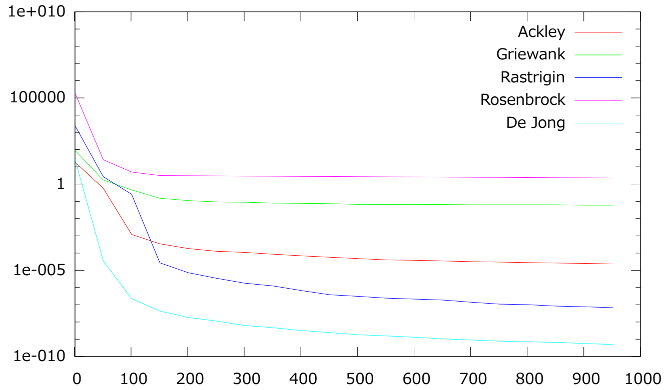
System	Rosenbrock		De Jong	
SGA Std	$0.069\,791\,13$	$\pm 0.015\,978\,85$	$0.072\,843\,56$	$\pm 0.013\,871\,39$
SGA Mem	$7.444\,279 \times 10^{-5}$	$\pm 0.000\,127\,459\,5$	$2.007\,843 \times 10^{-22}$	$\pm 4.424\,505 \times 10^{-22}$
FLOCK Std	$0.126\,153\,6$	$\pm 0.088\,960\,07$	$0.071\,510\,14$	$\pm 0.009\,508\,027$
FLOCK Mem	$0.014\,950\,56$	$\pm 0.022\,712\,15$	$1.288\,522 \times 10^{-6}$	$\pm 2.031\,521 \times 10^{-6}$
EMAS Std	$0.016\,752\,47$	$\pm 0.009\,061\,646$	$0.013\,228\,51$	$\pm 0.006\,898\,98$
EMAS Mem	$8.764\,745 \times 10^{-5}$	$\pm 0.000\,389\,931\,5$	$1.934\,059 \times 10^{-21}$	$\pm 3.741\,526 \times 10^{-21}$

TABLE V. AVERAGE NUMBER OF FITNESS CALLS DURING 1000 STEPS (COMPUTED FOR 30 RUNS) FOR MEMETIC AND STANDARD VERSIONS OF EXAMINED COMPUTING SYSTEMS

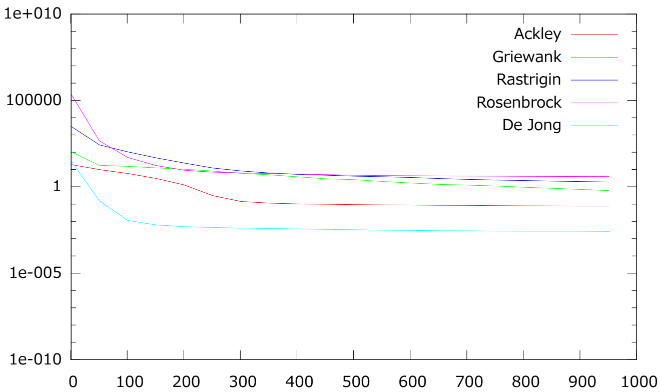
System	Ackley	Griewank	Rastrigin	Rosenbrock	De Jong
SGA Std	100 000	100 000	100 000	100 000	100 000
SGA Mem	299 800	299 800	299 800	299 800	299 800
FLOCK Std	85 386.6664	97 800.0024	93 293.3336	87 506.6668	74 133.3328
FLOCK Mem	255 640	287 200	243 640	211 480	238 559.99
EMAS Std	349.466 87	371.899 98	364.866 71	359.433 51	365.633 22
EMAS Mem	963.001 05	996.299 97	966.602 61	925.199 94	965.600 55



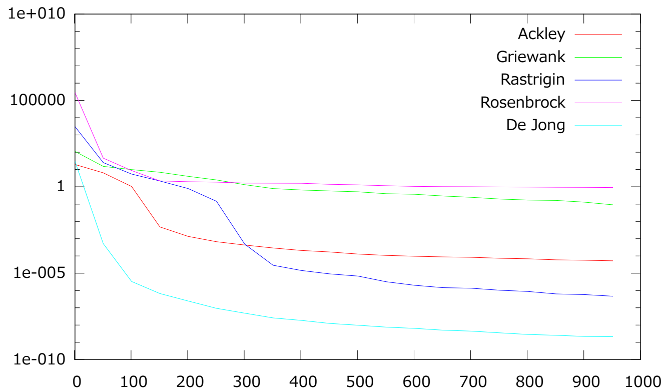
(a) Standard SGA best fitness



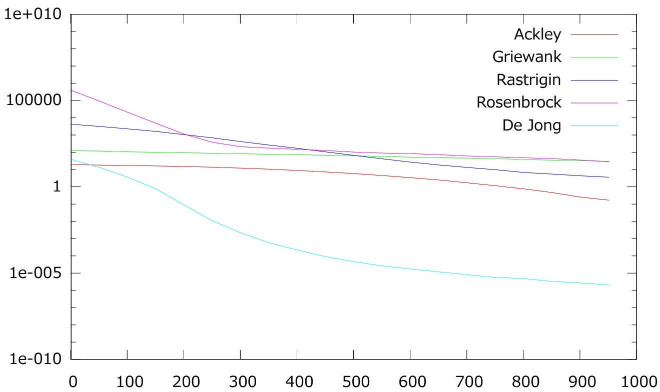
(b) Memetic SGA best fitness



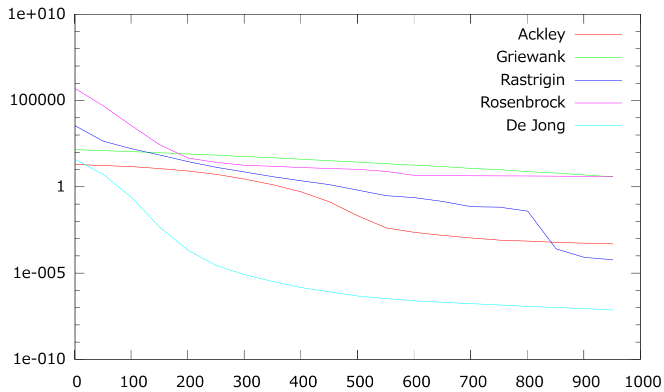
(c) Standard FLOCK best fitness



(d) Memetic FLOCK best fitness



(e) Standard EMAS best fitness



(f) Memetic EMAS best fitness

Fig. 3. Fitness for memetic and standard versions of examined computing systems