Agent-based Evolutionary Computing for Difficult Discrete Problems

Michał Kowol, Aleksander Byrski, and Marek Kisiel-Dorohinicki

AGH University of Science and Technology,
Al. Mickiewicza 30, 30-059 Kraków, Poland
michal.kowol@gmail.com, olekb@agh.edu.pl, doroh@agh.edu.pl

Abstract
Hybridizing agent-based paradigm with evolutionary computation can enhance the field of meta-heuristics in a significant way, giving to usually passive individuals autonomy and capabilities of perception and interaction with other ones, treating them as agents. In the paper as a follow-up to the previous research, an evolutionary multi-agent system (EMAS) is examined in difficult discrete benchmark problems. As a means for comparison, classical evolutionary algorithm (constructed along with Michalewicz model) implemented in island-model is used. The results encourage for further research regarding application of EMAS in discrete problem domain.

Keywords:

1 Introduction
In the paper, as a follow-up of the research presented in [20], a hybrid evolutionary-agent [19] approach to solving continuous optimisation problems is further researched. Generally in the state-of-the-art (see e.g., [25] or [11]) an evolutionary algorithm is used by an agent to aid realisation of some of its tasks, often connected with learning or reasoning, or to support co-ordination of some group (team) activity. Other approaches use the notion of an agent for constituting a management infrastructure for a distributed realisation of an evolutionary algorithm [26]. However, in order to utilize in full the features of evolutionary processes, that are decentralised by nature and indeed one may imagine the incorporation of evolutionary processes into a multi-agent system at a population level [18], making the agents not only managing entities for the solutions, but closely integrating them with these. So, besides interaction capabilities, agents may follow such actions as reproduction or removal from the system (death), based on predefined utility functions or resources they share, exchange and gather. In this way constructed evolutionary multi-agent system (EMAS) and its numerous variants have been successfully applied to different optimisation problems (e.g., single-criteria, multi-criteria, discrete, continuous) [9].

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2014

© The Authors. Published by Elsevier B.V.
doi:10.1016/j.procs.2014.05.093
This paper is devoted to a closer examination of the efficiency of EMAS in the problem of optimisation of difficult discrete problems, namely finding Low Autocorrelation Binary Sequences [17] and generation of Golomb Ruler [2]. Therefore, after recalling the basics of EMAS and highlighting the most important aspects of the problems tackled, the experimental setting of the examined systems are given and the outcomes of the experiments are described and discussed. The paper is finished with a concluding section.

2 Evolutionary agent-based optimization

Over the years evolutionary algorithms proved to be an effective universal technique for solving optimisation problems [4]. Instead of directly solving the given problem, subsequent populations of potential solutions are constructed, modelling phenomena of natural evolution. This process consists of two components: selection performed based on the solutions evaluation (fitness function), and generation of new solutions with the use of 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).

In contrast, the main components of evolutionary processes—inheritance and selection—are modelled in EMAS via agent actions of death and reproduction (see Fig. 1). Reproduction consists in the production of a new individual in cooperation with one of its neighbours that is chosen randomly, with the solution inherited from its parent(s) with the use of variation operators (mutation and recombination). Assuming no global knowledge available and the autonomy of agents, selection is based on the non-renewable resources [10].

In the simplest possible model of EMAS there is one type of agents and one resource defined (called life energy). 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 agents’ actions [18]:

- **Reproduction** – performed when the agent’s energy raises above a certain level, a part of energy (usually half of its initial value) is passed to a new agent 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 neighbours.

- **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 only if it meets an appropriate neighbour).

The topology of an island, defining the structure of inter-agent relations may be random (full graph of connections between the agents), but in order to enhance diversity of the population, an additional level of population decomposition (beside the evolutionary islands) may be introduced. Thus, a two-dimensional square lattice may be considered. In such lattice, different neighbourhoods (e.g., Moore’s) and boundary conditions (e.g., periodic, reflexive and fixed) may be utilised. In such an island, the agents may interact between themselves only providing they are in the zone of each other’s neighborhood.
3 Selected difficult discrete problems

In this section two problems (namely for LABS and Golomb Ruler) are described. These problems are difficult to solve, but quite easy to encompass in evolutionary environment, giving opportunity to explore in-depth search capabilities of the presented systems.

3.1 LABS

Low Autocorrelation Binary Sequence (LABS) is an NP-hard combinatorial problem with simple formulation. It has been under intensive study since 1960s by Physics and Artificial Intelligence communities. In consists in finding a binary sequence \( S = \{s_0, s_1, \ldots, s_{L-1}\} \) with
length $L$ where $s_i \in \{-1, 1\}$ which minimizes energy function $E(S)$:

$$C_k(S) = \sum_{i=0}^{L-k-1} s_is_{i+k} \quad E(S) = \sum_{k=1}^{L-1} C_k^2(S).$$

LABS has many applications in telecommunication (synchronization, pulse compression, satellite and space applications [15], digital signal processing, high-precision interplanetary radar measurements [31]), meteorology (calibration of surface profile meteorology tools [5]), physics (ising spin glasses, configuration state analysis, statistical mechanics [6]) and chemistry [30, 29, 6, 28, 12].

The search space for the problem with length $L$ has size $2^L$ and energy of sequence can be computed in time $O(L^2)$. LABS problem has no constrains, so $S$ can be represented naturally as an array of binary values [17]. In this problem all elements are correlated – there are no blocks of good solutions.

One of the reason of high complexity of problem is that in LABS all elements are correlated. One change that improves some $C_i(S)$, has also an impact on many other $C_j(S)$ and can lead to big changes of solution’s energy.

The second problem is that, LABS has only few global optima for most values of $L$ [14]. The search space is dominated by local optima. In [16] Halim compares search space to a “golf field”, where GO are deep, isolated and are spreader just like “golf holes”.

### 3.2 Golomb Ruler

A correct $n$-marks Golomb Ruler is an ordered, distinct, nonnegative, increasing sequence of integers $(g_1, g_2, \ldots, g_n)$ such that $g_i < g_{i+1}$ and all distances $g_j - g_i$ for $1 \leq i < j \leq n$ are unique. By convention the first element of the sequence $g_1$ equals 0 and the last element $a_n$ is the length of the ruler.

Golomb Ruler was formulated by professor Solomon Wolf Golomb in 1977 [7]. It has various applications in radio communication (signal interference), coding theory, X-ray crystallography and radio astronomy [1, 7, 21, 22, 13, 24].

Finding the shortest $n$-marks Golomb ruler is challenging NP-hard combinator problem. The search space is tremendous and it grows exponentially with number of marks [27]. This is the main obstacle to discovering new rulers because $(n+1)$-marks problem is much larger then $n$-marks problem.

### 4 Experiments

The results presented in this section were obtained using distributed, extensible, component computing environment AgE 2.7.0\(^1\). This platform is being developed as open-source project by the Intelligent Information Systems Group at AGH-UST. It allows to execute distributed agent-based simulation and computational tasks [8, 23]. A core of the platform was written in Java, and several components in Scala\(^2\) were recently added.

It is quite difficult to find a proper competitor for EMAS (as it is a one-of-kind system, directly hybridizing notions of agency with evolutionary computing), as The EA used for comparison followed Michalelewicz model [3] and was selected as a relatively similar, general-purpose

---

\(^1\)http://age.iisg.agh.edu.pl

\(^2\)http://www.scala-lang.org
optimisation algorithm, not utilising any agent-oriented features. The parameters of both systems were made as similar, as it was possible (e.g. tournament selection was used for EA as it is very similar to meeting mechanism present in EMAS). In the cases of variation operators (crossover and mutation) the were of course retained completely the same for both systems. The efficiency was measured with regard to time, instead of system steps (as it is usually difficult to compare EMAS and PEA in this way, because one system step means something completely different in these two systems: in EA one step processes the whole generation, while in EMAS—one agent).

In EA and EMAS, tournament selection with tourney size of 2 was used. For LABS problem, uniform recombination was used, since it provides a new promising starting point for the algorithm. In Golomb Ruler problem, one point crossover was applied. In EA population size, chance to mutation and recombination probability are 50, 0.75, 0.5 for LABS and 100, 0.2, 0.8 for Golomb, respectively. In EMAS agents’s count at the beginning, energy of reproduction, death and transfer are 50, 45, 0, 5 for both problems. Each experiment was repeated 30 times with the same parameters and standard deviation was presented in the graphs in addition to the actual results. Total execution time is 900 seconds for LABS and 300 seconds for Golomb, and in fact the time was marked on X-axis (see Figs. 2,3). In figures and tables mean values of fitness with standard deviations of the best-so-far solution are given. The fitness results were scaled with respect to the instances of the problems tackled, and 0 means the best know optima (minimisation assumed).
It took about 20 hours of computations on a workstation with Intel Core 2 Quad Q8300 2.5 Ghz (4 cores), 8 GB of DDR3 RAM, Windows 7 x64 to get results for LABS and Golomb Ruler problems presented in this paper.

For simple instance of LABS problem presented in 2a EMAS turned out to be significantly worse at the beginning and close (though a little bit worse) at the end of computation. On the other hand, for significantly harder problem 2b EMAS is worse only at the beginning of computation. After some time EMAS evidently prevails over EA, it seems that the latter becomes stuck in a local extremum, while EMAS is still capable of exploring the search space (see final results presented in Tab. 1).

Note that EMAS calls fitness function significantly rarer than EA (see Figs. 2c, 2d, where aggregated counts of fitness function calls were presented). Based on the final results shown in Tab. 1, it is easy to compute, that computing with EMAS requires 324 times less fitness function computation (for $L = 48$) and 22 times less ($L = 201$).

The results of experiments tackling Golomb Ruler problem show for both selected instances that EMAS becomes significant better that EA (see Figs. 3a, 3b for comparing the actual solutions obtained in the course of computation). EMAS still retains significantly lower cost of computing (in the means of fitness function count, see Figs. 3c, 3d). Again, final values obtained for both systems were shown in Tab. 1.

Note that regardless of the number of fitness function calls, in three of the four conducted experiments, EMAS obtained significantly better results faster than EA (as it may be seen in the graphs showing fitness value in relation to time of computation).

5 Conclusion

In this paper, following already obtained preliminary results in the field of discrete optimisation, using agent-based methods, new and more in-depth results were shown. The authors wanted to compare EMAS with as similar as possible not-agent base algorithm. It was quite hard to find proper competitor for EMAS, since it is unique, one-of-a-kind system. We considered several algorithms and we found PEA the most suitable for our need. We configured as much as possible parameters to maximize similarity between two systems. Current implementation of systems is sequential, but the work to parallelize platform are nearly finished and we are planning to carry out new experiments. EMAS turned out to be significantly better in the means of obtained final fitness in most of the conducted tests than EA. The additional important feature of EMAS was significantly lower computing cost, measured both using time counted from the beginning of computation, and number of fitness function calls. The advantages of the former feature is self-evident, while for the latter, it will be much more visible, when complex fitness functions are employed. Some of preliminary results applied to reverse problems were actually obtained (see [32]), however in the near future, continuation of this topic is envisaged.

Agent-based computing paradigm has already been studied, and supported by a number of scientific projects. One of such notable examples is ParaPhrase\(^3\), 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.

\(^{3}\)http://paraphrase-ict.eu
(a) EMAS and EA fitness for simple instance of Golomb problem (10–marks)
(b) EMAS and EA fitness for difficult instance of Golomb problem (14–marks)
(c) EMAS and EA fitness count for 10–marks Golomb ruler
(d) EMAS and EA fitness count for 14–marks Golomb ruler

Figure 3: EMAS and EA comparison for simple and difficult instance of Golomb Ruler problems

Table 1: Final result

(a) LABS $L = 48$

|          | EA       | EMAS     |
|----------|----------|----------|
| fitness  | $0.43 \pm 0.03$ | $0.44 \pm 0.04$ |
| fitness count | $4.2 \times 10^4 \pm 1.77 \times 10^4$ | $1.3 \times 10^4 \pm 4.4 \times 10^3$ |

(b) LABS $L = 201$

|          | EA       | EMAS     |
|----------|----------|----------|
| fitness  | $0.64 \pm 0.01$ | $0.45 \pm 0.02$ |
| fitness count | $2.72 \times 10^6 \pm 1.21 \times 10^6$ | $1.2 \times 10^5 \pm 2.93 \times 10^3$ |

(c) 10–marks Golomb Ruler

|          | EA       | EMAS     |
|----------|----------|----------|
| fitness  | $0.35 \pm 0.10$ | $0.20 \pm 0.05$ |
| fitness count | $8.47 \times 10^7 \pm 1.77 \times 10^7$ | $8.7 \times 10^4 \pm 3.58 \times 10^3$ |

(d) 14–marks Golomb Ruler

|          | EA       | EMAS     |
|----------|----------|----------|
| fitness  | $0.80 \pm 0.15$ | $0.53 \pm 0.05$ |
| fitness count | $6.79 \times 10^9 \pm 8.25 \times 10^9$ | $9.03 \times 10^4 \pm 1.28 \times 10^4$ |
Acknowledgment

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] Wallace C Babcock. Intermodulation interference in radio systems. *Bell Systems Technical Journal*, 32:63–73, 1953.
[2] W.C. Babcock. Intermodulation interference in radio systems/frequency of occurrence and control by channel selection. *Bell System Technical Journal*, 31:63–73, 1953.
[3] T. Bäck, D. Fogel, and Z. Michalewicz. *Vol. 1, Evolutionary Computation: Basic Algorithms and Operators, Vol. 2, Evolutionary Computation: Basic Algorithms and Operators Advanced Algorithms and Operators*. Institute of Physics Publishing, Bristol and Philadelphia, 2000.
[4] T. Back, U. Hammel, and H.-P. Schwefel. Evolutionary computation: Comments on the history and current state. *IEEE Trans. on Evolutionary Computation*, 1(1), 1997.
[5] Samuel K Barber, Paul Soldate, Erik H Anderson, Rossana Cambie, Wayne R McKinney, Peter Z Takacs, Dmytro L Voronov, and Valeriy V Yashchuk. Development of pseudorandom binary arrays for calibration of surface profile metrology tools. *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures*, 27(6):3213–3219, 2009.
[6] J Bernasconi. Low autocorrelation binary sequences: statistical mechanics and configuration space analysis. *Journal de Physique*, 48(4):559–567, 1987.
[7] Gary S Bloom and Solomon W Golomb. Applications of numbered undirected graphs. *Proceedings of the IEEE*, 65(4):562–570, 1977.
[8] A. Byrski and M. Kisiel-Dorohinicki. Agent-based model and computing environment facilitating the development of distributed computational intelligence systems. In *Computational Science - ICCS 2009, Proc. of 9th International Conference*, volume 5545 of LNCS. Springer-Verlag, 2009.
[9] Aleksander Byrski, Rafał Dreżewski, Leszek Siwik, and Marek Kisiel-Dorohinicki. Evolutionary multi-agent systems. *The Knowledge Engineering Review*, Accepted for publication, 2012.
[10] 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.
[11] Shu-Heng Chen, Yasushi Kambayashi, and Hiroshi Sato. *Multi-Agent Applications with Evolutionary Computation and Biologically Inspired Technologies*. IGI Global, 2011.
[12] Viviane M de Oliveira, José F Fontanari, and Peter F Stadler. Metastable states in short-ranged p-spin glasses. *Journal of Physics A: Mathematical and General*, 32(50):8793, 1999.
[13] R Gagliardi, J Robbins, and Herbert Taylor. Acquisition sequences in ppm communications (corresp.). *Information Theory, IEEE Transactions on*, 33(5):738–744, 1987.
[14] José E Gallardo, Carlos Cotta, and Antonio J Fernández. Finding low autocorrelation binary sequences with memetic algorithms. *Applied Soft Computing*, 9(4):1252–1262, 2009.
[15] R. Garelo, N. Boujniah, and Y. Jia. Design of binary sequences and matrices for space applications. In *Satellite and Space Communications, 2009. IWSSC 2009. International Workshop on*, pages 88–91, 2009.
[16] Steven Halim, Roland HC Yap, and Felix Halim. Engineering stochastic local search for the low autocorrelation binary sequence problem. In *Principles and Practice of Constraint Programming*, pages 640–645. Springer, 2008.
[17] Antonio J. Fernández Jos E. Gallardo, Carlos Cotta. Finding low autocorrelation binary sequences with memetic algorithms. *Applied Soft Computing*, 9, 2009.

[18] Marek 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, 2002.

[19] Marek Kisiel-Dorohinicki, Grzegorz Dobrowolski, and Edward Nawarecki. Agent populations as computational intelligence. In Leszek Rutkowski and Janusz Kacprzyk, editors, *Neural Networks and Soft Computing*, Advances in Soft Computing. Physica-Verlag, 2003.

[20] M. Kolybacz, M. Kowol, L. Lesniak, A. Byrski, and M. Kisiel-Dorohinicki. Efficiency of memetic and evolutionary computing in combinatorial optimisation. In *In. Proc. of 27th European Conference on Modelling and Simulation ECMS 2013 : May 27thMay 30th, 2013, Alesund, Norway*. 2013.

[21] GD Martin. Optimal convolutional self-orthogonal codes with an application to digital radio. In *ICC’85; International Conference on Communications*, volume 1, pages 1249–1253, 1985.

[22] Alan Moffet. Minimum-redundancy linear arrays. *Antennas and Propagation, IEEE Transactions on*, 16(2):172–175, 1968.

[23] K. Pietak, A. Woś, A. Byrski, and M. Kisiel-Dorohinicki. Functional integrity of multi-agent computational system supported by component-based implementation. In *Proc. of the 4th International Conference on Industrial Applications of Holonic and Multi-agent Systems*, volume 5696 of LNAI. Springer-Verlag, 2009.

[24] J Robinson and A Bernstein. A class of binary recurrent codes with limited error propagation. *Information Theory, IEEE Transactions on*, 13(1):106–113, 1967.

[25] Ruhul Sarker and Tapabrata Ray. *Agent-Based Evolutionary Search*. Springer, 2010.

[26] Robert Schaefer and Joanna Kołodziej. Genetic search reinforced by the population hierarchy. *Foundations of Genetic Algorithms*, 7, 2003.

[27] James B Shearer. Some new optimum golomb rulers. *Information Theory, IEEE Transactions on*, 36(1):183–184, 1990.

[28] Peter F Stadler. Landscapes and their correlation functions. *Journal of Mathematical chemistry*, 20(1):1–45, 1996.

[29] R Turyn. Sequences with small correlation. *Error correcting codes*, pages 195–228, 1968.

[30] R Turyn and J Storer. On binary sequences. *Proceedings of the American Mathematical Society*, 12(3):394–399, 1961.

[31] Abhishek Ukil. Low autocorrelation binary sequences: Number theory-based analysis for minimum energy level, barker codes. *Digital Signal Processing*, 20(2):483–495, 2010.

[32] Krzysztof Wróbel, Paweł Torba, Maciej Paszyński, and Aleksander Byrski. Evolutionary multi-agent computing in inverse problems. *Computer Science*, 14(3), 2013.