Paper

Agent-based Optimization of Advisory Strategy Parameters

Mateusz Polnik, Mateusz Kumięga, and Aleksander Byrski

Department of Computer Science, AGH University of Science and Technology, Kraków, Poland

Abstract—In this paper, an application of Evolutionary Multi-agent Systems (EMAS) and its memetic version to the optimization of advisory strategy (in this case, Sudoku advisory strategy) is considered. The problem is tackled using an EMAS, which has already proven as a versatile optimization technique. Results obtained using EMAS and Parallel Evolutionary Algorithm (PEA) are compared. After giving an insight to the possibilities of decision support in Sudoku solving, an exemplary strategy is defined. Then EMAS and its memetic versions are discussed, and experimental results concerning comparison of EMAS and PEA presented.

Keywords*—*global optimization, memetic computing, multi-agent computing.

1. Introduction

Decision support constitutes a broad range of different techniques, mostly related to artificial intelligence, aimed at helping the human (decision-maker) in different activities, such as choosing the most feasible strategy for investing in financial instruments, performing a diagnosis of a faulty system or predicting product revenue in the market [1].

Sometimes, a valuable advice may be given using a predefined model that will help in simulating or solving a certain task. Such model may become a source of knowledge, straightforwardly supporting the user in performing certain decision-making tasks. Such a model may be proposed, e.g., in the form of a set of equations constructed by an expert, but also, it is possible, that may be constructed in an automated way [2].

As an example, the process of optimization of neural network architecture may be mentioned (as neural networks may in a natural way become a part of decision support system and serve as means for solving approximation problems, e.g. classification and prediction, as well as control problems, i.e. management of some process or device). Even though the use of neural networks replaces the necessity to solve the problem in a deterministic way, one still needs to define network parameters, such as its structure, learning coefficients etc., which should be suitable for the given problem. This usually requires carrying out numerous experiments, so it is a very time consuming job and can be performed only by the specialists [3].

At the same time, techniques of evolutionary computation were successfully used to solve difficult search and optimization problems and it was also shown that they may be useful to support search for optimal parameters of a certain model (e.g., optimal neural network architecture). Although the classical evolutionary algorithms can be easily applied to search for optimal parameters of a certain model, additional advantages may be expressed by applying more complex search methods, such as agent-based computing.

Evolutionary Multi-agent Systems (EMAS) proposed by Cetnarowicz [4] and further developed by Byrski and Kisiel-Dorohinicki have already proven as an effective tool for dealing with global optimization problems (see, e.g., [5]–[7]). Moreover, a significant effort has been made, in order to give formal rationale for conducting the search (see, e.g., [8]–[11]) In these systems global control well known from evolutionary-like computing [12] is replaced by a distributed selection mechanism using non-renewable resources. The agents are introduced and removed from the population in the course of reproduction and death actions, influenced by the amount of resources owned by certain agents.

In this paper, an application of EMAS and its memetic version to the parametric optimization of parameters of the advisory strategy is presented. The case study is based on an original Sudoku advisory strategy, helping in choosing correct moves in the course of solving of this puzzle.

In the beginning of this work, several Sudoku advisory strategies are identified, and an original advisory strategy is presented. Later memetic agent-based computing systems are shortly discussed and finally the experimental results concerning the application of the EMAS and its memetic versions to the identification of the strategy parameters are shown and the paper is concluded.

2. Sudoku Advisory Strategies

Sudoku is a worldwide-known number-placement puzzle, in which the user is given the task to fill a 9×9 lattice with digits in such way that each column, row and each of nine 3×3 sub-lattices that compose the lattice contain all of the digits from the range 1 to 9. As a starting point, partially completed lattice is supplied, that usually has one unique solution [13], [14]. Sudoku is a NP-complete constraint satisfaction problem. The proof can be found in [15]. The fact of Sudoku's NP-completeness makes solvers using solely brute-force techniques infeasible.

In this paragraph, dedicated advisory strategies for Sudoku problem are discussed. They should not be treated as ap-

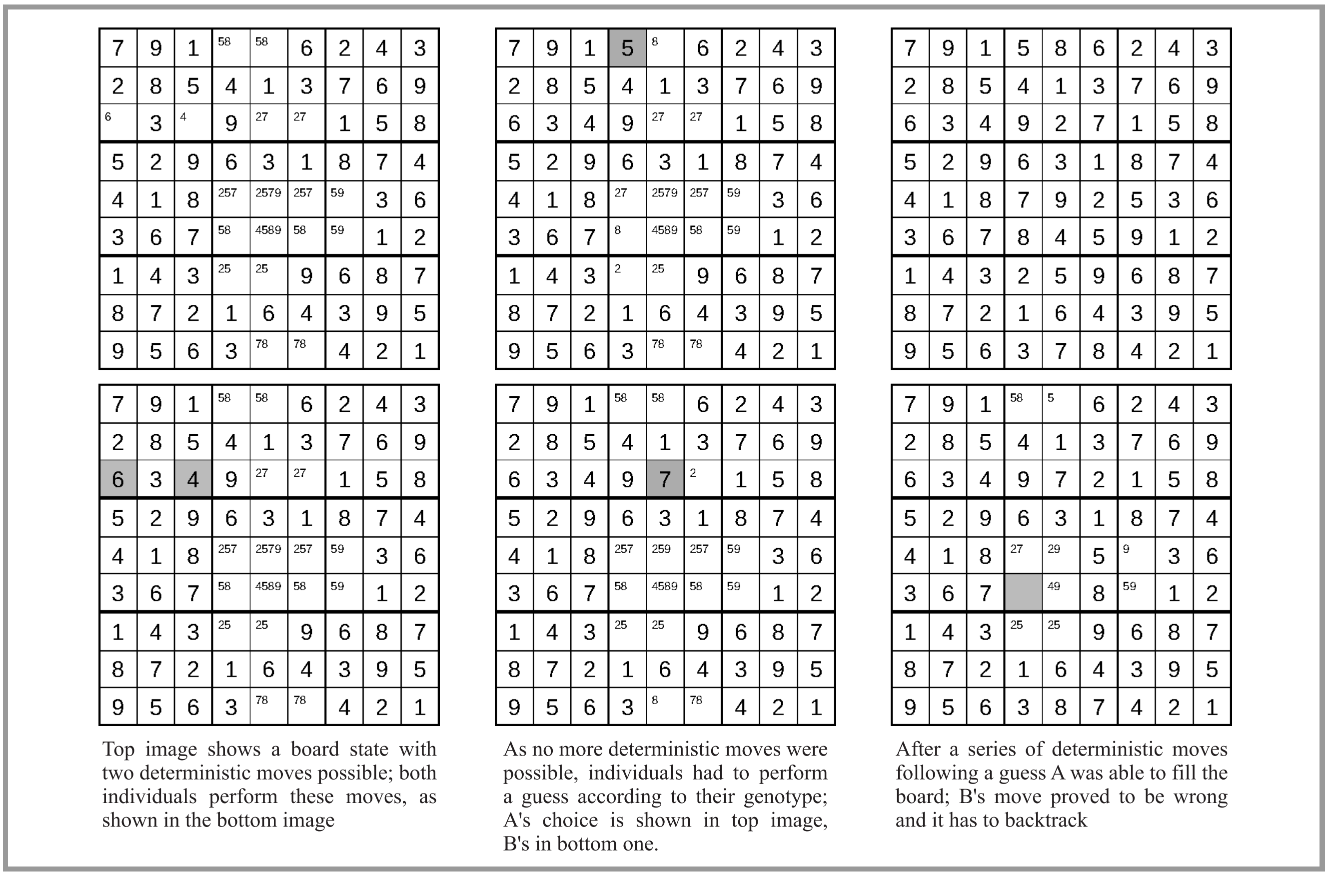

Fig. 1. Example illustrating a portion of board solving process with moves performed by two different individuals, **A** and **B**; small numbers represent $S(p)$ sets.

proaches to solve this puzzle directly, rather that as means for suggesting of the strategy, that may be used by the human. Open-source Sudoku solvers [16] implement various strategies, formulated by Sudoku community, to detect incorrect movements in advance and reduce the number of backtracks [17]. These strategies allow to either exclude some movement possibilities or make a deterministic move to satisfy Sudoku's constraints in a specific board setting. Paragraph below provides a summary of a few popular strategies described in [16], [18].

Naked Pairs – let A and B be the only candidate movements for 2 empty fields in a row, column or 3×3 block. No other empty field within the same unit can be filled with A or B.

X-Wing – let A be a candidate movement in 4 empty fields that are located in the vertices of rectangle. Any other empty field that share row or column with the rectangle's vertices cannot be filled with A.

Sword Fish – let A be a candidate movement in empty fields that share 3 different rows or columns. Any other empty field in each of the rows or columns cannot be filled with A.

The detailed overview of Sudoku advisory strategies is available in [19]. The aforementioned strategies can be also used to assess the hardness of Sudoku boards [17]. Different approach to solve and estimate Sudoku boards using continuous-time dynamical system and Richter type scale respectively is described in [15].

The Proposed Strategy

An original strategy of solving Sudoku puzzle based on the way human usually resolves this puzzle [13] is defined as follows. The strategy is supposed to point out the subsequent field of the lattice to fill it out with one of the feasible digits following there is not known any other movement enforced by Sudoku constraints. Therefore, for each field of the lattice, denoted here as (x, y), for all empty fields located in the row x and column y, and all feasible digits i, the value of the following weight function is computed:

$$W(x,y,i) = a_1 \cdot \mathit{Fill33}(x,y) + a_2 \cdot \mathit{FillRow}(x) + a_3 \cdot \mathit{FillCol}(y) + a_4 \cdot \mathit{Occ}(i)\,, \qquad (1)$$

where:

– $\mathit{Fill33}(x,y)$ is the function computing the filling level of the 3×3 block where (x,y) field is located,

– $\mathit{FillRow}(x)$ computes the value describing the filling level of the row x,

– *FillCol*(y) computes the value describing the filling level of the column y,

– *Occ*(i) computes the count of the fields with the number i.

Then the move consisting in putting the digit i into the field (x,y) is made, for the field bearing the extremal value of the function $W(x,y,i)$ (minimum or maximum, depending on the exact definition of the *Fill33*, *FillRow*, *FillCol* and *Occ* functions).

The problem of optimization of the proposed Sudoku advisory strategy may be treated as parametric optimization [20] (maximization in this case) of the function $W(x,y,i)$ depending on the parameters $a_k \in [-3,3]$, $k \in [1,4]$, used to advise the subsequent moves in Sudoku solving. This may be accomplished with an evolutionary approach. In order to do this, the pattern sought is encoded as a following weight vector:

$$[a_1,a_2,a_3,a_4], a_k \in [-3,3], k \in [1,4],$$

and the fitness function is defined as follows as a multiplicative inverse of number of non-feasible decision undertaken by the individual in the course of solving a series of lattices according to the following procedure (see also Fig. 1 for illustration).

1. Make all deterministic moves:

   - A deterministic move is the one that follows straightforwardly the Sudoku rules (in each column, row and block at most one number of certain value can be located, without backtracking or contradictions).

   - For each field p of Sudoku board a set of numbers $S(p)$ is determined, that can be filled into this field without breaking the Sudoku rules.

   - If exists p for which $S(p)$ contains only one symbol s, it is removed from all other $S(\cdot)$ located in the same 3×3 block, column or row.

   - If in the course of reducing $S(\cdot)$ sets, a new set of cardinality 1 is obtained, the procedure is repeated for this new set.

   - The algorithm is finished when all sets $S(\cdot)$ contain only one element or during the actualization of the $S(\cdot)$, no one-element set was obtained.

2. If the board is not solved, make a move according to the current strategy, otherwise finish the move according to the strategy and increase the counter of non-feasible decisions for the evaluated solution.

3. If the board is not solved, go to step 1.

3. Evolutionary Agent-based Computing

In evolutionary multi-agent systems, an agent represents solutions for a given problem. Core properties of the agent are encoded in its genotype and inherited from its parent(s) with the use of mutation and recombination operators. Besides, an agent may possess some knowledge acquired during its life, which is not inherited. Both inherited and acquired information determines the behavior of an agent in the system (phenotype). Assuming that no global knowledge is available and autonomy of the agents, selection is based on non-renewable resource, most often called *life energy* [4]. Thus a decisive factor of the agent's activity is its fitness, expressed by the amount of energy it possesses. The agent gains energy as a reward for 'good' behavior, and looses energy because of 'bad' behavior. Selection is realized in such a way that agents with high energy level are more likely to reproduce, while low energy increases the possibility of death. The agents are located on islands, which constitute their local environment where direct interactions may take places, and represent a distributed structure of computation. Obviously, agents are able to change their location, which allows for diffusion of information and resources all over the system [21].

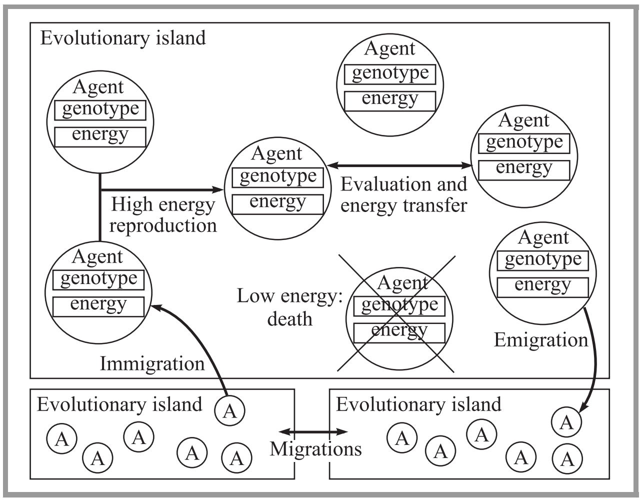

Fig. 2. Evolutionary multi-agent system.

EMAS agents may perform the following actions (see Fig. 2):

- *Reproduction* – performed when the agent's energy raises above a certain level, followed by production of a new individual in cooperation with one of its neighbors, with genotype based on parents' genotypes (crossed over and mutated) and part of energy (usually half of its initial value) also passed from each of its parents.

- *Death* – agent is removed from the system when its energy falls below a certain level, the remaining energy is distributed among its neighbors.

- *Evaluation* – agent chooses its neighbor and compares the fitness of its genotype with its own. In the case when the neighbor is better, it receives part of the agent's energy, and vice versa.
- *Migration* – agent (with some probability) may migrate, then it is removed from one evolutionary island and moved to another (random) according to predefined topology.

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 neighbor).
Implementation of Baldwinian and Lamarckian memetics in EMAS is carried out in the following way.

Baldwinian memetics – this implementation is done in a similar way as in classical evolutionary computing: the evaluation operator is enhanced with local search algorithm. The evaluation of a certain individual starts the local search from this individual and returns the fitness of the solution found instead of the original fitness value.

Lamarckian memetics – a dedicated mutation operator is called in the course of agent's life, therefore its genotype may be changed whenever this action is undertaken.

4. Experimental Results

The Sudoku strategy proposed in this paper was evaluated using four hard level Sudoku boards generated by [16]. The generator is assessing the level of difficulty by scoring the hardness of each strategy before and summing total score of movements that constitute the final solution. Although the program provides a vast array of tune settings, they were not used in order not to favor any specific solving strategy. The purpose of examining multiple boards was to exclude the risk of over fitting.
The results have been obtained using a dedicated system implemented with Python technology.
The configuration of the tested systems is presented as follows:

- *common parameters* – normal distribution-based mutation of one randomly chosen gene, single-point crossover, the descendant gets parts of its parents genotype after dividing them in one randomly chosen point, 15 individuals located on each island, all experiments were repeated 30 times and standard deviation (or other statistical measures, such as median and appropriate quartiles for box-and-whiskers plots) was computed; allopatric speciation (island model), 3 fully connected islands, 150 steps of experiment, genotype of length 4, agent/individual migration probability 0.01;
- *PEA-only parameters* – mating pool size: 8, individuals migrate independently (to different islands);
- *EMAS-only parameters* – initial energy: 100, received by the agents in the beginning of their lives, minimal reproduction energy: 90, required to reproduce, evaluation energy win/lose: 40/–40, passed from the loser to the winner, death energy level: 0, used to decide which agent should be removed from the system, boundary condition for the intra-island lattice: fixed, the agents cannot cross the borders, intra-island neighborhood: Moore's, each agent's neighborhood consists of 8 surrounding cells, size of 2-dimensional lattice as an environment: 10×10, all agents that decided to emigrate from one island, will immigrate to another island together (the same for all of them).

The local search in memetic versions was isotropic mutation – it is a method aimed at generating uniform sampling points on and within N-dimensional hyper-spheres. The idea of the Isotropic method algorithm is as follows: firstly, the N normal distributed numbers z_i are generated. Then the vectors x are computed by making a projection onto surface by dividing each generated number z_i by $r = \sqrt{\sum_{i=1}^{N} z_i^2}$. Since the z vectors are isotropically distributed, the vectors x will be of norm 1 and also isotropically distributed. Therefore the points will be distributed uniform of the hypersphere. The generation of points inside the hypersphere may be achieved by rescaling the coordinates obtained in the previous steps. While rescaling, the dimension must be taken into consideration [22]. Such a mutation was performed in 10 phases, in each of the phase 10 mutations were made and the best result was passed to the next phase.

The detailed results obtained in the course of the experiments are presented in Figs. 3 and 4. It is easy to see that for the considered problem, EMAS obtained better results for all its memetic versions while maintaining stable population of agents. What is more the diversity measures clearly indicate, that this feature is significantly better in EMAS, at least in the beginning of computation, so the exploration phase is apparently longer. Relatively high dispersion of the results calls for detailed analysis of the problem stated, and possibly to employ more sophisticated methods (e.g., niching, [23]), in order to reach and clearly report more than only one extrema of the optimized problem.

Besides visual assessment of the obtained results, an insight into attained solutions is of course necessary. In Table 1

Table 1
Final results obtained by the researched systems

System	Fitness	Standard deviation
PEA	75.6	35.51
PEA + Baldwin	93.6	68.91
PEA + Lamarck	77.4	13.82
EMAS	**37.2**	**0.4**
EMAS + Baldwin	47	59.39
EMAS + Lamarck	80.4	33.83

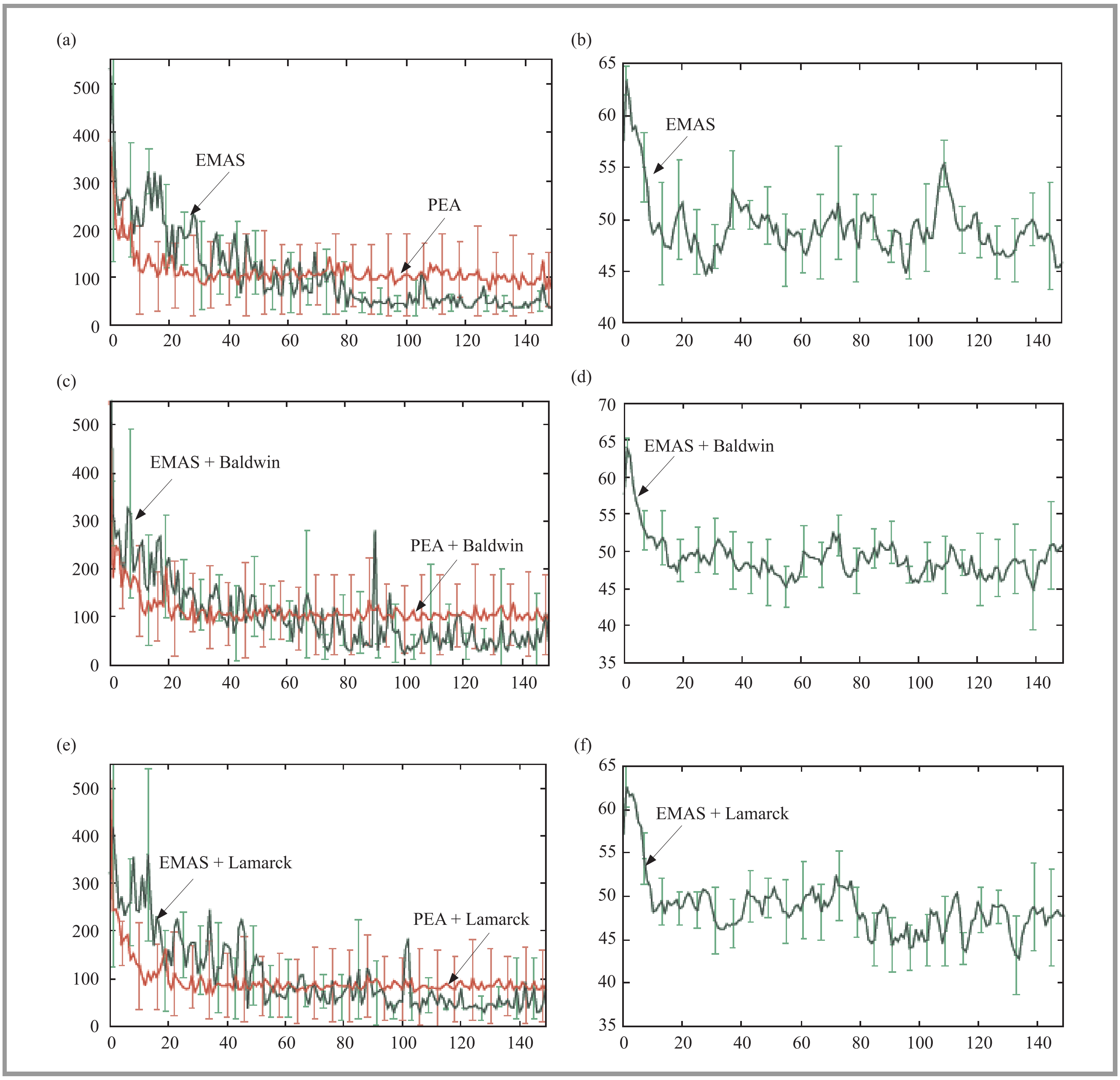

Fig. 3. Fitness and agent count obtained for all tested systems: (a) EMAS and PEA fitness, (b) EMAS agent count, (c) EMAS and PEA (Baldwin) fitness, (d) EMAS + Baldwin agent count, (e) EMAS and PEA (Lamarck) fitness, (f) EMAS + Lamarck agent count.

the obtained fitness value in the last (150th) step of the computation was presented. It is to note, that the best result has been reached by EMAS without modifications. The next one was apparently EMAS with Baldwinian memetics, unfortunately high dispersion of this results point out that it should be disqualified. In the case of Lamarckian memetics, the final result obtained is worse than in the case of PEA.

5. Conclusions

In the paper an agent-based approach to parametric optimization of advisory strategy was presented. As a case study, decision support strategy in Sudoku solving was considered. The problem of optimization of these parameters was defined based on an originally proposed decision support method, constructed and inspired by the most common way of solving this puzzle by human.

However, the main stress is put on applying the agent-based optimization metaheuristics to the above-stated problem. Here, Evolutionary Multi-agent System and its memetic modifications were considered and compared to classical Parallel Evolutionary Algorithm. The results obtained by EMAS turned out to be better than these obtained by PEA. One exception was Lamarckian memetic operator, as in this case PEA turned out to be better, but both results were in

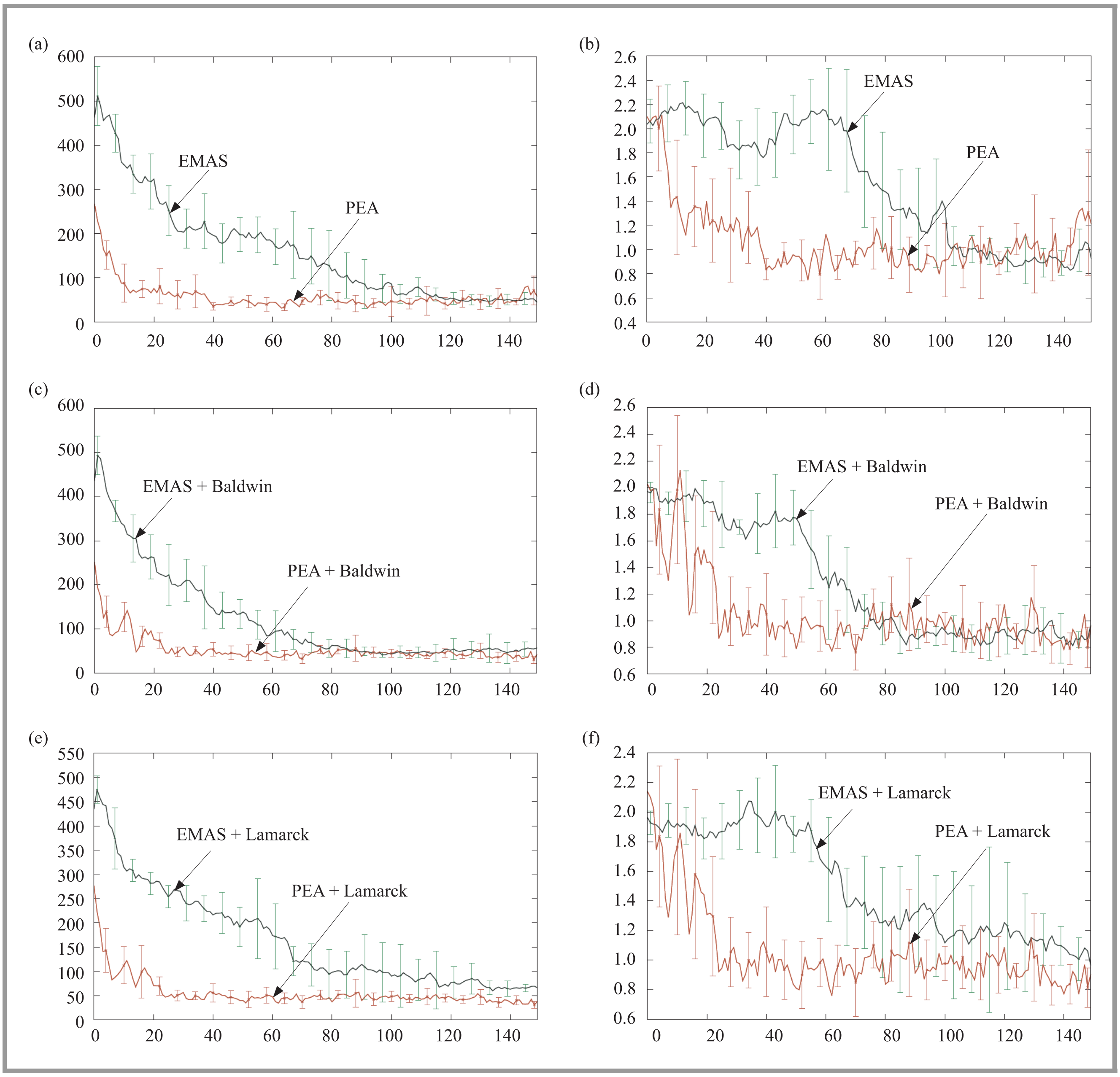

Fig. 4. Diversity obtained for all tested systems: (a) EMAS and PEA MSD diversity, (b) EMAS and PEA MOI diversity, (c) EMAS and PEA (Baldwin) MSD diversity, (d) EMAS and PEA (Baldwin) MOI diversity, (e) EMAS and PEA (Lamarck) MSD diversity, (f) EMAS and PEA (Lamarck) MOI diversity.

the same range and the dispersion measure did not allow to clearly choose the one better than another.

The obtained results support the already verified in other problems observation, that agent-based computing tends out to be better than classical algorithms. However, it should be noted that the standard deviation of the obtained outcomes is quite high, so therefore, putting an additional effort is required to cause higher repeatability of the experiments.

In the future, incorporation of the mentioned Sudoku solving strategies into the proposed method is envisaged. Tackling other difficult problems with EMAS and related approaches is also planned.

Acknowledgements

The work presented in this paper was partially supported by Polish National Science Centre research project No. N N516 500039 “Biologically inspired mechanisms in planning and management of dynamic environments”.

References

[1] V. L. Sauter, *Decision Support Systems: An Applied Managerial Approach*. New York: Wiley, 1997.

[2] A. Byrski, M. Kisiel-Dorohinicki, and E. Nawarecki, “Agent-based evolution of neural network architecture”, in *Proc. IASTED Int. Symp.: Applied Informatics*, M. Hamza, Ed. IASTED/ACTA Press, 2002.

[3] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Prentice Hall, 1998.

[4] K. Cetnarowicz, M. Kisiel-Dorohinicki, and E. Nawarecki, "The application of evolution process in multi-agent world (MAW) to the prediction system", in *Proc. 2nd Int. Conf. on Multi-Agent Sys. ICMAS'96*, pp. 26–32, AAAI Press, 1996.

[5] A. Byrski and M. Kisiel-Dorohinicki, "Agent-based evolutionary and immunological optimization", in *Proc. 7th Int. Conf. Comput. Sci. ICCS 2007*, Beijing, China, 2007. Springer, 2007.

[6] A. Byrski and M. Kisiel-Dorohinicki, "Immunological selection mechanism in agent-based evolutionary computation", in *Proc. Intel. Inform. Proces. and Web Mining IIS IIPWM'05: Gdańsk, Poland*, M. Klopotek, S. Wierzchon, and K. Trojanowski, Eds., Advances in Soft Computing. Springer, 2005.

[7] A. Byrski, R. Dreżewski, L. Siwik, and M. Kisiel-Dorohinicki, "Evolutionary multi-agent systems", *The Knowl. Engin. Rev.*, 2013 (accepted for printing).

[8] A. Byrski and R. Schaefer, "Stochastic model of evolutionary and immunological multi-agent systems: Mutually exclusive actions", *Fundamenta Informaticae*, vol. 95, no. 2–3, pp. 263–285, 2009.

[9] R. Schaefer, A. Byrski, and M. Smołka, "Stochastic model of evolutionary and immunological multi-agent systems: Parallel execution of local actions", *Fundamenta Informaticae*, vol. 95, no. 2–3, pp. 325–348, 2009.

[10] A. Byrski and R. Schaefer, "Formal model for agent-based asynchronous evolutionary computation", in *Proc. IEEE World Congr. Comput. Intel. 2009*, Trondheim, Norway, 2009. IEEE Computational Intelligence Society, IEEE Press.

[11] A. Byrski, R. Schaefer, and M. Smołka, "Asymptotic guarantee of success for multi-agent memetic systems", *Bull. Polish Acad. Sci. – Tech. Sci.*, no. 1, 2013.

[12] Z. Michalewicz, *Genetic Algorithms Plus Data Structures Equals Evolution Programs*. New York: Springer, 1994.

[13] W. Shortz, *The Little Black Book of Sudoku*. New York: St. Martin's Griffin, 2006.

[14] G. Civario, G. McGuire, and B. Tugemann, "There is no 16-clue sudoku: Solving the sudoku minimum number of clues problem", Cornell University Library, 2012.

[15] M. Ercsey-Ravasz and Z. Toroczkai, "The chaos within sudoku", *Scientific Reports*, no. 2, 2012.

[16] B. Hobiger, "Hodoku, a sudoku generator, solver, trainer and analyzer" [Online]. Available: http://hodoku.sourceforge.net/en/index.php

[17] A. Stuart, "Sudoku creation and grading", 2007 [Online]. Available: www.sudokuwiki.org

[18] A. Stuart, "Strategies for number puzzles of all kinds" [Online]. Available: www.sudoku.wiki.org

[19] A. Stuart, *The Logic of Sudoku*. UK: Michael Mepham Publishing, 2007.

[20] R. T. Rockafellar, "Nonsmooth analysis and parametric optimization", in *Methods of Nonconvex Analysis*, A. Cellina, Ed. *Lecture Notes in Mathematics*, vol. 1446, pp. 137–151. Berlin: Springer, 1990.

[21] M. Kisiel-Dorohinicki, "Agent-oriented model of simulated evolution", in *SofSem 2002: Theory and Practice of Informatics*, W. I. Grosky and F. Plasil, Eds. *LNCS*, vol. 2540. Springer, 2002.

[22] M. Luban and L. P. Staunton, "An efficient method for generating a uniform distribution of points within a hypersphere", *Comp. in Physics*, vol. 2, pp. 55–60, 1988.

[23] D. Krzywicki, "Niching in evolutionary multi-agent systems", *Comp. Sci.*, vol. 14, no. 1, 2013.

Mateusz Polnik obtained the B.Sc. in 2012 and is currently a student of a final year of M.Sc. studies at AGH University of Science and Technology in Kraków. He worked as a software developer at AGH-UST in tele-medicine project. His scientific interests include data distribution systems, relational databases and software architecture.

E-mail: polniks@gmail.com
Department of Computer Science
AGH University of Science and Technology
Mickiewicza Av. 30
30-059 Kraków, Poland

Mateusz Kumięga obtained the B.Sc. in 2012 and is currently a student of a final year of M.Sc. studies at AGH University of Science and Technology in Kraków. He is interested in computational intelligence and machine learning. His M.Sc. thesis concerns co evolutionary approach to computer chess.

E-mail: kumateus@student.agh.edu.pl
Department of Computer Science
AGH University of Science and Technology
Mickiewicza Av. 30
30-059 Kraków, Poland

Aleksander Byrski obtained his Ph.D. in 2007 at AGH University of Science and Technology in Kraków. He works as an assistant professor at the Department of Computer Science of AGH-UST. His research focuses on multi-agent systems, biologically-inspired computing and other soft computing methods.

E-mail: olekb@agh.edu.pl
Department of Computer Science
AGH University of Science and Technology
Mickiewicza Av. 30
30-059 Kraków, Poland