The Knowledge Engineering Review, Vol. 30:2, 171–186. © 2015, Cambridge University Press
DOI: 10.1017/S0269888914000289   Printed in the United Kingdom

Evolutionary Multi-Agent Systems

Aleksander Byrski, Rafał Dreżewski, Leszek Siwik, Marek Kisiel-Dorohinicki

AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Kraków, Poland
E-mail: {olekb,drezew,siwik,doroh}@agh.edu.pl

Abstract

The aim of this paper is to give a survey on the development and applications of evolutionary multi-agent systems (EMAS). The paper starts with a general introduction describing the background, structure and behaviour of EMAS. EMAS application to solving global optimisation problems is presented in the next section along with its modification targeted at lowering the computation costs by early removing certain agents based on immunological inspirations. Subsequent sections deal with the elitist variant of EMAS (elEMAS) aimed at solving multi-criteria optimisation problems, and the co-evolutionary one (CoEMAS) aimed at solving multi-modal optimisation problems. Each variation of EMAS is illustrated with selected experimental results.

1 Introduction

During the last decades intelligent/autonomous software agents have been gaining more and more applications in various domains, such as power systems management (McArthur, Catterson & Hatziargyriou 2007), flood forecasting (George, Gleizes, P.Glize & C.Regis 2003), business process management (Jennings, Faratin, Johnson, Norman, OBrien & Wiegand 1996), intersection management (Dresner & Stone 2008), or solving difficult optimization problems (Lobel, Ozdaglar & Feijer 2011) just to mention a few. The key to understand the concept of a multi-agent system (MAS) is an intelligent interaction (like coordination, cooperation, or negotiation). Thus multi-agent systems are ideally suited for representing problems that have many solving methods, involve many perspectives and/or may be solved by many entities (Wooldridge 2009). That is why, one of major application areas of multi-agent systems is large-scale computing (Uhruski, Grochowski & Schaefer 2008, Bouvry, González-Vélez & Kołodziej 2011).

Agents play an important role in the integration of AI sub-disciplines, which is often related to the hybrid design of modern intelligent systems (Russell & Norvig 2009). As the title suggests, the article deals with a hybrid evolutionary-agent approach. In most of similar applications reported in the literature (see e.g. Sarker & Ray (2010), Chen, Kambayashi & Sato (2011) for a review), 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 coordination of some group (team) activity. In other approaches, agents constitute a management infrastructure for a distributed realisation of an evolutionary algorithm (Schaefer & Kołodziej 2003).

Evolutionary processes are by nature decentralised and therefore one may imagine the incorporation of evolutionary processes into a multi-agent system at a population level. It means that agents are able to *reproduce* (generate new agents), which is a kind of cooperative interaction and may *die* (be eliminated from the system), which is the result of competition (selection). A similar idea with limited autonomy of agents located in fixed positions on some lattice (like in a cellular model of parallel evolutionary algorithms) was developed by e.g. Zhong, Liu, Xue & Jiao (2004). The key idea of the decentralised model of evolution employed by an *evolutionary multi-agent system*—EMAS (Kisiel-Dorohinicki 2002) was to ensure full autonomy of agents.

Such a system consists of a relatively large number of rather simple (reactive), often homogenous agents, which possess or produce solutions to the same problem (a common goal), it is rather closed and static, but non-determinitic (Moya & Tolk 2007). Because of both computational simplicity and a huge number of the agents, the influence of each single agent's behaviour on the overall system operation may be neglected, which allows for the efficient realisation in large-scale environments with lightweight infrastructure (Byrski, Dębski & Kisiel-Dorohinicki 2012).

The article aims to summarise research done for 15 years since the idea of EMAS was proposed by Cetnarowicz (1996). First, the background information and general structure of EMAS are presented at the beginning of the paper. Next, a basic model of EMAS and several of its variations (immunological, elitist, co-evolutionary) are described along with applications in different optimisation domains (global, multi-criteria and multi-modal). Each application is illustrated with selected experimental results recalled from the authors' publications. References show both the most important contribution of the authors and their collaborators in the field, as well as state-of-the-art that help to locate the presented work against the background of population-based metaheuristics.

2 Towards decentralised evolutionary computation

Evolutionary computation – a heuristic problem-solving approach based on models of organic evolution – has been successfully used for difficult optimisation problems for over 40 years (Fogel 1998, de Castro 2006, de Jong 2002). This approach is particularly useful when classical computational methods turn out to be ineffective, because corresponding models are too complex, or formulas applied too complicated, or even some formulations must be rejected in the face of numerical instability of available solvers. Unfortunately, there is no guarantee to find satisfactory solutions by evolutionary algorithms, and their performance aspects for particular problems must be verified empirically. This is simply because theoretical analyses make a number of assumptions in regard to the algorithm which limit their validity in real-world scenarios. Yet an evolutionary algorithm may often give an approximate solution with controllable adequacy, which means that a solving process can be stopped by a decision maker anytime he is satisfied.

Practice proves that an evolutionary algorithm works properly (e.g. in terms of searching for a globally optimal solution) if the population consists of fairly different individuals, i.e. the so-called diversity in the population is preserved (Bäck, Fogel & Michalewicz 1997). Yet many algorithms tend to prematurely loose this useful diversity and therefore the risk arises that the population might get stuck in some part of the search space (e.g. in the basin of attraction of a certain local extremum instead of searching for a global one). Loosing the population diversity also limits the possibilities of their application in some areas such as multi-objective optimisation or multi-modal optimisation.

The situation described above may be related to the fact that the model of evolution employed by simple evolutionary algorithms lacks many important features observed in organic evolution (Bäck, Hammel & Schwefel 1997). This includes dynamically changing environmental conditions, lack of global knowledge, no generational synchronisation, co-evolution of species, evolving genotype-phenotype mapping, etc. That is why many variations of classical evolutionary algorithms were proposed, introducing additional mechanisms following the most important phenomena in evolutionary biology. Among these, decomposition and co-evolutionary approaches had the most inspiring effect on the genesis of evolutionary multi-agent systems.

2.1 Decomposition and co-evolutionary techniques

Niching (or *speciation*) techniques (Mahfoud 1995) are aimed at introducing useful population diversity by forming subpopulations (species). *Allopatric* (or *geographic*) *speciation* occurs when individuals of the same species become isolated due to geographical or social changes. *Decomposition* approaches of the so-called *parallel evolutionary algorithms* (PEA) model such phenomena by introducing non-global selection/mating and some spatial structure of population (Cantú-Paz 1995).

In a *coarse-grained* PEA (also known as *regional* or *multiple deme* model), the population is divided into several subpopulations (regions, demes), selection/mating is limited to individuals inhabiting one region and a migration operator is used to move (copy) selected individuals from one region to another.

In a *fine-grained* PEA (also called a *cellular* model) individuals are located in some spatial structure (e.g. grid) and selection/mating is performed in the local neighbourhood (in this case one may imagine overlapping subpopulations).

In *co-evolutionary* algorithms the fitness of each individual is not computed directly based on the definition of the problem to be solved, but results from interactions with other individuals in the population. In *cooperative* co-evolutionary algorithms, a problem is decomposed into several subproblems solved by different algorithms in separate subpopulations (Potter & De Jong 2000). Cooperation between individuals from different subpopulations takes place only during a phase of computing fitness for the complete solution. The fitness value of the individual depends only on how well the selected group composed of representatives of all subpopulations (a complete solution), solves a given problem.

In *competitive* co-evolutionary algorithms usually two individuals compete against each other in a tournament and their "competitive fitness" corresponds to their performance in that competition (Paredis 1995). In each step of the algorithm, a given individual competes with different opponents and the results obtained have an impact on its current fitness. This mechanism can be applied irrespective of the number of populations used in the algorithm – it can be used even if there is only a single population. In this case, opponents are chosen from the same population. In the case of using several populations competitors are chosen from different subpopulations. Then, individuals mate with partners coming from the same subpopulation (species).

2.2 Agent-based model of evolution

In the approaches described above, the exact course of evolutionary processes depend on some context of a particular individual or subpopulation (e.g. the location in the population structure). The idea of decentralised model of evolution realised as an evolutionary multi-agent system may be considered as a step further in that direction.

Agents of EMAS represent or generate solutions for a given optimisation problem. They are located on islands (representing distributed structure of computation), which are their local environment, where direct interactions may take places. Obviously, agents are able to change their location, which allows diffusion of information and resources all over the system (Kisiel-Dorohinicki 2002).

In EMAS phenomena of inheritance and selection – the main components of evolutionary processes – are modelled via agent actions of *death* and *reproduction*. Inheritance is accomplished by an appropriate definition of reproduction, like in classical evolutionary algorithms. 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 determines the behaviour of an agent in the system (phenotype).

Assuming that no global knowledge is available (which makes it impossible to evaluate all individuals at the same time) and the autonomy of the agents (which makes reproduction be achieved asynchronously), selection is based on the non-renewable resources (Cetnarowicz, Kisiel-Dorohinicki & Nawarecki 1996). 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' behaviour, and looses resources as a consequence of 'bad' behaviour. Selection is then realised in such a way that agents with a lot of resources are more likely to reproduce, while the 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) (Franklyn & Graesser 1997).

To recapitulate: the main advantage of the approach under discussion is that it covers various specialised evolutionary techniques in one coherent model, which enables the following:

- local selection allows intensive exploration of the search space and explicitly defined living space facilitates implementation in a distributed computational environment, which is similar to parallel evolutionary algorithms,

- evaluation of agents, or more generally, the way a phenotype (behaviour of the agent) is developed from a genotype (inherited information) depends on its interaction with the environment, like in co-evolutionary algorithms.

3 EMAS in global optimisation

Assuming that some search space U is defined and a search criterion is given by function $f : U \to \mathbb{R}$ (often called *fitness* function in evolutionary computation), global optimisation task is usually defined as a search for a solution:

$$\vec{x}^{*} = arg \max_{\vec{x} \in D} f(\vec{x}),$$

where $D \subseteq U$ is a feasible set of solutions, and $arg\max$ stands for the argument of the function which gives its maximal value. For many techniques U must be a metric space with measure d defined, typically $U = \mathbb{R}^n,\ n \in \mathbb{N}$. Depending on the characteristics of D set, the optimisation task may be classified as continuous, discrete, combinatorial or mixed. Constrained optimisation may also be considered (Horst & Pardalos 1995). Moreover, it is easy to see that the problem of maximization may easily be changed into minimisation when needed, by negating the value of the function under consideration.

Solving optimisation problems with evolutionary algorithms requires choosing the encoding of solutions according to the search space definition, crossover and mutation operators appropriate for the encoding, as well as configuring a selection mechanism, or other components of specialised techniques like migration strategies for the island model of parallel evolutionary algorithms (Bäck, Fogel & Michalewicz 1997). The use of EMAS needs similar decisions concerning the encoding and variation operators, but the configuration of the system is different.

3.1 Basic model of EMAS

Figure 1(a) shows the simplest possible model of an evolutionary multi-agent system, with one type of agents and one resource (called *energy*) defined. Genotypes of agents represent feasible solutions to the problem. Energy is transferred between agents in the process of *evaluation*. When the agent finds out that one of its neighbours (e.g. randomly chosen), has lower fitness, it takes a part of its neighbour's energy, otherwise it passes part of its own energy to the evaluated neighbour. The level of life energy triggers the following actions:

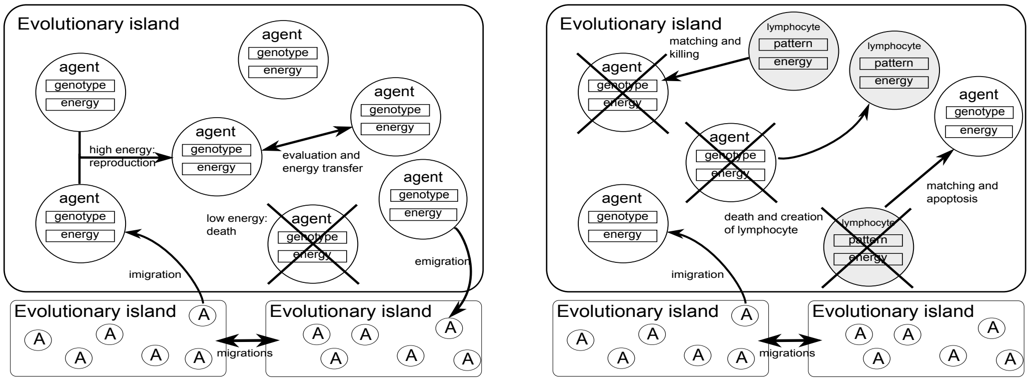
- *Reproduction* – performed when the agent's energy raises above a certain level, followed by production of a new individual in cooperation with one of its neighbours, with genotype based on parents' genotypes (crossed over and mutated) and part of energy also taken from its parents.
- *Death* – the agent is removed from the system when its energy falls below a certain level, the remaining energy is distributed among its neighbours.
- *Migration* – the agent may migrate when its energy rises above a certain level, then it is removed from one evolutionary island and moved to another according to predefined topology.

Each action is attempted randomly with a 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).

3.2 EMAS with immunological selection

The main idea of applying immunological inspirations to speeding up the process of selection in EMAS is based on the assumption that 'bad' phenotypes come from 'bad' genotypes. Immune-inspired approaches were applied to many problems, such as classification or optimisation (e.g., Dasgupta & Nino (2008)). The most frequently used algorithms of clonal and negative selection correspond to their origin and are used in a variety of applications (Wierzchoń 2002).

The general structure of immunological EMAS (iEMAS) is shown in Figure 1(b). A new group of agents (acting as lymphocyte T-cells) is introduced (Byrski & Kisiel-Dorohinicki 2007). They are responsible for recognising and removing agents with genotypes similar to the genotype patterns of these lymphocytes. Another approach may introduce specific penalty applied by T-cells for recognised agents (certain amount of the agent's energy is removed) instead of removing them from the system. Of course there must exist some predefined affinity (lymphocyte-agent matching) function which may be based, e.g. on the percentage difference between corresponding genes.



(a) Evolutionary multi-agent system

(b) Immunological evolutionary multi-agent system

Figure 1 Evolutionary and immunological types of EMAS

Agents-lymphocytes are created in the system after the action of death. The late agent genotype is transformed into lymphocyte patterns by means of mutation operator, and the newly created lymphocyte (or group of lymphocytes) is introduced into the system. In both cases, new lymphocytes must undergo the process of negative selection. In a specific period of time, the affinity of immature lymphocyte patterns with 'good' agents (possessing relatively high amount of energy) is tested. If it is high (lymphocytes recognise 'good' agents as 'non-self') they are removed from the system. If affinity is low, it is assumed that they will be able to recognise 'non-self' individuals ('bad' agents) leaving agents with high energy intact. The life span of lymphocytes is controlled by specific, renewable resource (strength) used as a counter by the lymphocyte agent.

Therefore, EMAS is enhanced by adding lymphocyte agents, altering the action of the agent's death and adding three lymphocyte-related actions:

- *Death* – EMAS action of death is redefined: during this action the agent produces one or more lymphocyte agents passing its mutated genotype to them and setting their strength to the maximum value.
- *Killing* – mature lymphocyte (with energy below a certain level) removes (or weakens) one of its neighbouring agents, if it finds that the genotype of this agent matches its own using predefined affinity function. Immature lymphocytes (with strength excessing a certain level) are checked whether they match an agent with high energy, in this case the lymphocyte is removed from the system,
- *Apoptosis* – lymphocyte with zero level of strength is removed from the system.
- *Give* – this action controls the negative selection process and overall lymphocyte acting time by simply decreasing the level of lymphocyte strength, allowing it to perform other actions (e.g., killing and apoptosis).

The concept of iEMAS is especially advantageous in applications requiring time-consuming fitness evaluation like the evolution of neural network architecture (Byrski & Kisiel-Dorohinicki 2007).

3.3 EMAS and iEMAS optimisation results

Experiments concerning minimisation of the benchmark functions presented below were reported by (Byrski & Kisiel-Dorohinicki 2007). EMAS, iEMAS and a classical parallel evolutionary algorithm were checked against popular benchmarks (Digalakis & Margaritis 2002) in order to test their efficiency

(10-dimensional functions of Ackley, De Jong, Griewank and Rastrigin). Variation operators of discrete crossover and uniform mutation were used. In PEA tournament selection (being the most similar selection mechanism to energetic selection principle in EMAS) was used. The systems consisted of 3 evolutionary islands with 30 agents (or individuals in PEA) in the initial configuration.

Best fitness values in consecutive steps of the system's work obtained for all systems for Rastrigin benchmark are presented in Figure 2. The experiments were repeated 10 times and the average together with standard deviation was shown. It may be seen that, agent-based systems outperformed the classical one (PEA) in the means of getting close to the desired solution.

It must also be noted that iEMAS was a little worse than EMAS, yet other advantages of this system will follow. Final results obtained for other benchmarks are presented in Table 1 (standard deviation denoted by ± sign).

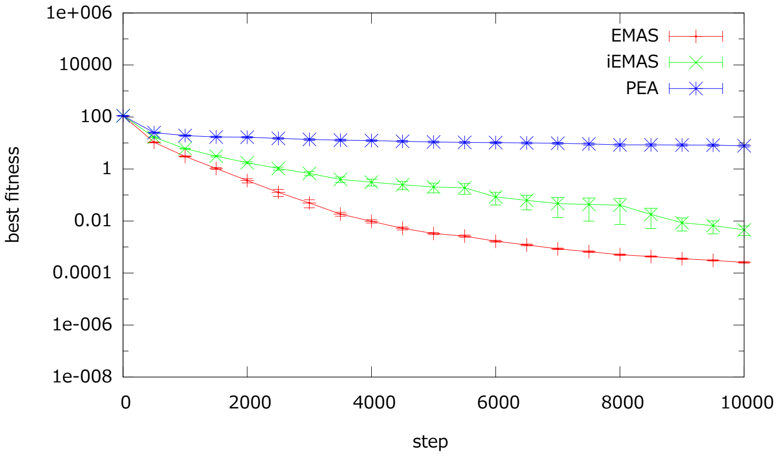


Figure 2 Best fitness for EMAS, iEMAS and PEA for Rastrigin benchmark

Table 1 Best fitness in 10000th step of the system's work for EMAS, iEMAS and PEA

Benchmark	PEA		EMAS		iEMAS	
Ackley	1.97	±0,07	$1{,}38 \cdot 10^{-3}$	$\pm 3{,}7 \cdot 10^{-5}$	$4{,}17 \cdot 10^{-3}$	$\pm 1{,}40 \cdot 10^{-4}$
De Jong	$6{,}49 \cdot 10^{-6}$	$\pm 9{,}60 \cdot 10^{-7}$	$1{,}31 \cdot 10^{-6}$	$\pm 6{,}86 \cdot 10^{-8}$	$7{,}9 \cdot 10^{-6}$	$\pm 5{,}95 \cdot 10^{-7}$
Griewank	$7{,}58 \cdot 10^{-3}$	$\pm 2{,}33 \cdot 10^{-3}$	0.02	$\pm 5{,}05 \cdot 10^{-3}$	0.11	±0,03
Rastrigin	7.81	±0,81	$2{,}58 \cdot 10^{-4}$	$\pm 1{,}68 \cdot 10^{-5}$	$4{,}58 \cdot 10^{-3}$	$\pm 1{,}83 \cdot 10^{-3}$

Table 2 Fitness count in 10000th step of system's work for EMAS, iEMAS and PEA

Benchmark	PEA	EMAS		iEMAS	
Ackley	$6 \cdot 10^5$	57 840	±43	25 596	±149
De Jong	$6 \cdot 10^5$	57 769	±43	26 172	±158
Griewank	$6 \cdot 10^5$	58 484	±62	30 829	±379
Rastrigin	$6 \cdot 10^5$	57 787	±29	27 727	±220

The fitness count for the presented optimisations shows that agent-based systems EMAS and iEMAS are far more effective than PEA, since they need far fewer computations of fitness functions to obtain similar or even better results than PEA. The efficiency results for all benchmarks are presented in Table 2.

The experiments demonstrated that though the accuracy in localising the optimum by EMAS and iEMAS in comparison to the classical PEA was similar, the efficiency was much greater (the number of fitness function calls was even one order of magnitude higher in PEA than in EMAS and iEMAS). Although EMAS' complexity can become an important factor of the computing power consumption (cf. Wolpert & Macready (1997)), this technique may still be used as effective when dealing with problems where complex fitness function is evaluated.

4 EMAS in multi-criteria optimisation

When a lot of different (often contradictory) factors have to be considered, the decision-maker deals with an ambiguous situation: alternatives that are optimal from the point of view of one criterion may turn out to be absolutely unacceptable considering the others. This situation is reflected by the notion of *Pareto optimality*. Assuming that alternatives are evaluated by the system of M functions (objectives) denoted here by the following vector $F = [f_1, f_2, \ldots, f_M]^T$, where: $f_m : U \to \mathbb{R}$, $m = 1, 2, \ldots, M$ (cf. Section 3), alternative $\vec{x}^a$ is said to be dominated by $\vec{x}^b$ (which is often denoted by $\vec{x}^b \succeq \vec{x}^a$) if and only if (assuming maximisation of all the objectives):

$$\forall m \;\; f_m(\vec{x}^a) \leq f_m(\vec{x}^b) \text{ and } \exists n \in \{1, 2, \ldots, M\} \; : \; f_n(\vec{x}^a) < f_n(\vec{x}^b)$$

The goal of solving multi-criteria optimisation problem is to find all non dominated alternatives $\vec{x}^*$ in a feasible solution set D which constitutes the so-called Pareto set (and in objective space the so-called Pareto frontier).

For the last 20 years a variety of evolutionary multi-criteria optimisation techniques have been proposed – see Coello Coello, Lamont & Van Veldhuizen (2007), Deb (2001), Veldhuizen & Lamont (2000), Fonseca & Fleming (1995) for a review. Deb's typology of evolutionary multi-objective algorithms (EMOAs) distinguishes elitist and non-elitist algorithms for the first time (Deb 2001). The main difference between these two groups of techniques consists in utilizing the so-called elite-preserving operators that give the best individuals (the elite of the population) an opportunity to be directly carried over to the next generation, regardless of the actual selection mechanism used. Of course, if the algorithm finds a better solution than the one in the elite, this solution becomes a new elitist solution. Unfortunately, in the case of multi-criteria optimisation, since there are multiple objective functions, it is not as straightforward as in the single-objective case to identify the elite. In such situations, the non-dominating ranking comes to our rescue (Deb 2001, p.240).

During research to date, many different realisations of EMAS-based multi-objective optimisation problems solvers have been proposed, e.g. elitist multi-objective EMAS (e.g. Siwik & Natanek (2008)), EMAS with cultural and immunological mechanisms (Siwik & Dreżewski 2009) and the whole set of coevolutionary EMASs (Dreżewski & Siwik 2010). In the next Section, elitist evolutionary multi-agent system (elEMAS) will be briefly described as a good illustration of multi-objective EMAS.

4.1 Elitist evolutionary multi agent system for multi-objective optimisation

Elitism can be introduced into EMAS in many different ways. In the case that is considered, it is based on a slightly modified structure of environment (see Figure 3). Compared to the structure of environment shown in Figure 1(a), modifications consist in introducing an additional, the so-called elitist island and special actions that can be performed (only) by selected agents, allowing them to migrate to this very island. The elitist island only accepts immigrant agents, there is no outbound migration from there. Thus agents which have decided to migrate to this island are not able to go back from elitist to ordinary island(s), and cannot take part in the process of evolution.

A mechanism allowing to identify elitist agents is based on the level of additional resource called *prestige*, which is gathered (and is not lost) by agents during their life. At the beginning of their life, the level of *prestige* equals *zero*. Then, every time an agent dominates (in a sense of *domination relation*) any other agent, its level of *prestige* increases, and so it may be assumed that agents with a high level of this specific resource belong to the elite. This mechanism allows the realisation of the above-mentioned idea of *non-dominating ranking* in a really elegant, easy to understand and implement way that does not require any additional complicated operations and computations.

It turns out that applying elitist operator(s) to an evolutionary multi-agent system – despite other presented advantages – gives an opportunity to introduce additional mechanism(s) responsible for

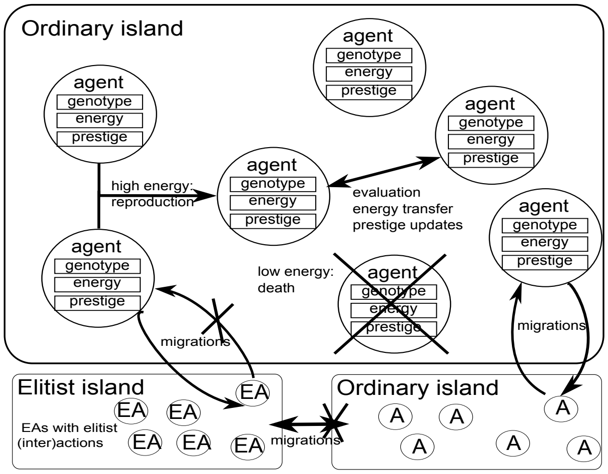


Figure 3 Elitist evolutionary multi-agent system

improving the spreading of the individuals over the Pareto frontier. One of such mechanisms can be realized as follows: during meetings, an agent is able to check if a solution represented by its opponent is located in its surroundings at the frontier (i.e. if the distance between solutions represented by agents is less than a given value ε). If so, the agent increases its personal counter storing the number of individuals located in the same fragment of the Pareto frontier. Additionally, asking its opponents about the number of individuals located in their surroundings, the agent is able to gather with time (partial) knowledge of the average number of individuals located in some areas of the Pareto frontier (i.e. in surrounding areas represented by agents it met). Consequently, if the agent becomes the elitist one, it can make a decision about migrating to the elitist island only when the number of individuals located in its surroundings is greater than the average number of individuals located in other regions of the Pareto frontier.

The mechanism described allows deeper exploration of those areas of the Pareto frontier which are worse sampled than its other regions. A similar mechanism can also be applied to the space of decision variables, thus concerning the Pareto set. Of course both mechanisms can be used simultaneously, and this can be of great importance in real-life problems where both, Pareto set and Pareto frontier are often disconnected.

Additionally, ε parameter can be managed adaptively by agents. In the simplest form, it can be linearly smaller and smaller with the agent's lifetime. In more advanced implementations it can be decreased by agents on the basis of their interactions with the environment and other agents.

4.2 Multi-criteria optimisation in noisy environment

Considering advantages of EMAS-based optimisation solvers over classical evolutionary algorithms, the so-called *soft selection* has to be underlined. Unlike classical algorithms, if an individual is dominated by another one, it is not immediately removed from the population but its energy level is decreased (depending on parameter values) so it has to be dominated several times to be eventually removed from the population.

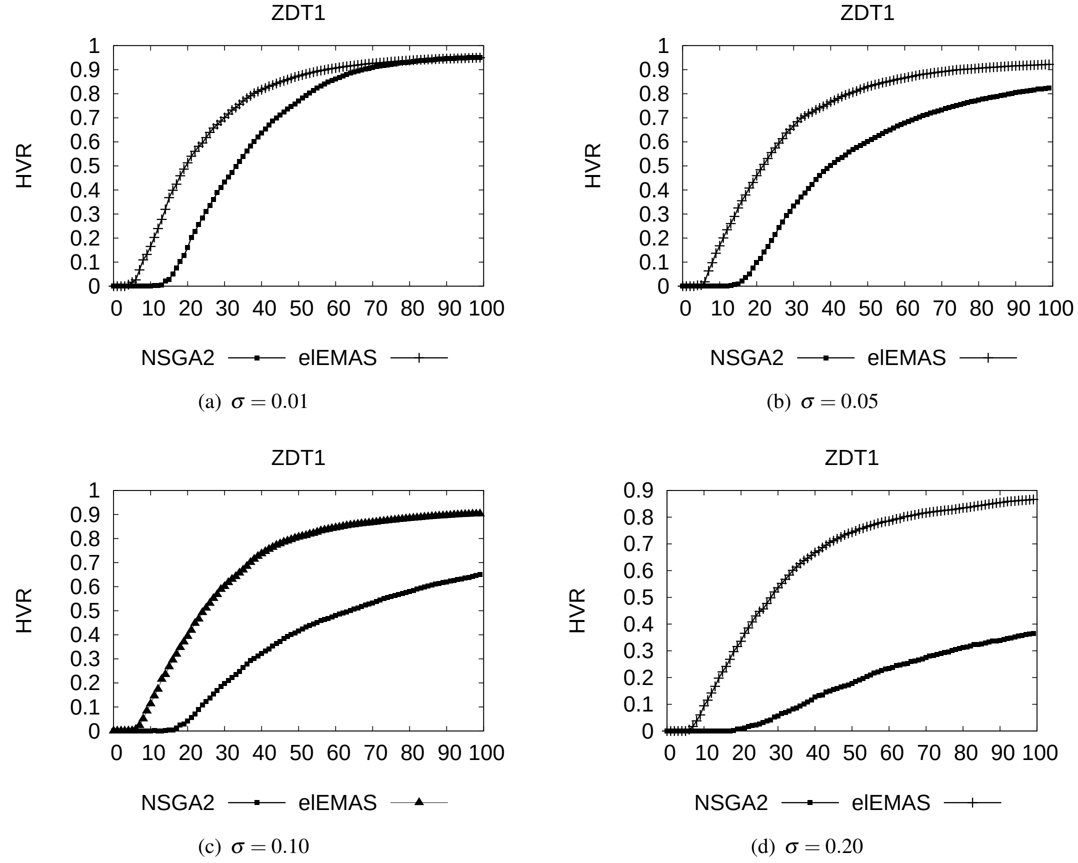
Such a mechanism can be especially important if optimisation in a noisy environment is considered. Unlike mathematical models, in real-life problems noise cannot be avoided and it may come from many different sources, e.g. distortions in input data, wrong sampling, as well as approximation faults. In such an environment, worse solutions may be falsely perceived by the algorithm as better ones. Consequently, worse solutions may be carried over to the next generations. It depends of course on the 'scale of noise' but it is possible that in this case the algorithm will drift towards the misleading 'optimal' result set.

Considering noise at the level of fitness function evaluation, this noisy fitness function takes the form of $F_{noise} = F + \xi$, where F is fitness function without noise influence, and ξ is the function of noise (Bui, Essam, Abbas & Green 2004). To simulate noise, normal or uniform distribution can be used. Noise used for experiments was generated with Gaussian distribution $N(\mu,\sigma)$, where $\mu = 0$ and σ varies from 0 to 0.20 (which represents different levels of noise). Since the magnitude of objective functions for the Pareto frontier is between $[0,1]$, the amount of noise (especially for value 0.20) is significant. Such a range of noise allows to present the influence of noise on the quality of the result sets obtained, and consequently present the quality of algorithm.

In order to compare the performance of Non-dominated Sorting Genetic Algorithm 2 (NSGA2) (as a commonly used benchmark) and elEMAS, all common parameters are set equally especially population size and parameters of variation operators. Values of the most important ones are presented in Table 3. Since hypervolume (HV) or hypervolume ratio (HVR) (Van Veldhuizen 1999) allows to estimate both: the convergence to as well as distribution of solutions over the whole approximation of the Pareto frontier, it is one of the most commonly and most frequently used measure as the main metric for comparing quality of the result sets obtained.

Table 3 Common parameters for NSGA2 and elEMAS

Parameter	ZDT1-3	ZDT4
Probability of crossover, p_c	0.9	0.9
Probability of mutation, p_m	0.033	0.1
Crossover η	10	15
Mutation η	10	20



(a) $\sigma = 0.01$

(b) $\sigma = 0.05$

(c) $\sigma = 0.10$

(d) $\sigma = 0.20$

Figure 4 Comparison of HVR(t) values with different values of σ obtained for solving ZDT1 problem

Due to space limitations, only some exemplary but demonstrative results are presented here. As can be seen on the diagrams in Figure 4 showing values of HVR metric for Zitzler ZDT1 problem (Zitzler 1999), although NSGA2 algorithm is commonly perceived as one of the most effective and most commonly used evolutionary algorithm for multi-objective optimisation, however, elitist evolutionary multi-agent system seems to be certainly a better alternative in a noisy environment. What is more, the higher noise is, the worse the results obtained by NSGA2 are. Such a characteristic is not observed in the case of EMAS-based algorithms. So it can be said that a 'soft' selection mechanism allows to avoid mistakes resulting from noise. For a relatively large amount of noise, elEMAS unlike NSGA2 manages to converge to the Pareto frontier.

5 EMAS in multi-modal optimisation

The goal of multi-modal optimisation is to find all local maxima of the $f(\cdot)$ function in a feasible solution set D (cf. Section 3). It means that all points $\vec{x}^* \in D$ are sought that the following condition is true (Bäck & Schwefel 1996):

$$\exists \varepsilon > 0 \quad \forall \vec{x} \in D \quad d(\vec{x}, \vec{x}^*) < \varepsilon \Rightarrow f(\vec{x}) \leqslant f(\vec{x}^*) \tag{1}$$

The goal of evolutionary multi-modal optimisation techniques (niching techniques) is to find all (or most) of these solutions (Mahfoud 1995).

In classical evolutionary algorithms, forming and stable maintaining subpopulations (species) are achieved by applying *niching* techniques like allopatric speciation based on the restricted application of selection and/or recombination mechanisms, as described in Section 2. Alternatively in *sympatric* speciation new species evolve while inhabiting the same geographic region. It is modelled in evolutionary algorithms by techniques based on the modification of the parent selection mechanism like *fitness sharing*, or techniques based on the modification of the mechanism of selecting individuals for a new generation like *deterministic crowding* (Mahfoud 1992). Ideally, each of the species should be placed within one of the basins of attraction of local maxima, which in practice might be difficult to achieve.

Although co-evolutionary techniques and *sexual selection* mechanism (Sánchez-Velazco & Bullinaria 2003) promote the population diversity, they were not widely used as a basis for constructing niching techniques. These techniques were a point of reference for the model of co-evolutionary multi-agent system (CoEMAS) (Dreżewski 2003).

5.1 Co-evolutionary multi-agent systems

As in the case of classical evolutionary algorithms, basic EMAS-based systems, without any special mechanisms that promote useful population diversity are unable to solve multi-modal optimisation problems because they are not able to form species located within basins of attraction of many different local optima. They would rather locate only one solution, however, population diversity would be much higher than in the case of classical evolutionary algorithm.

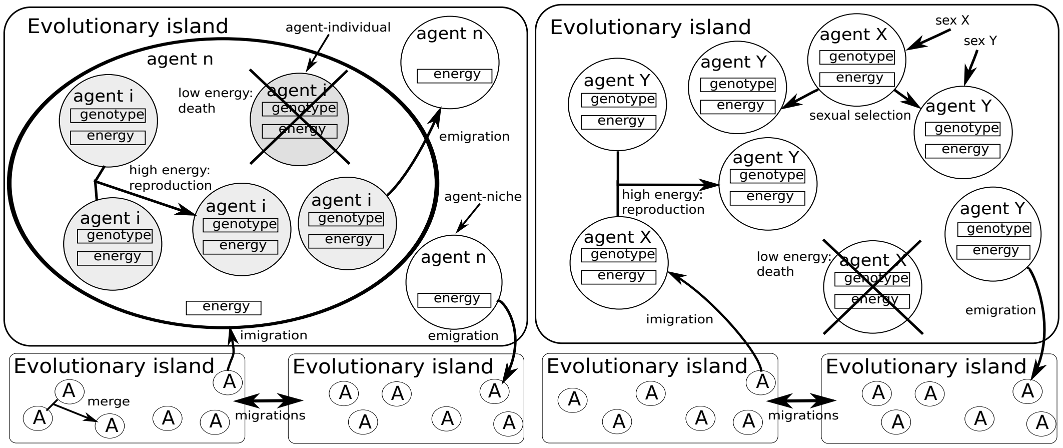
The basic idea behind the CoEMAS model is to introduce the possibility of co-evolution of species and co-evolution of sexes (sexual selection) within EMAS. In the exemplary systems presented in the next sections, co-evolutionary interactions between species and sexual selection are used in order to construct niching techniques in evolutionary multi-agent system and to promote useful population diversity.

In the **co-evolutionary multi-agent system with co-evolving species** (nCoEMAS) there exist two different types of computational agents. They will be called 'niches' and 'individuals' (*agent n* and *agent i* in Figure 5(a)). Individuals 'live' within niches; they can search for a suitable agent-niche and migrate into it. They can even 'create' new agent-niche when they are not able to find a suitable niche and migrate into it. The main goal of the mechanism of migrating between and creating new niches is to split the population of agents-individuals into such subpopulations (species) so that each of them would be located within a distinct basin of attraction of local maxima.

Mating is restricted only to agents-individuals located within the same agent-niche. Reproduction, as in the case of basic EMAS model, is initiated by an agent that has enough resources to reproduce. The agent initiating a reproduction process searches for other ready for reproduction partners from the same niche and chooses one of them. Two children are generated and each offspring receives some amount of resources from each of their parents.

Agents-niches can migrate within an environment, but only when they have enough resources, because each migration costs some amount of resources which is returned to the environment. Two agents-niches can merge with each other: merging takes place when centers of gravity of sub-populations, which belong to two different niches, are located within the basin of attraction of the same local optima. The system is applied to multi-modal optimisation problems, therefore, the goal is to localize most of basins of attraction of local maxima.

Resources circulate between an environment and agents. The whole amount of resources within the system is constant so the number of agents cannot grow infinitely. Environment gives equal amount of resources to each agent-niche, and then they redistribute resources among agents-individuals proportionally to their fitness.



(a) CoEMAS with co-evolving species

(b) CoEMAS with sexual selection

Figure 5 Co-evolutionary multi-agent systems

Multi-agent system with sexual selection (sCoEMAS) utilizes co-evolution of sexes and sexual selection in order to obtain the effect of speciation. In sCoEMAS there exist two sexes: female and male (*agent X* and *agent Y* in Figure 5(b)).

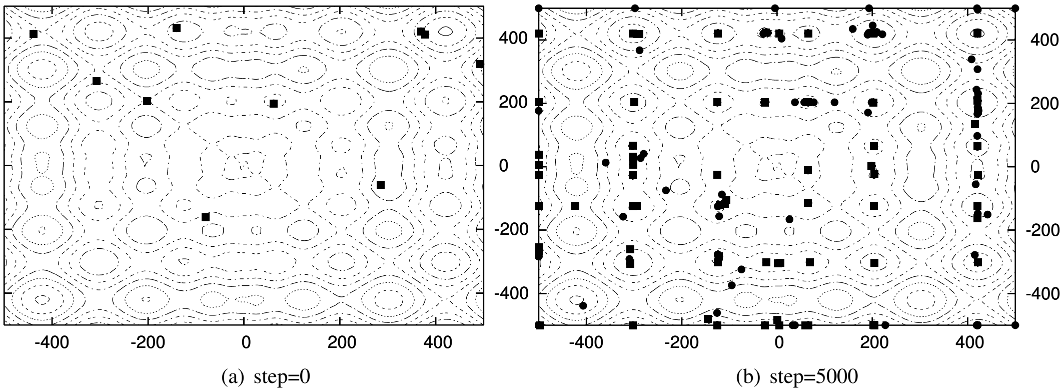
Reproduction process is initiated by a female agent when it has enough resources to reproduce. Generally, female agents use more resources than male agents during the reproduction process and that is why they reproduce less frequently than male agents. Therefore, there are always more male agents ready for reproduction than female agents and female agents choose partners from several ready for reproduction male agents. A female agent located within an evolutionary island searches for a partner in such a way that it chooses one ready for reproduction male agent from the same island. The partner is chosen on the basis of genotype similarity. The more similar two agents from opposite sex are, the more probable is that the female agent will choose the male agent. Reproduction takes place when a pair composed of agents of the opposite sex is formed. Such a pair of agents is formed permanently for several steps of the algorithm because of difficulties and costs involved in searching for a partner. The offspring (two children) is generated with the use of mutation and recombination operators and receives some of the resources from parents: more from the female agent and less from the male one.

Agents can migrate within an environment but it costs some amount of resources. Similar to nCoEMAS, in sCoEMAS the whole amount of resources present within it is constant and the environment distributes resources between agents proportionally to their fitness.

5.2 Multi-modal optimisation results

In this Section, selected results of experiments with nCoEMAS and sCoEMAS applied for function minimisation are presented (the results were reported in (Dreżewski 2006) and (Dreżewski & Cetnarowicz 2007)). The results of co-evolutionary systems are compared to those obtained by classical niching techniques: deterministic crowding (DC) and fitness sharing (FS) mentioned earlier and the results obtained by the basic EMAS. During experiments, Rastrigin and Schwefel multi-modal test functions were used.

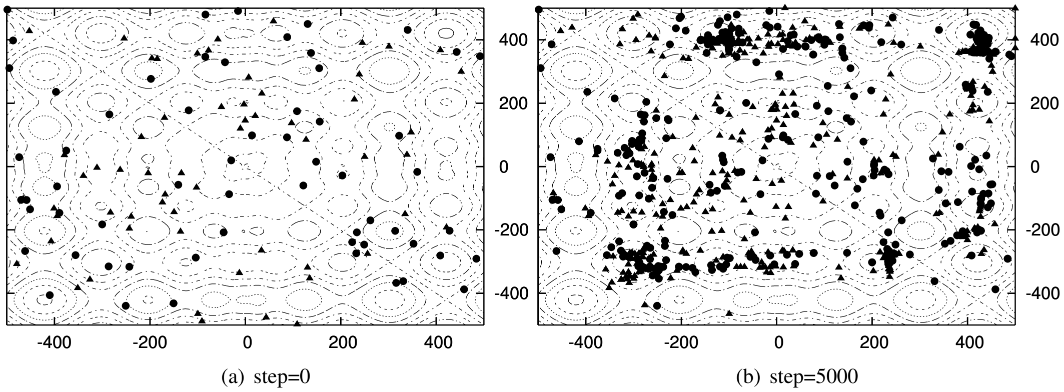
The results presented in Figures. 6 and 7 show the location of individuals within the search space during typical experiments with nCoEMAS and sCoEMAS and Schwefel test function. It can be seen that nCoEMAS forms very focused species, and individuals are located within the closest neighbourhood of local minima. The population diversity within species is not very high. Species are stable: once formed they do not disappear during the experiment. In the case of sCoEMAS, they are also formed and stably maintained during the experiment. However, diversity within species is much higher: agents are dispersed over the whole basins of attraction of local minima.



(a) step=0

(b) step=5000

Figure 6 Species formation processes in nCoEMAS during a typical experiment with Schwefel test function



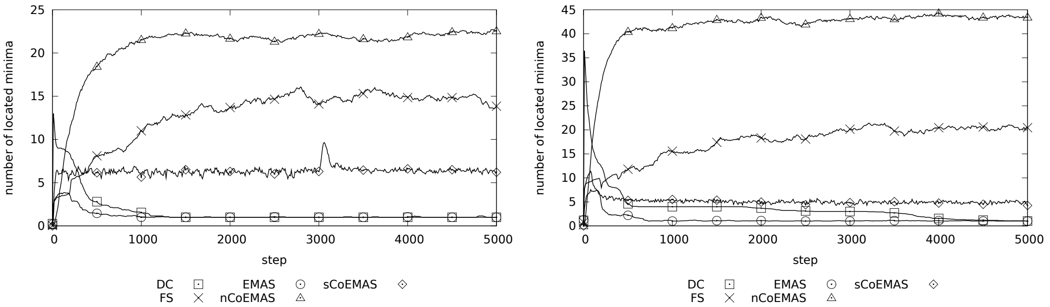
(a) step=0

(b) step=5000

Figure 7 Species formation processes in sCoEMAS during a typical experiment with Schwefel test function

Figure 8 shows the average number (from 20 experiments) of local minima neighbourhoods located. The local minima neighborhood was classified as 'located' when there was at least $ni_{min} = 3$ individuals closer than 0.05 (in the case of Rastrigin function) and 10.0 (in the case of Schwefel function) to these local minima.

nCoEMAS obtained the best results: on average, it located about 22 local minima neighborhoods of Rastrigin function, and over 40 local minima neighborhoods of Schwefel function. DC algorithm quickly located about 13 local minima neighborhoods of Rastrigin function and about 36 local minima neighborhoods of Schwefel function, but there was quite a strong tendency to lose almost all of them during the rest part of experiments. Species were not stably maintained in this case. FS technique detected and stably maintained about 14 local minima neighbourhoods of Rastrigin function and about 20 local minima neighbourhoods of Schwefel function on average. sCoEMAS located about 6 local minima neighbourhoods of Rastrigin function and 5 local minima neighbourhoods of Schwefel function. The results point to high population diversity. Agents are spread over the whole basins of attraction and they are not concentrated in the closest neighborhoods of local minima. EMAS without any niching mechanism was not able to stably populate more than one local minima neighborhood. However, what is not shown in Figure 8, it maintained high population diversity, but the whole population was located within the basin of attraction of only one minimum.



(a) Rastrigin (b) Schwefel

Figure 8 Average number of located minima of test functions by DC, FS, EMAS, nCoEMAS and sCoEMAS

6 Conclusion

The paper presented the development of agent-based evolutionary computation paradigm realised as an evolutionary multi-agent system (EMAS). For over 15 years since it was proposed for the first time, the paradigm itself as well as its extensions and modifications has proved to be an efficient tool in different applications in global, multi-modal and multi-criteria optimisation. Various extensions, such as immunological, elitist and co-evolutionary ones have been proposed and described. Over 80 papers, four Ph.D. and many M.Sc dissertations have been prepared and published so far as well. A large-scale infrastructure consisting of several clusters and a cloud-oriented virtualisation server has been prepared and is utilised in computing experiments.

Due to the space limit, many aspects of evolutionary multi-agent systems could not be discussed in details here. This includes self-sustainability that may be applied to maintaining stability of the system (with respect to an appropriate size of the population of agents), the capability to respond to the emergency situations occurring in the hardware (breakdown of computing nodes), as well as in the user requirements (e.g. change of computation parameters) (Byrski, Kisiel-Dorohinicki & Carvalho 2010). EMAS was also applied as a base for hybrid soft computing solvers for classification and prediction problems, with neural networks (Byrski & Kisiel-Dorohinicki 2007), fuzzy systems (Fusinska, Kisiel-Dorohinicki & Nawarecki 2007), pattern recognition (Byrski, Dobrowolski & Toboła 2008) and others.

EMAS-based approaches inevitably shares some of the flaws and limitations of evolutionary algorithms (cf. Section 2) including difficulties in their configuration and formal modelling. Some approaches developed in theoretical analysis of evolutionary algorithms (see, e.g. works of Vose or Rudolph) have been applied to EMAS – several formal models have been designed, aimed at describing the paradigm and analysing of its asymptotic features (see, e.g. Schaefer, Byrski & Smołka (2009)). Unfortunately, they take into consideration only particular cases of algorithms. As in the case of traditional (non agent-based) evolutionary algorithms since there are no accurate, general models – formal analysis of their behaviour or conclusions about their effective configurations are really limited.

Research on EMAS continues, and its main directions that should be mentioned here are the continuation of formal modelling focused on researching different asymptotic features, as well as building infrastructure and tools supporting the implementation of various systems. Applying EMAS for solving optimization problems in many different areas with varying features and characteristics makes it necessary to utilise flexible configuration mechanisms, which drives further development of distributed software frameworks, that will allow the construction of agent-based computing systems consisting of reusable components (Piętak, Woś, Byrski & Kisiel-Dorohinicki 2009).

Acknowledgement

The work presented in the 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" and AGH University of Science and Technology statutory fund.

References

Bäck, T., Fogel, D. & Michalewicz, Z., eds (1997), *Handbook of Evolutionary Computation*, IOP Publishing and Oxford University Press.

Bäck, T., Hammel, U. & Schwefel, H.-P. (1997), 'Evolutionary computation: Comments on the history and current state', *IEEE Trans. on Evolutionary Computation* **1(1)**, 3–17.

Bäck, T. & Schwefel, H.-P. (1996), Evolutionary computation: An overview, *in* T. Fukuda & T. Furuhashi, eds, 'Proceedings of the Third IEEE Conference on Evolutionary Computation', IEEE Press, pp. 20–29.

Bouvry, P., González-Vélez, H. & Kołodziej, J. (2011), *Intelligent Decision Systems in Large-Scale Distributed Environments*, Springer.

Bui, L. T., Essam, D., Abbas, H. A. & Green, D. (2004), Performance analysis of evolutionary multiobjective optimization methods in noisy environments, *in* '8th Asia Pacific Symposium on Intelligent and Evolutionary Systems', Monash University, Melbourne, Australia.

Byrski, A., Dębski, R. & Kisiel-Dorohinicki, M. (2012), 'Agent-based computing in an augmented cloud environment', *Computer Systems Science and Engineering* **27**(1), 5–20.

Byrski, A., Dobrowolski, J. & Toboła, K. (2008), Agent-based optimization of neural classifiers, *in* 'Conference on Evolutionary computation and global optimization 2008 : June 2nd4th, 2008, Szymbark, Poland'.

Byrski, A. & Kisiel-Dorohinicki, M. (2007), Agent-based evolutionary and immunological optimization, *in* 'Computational Science - ICCS 2007, 7th International Conference, Beijing, China, May 27 - 30, 2007, Proceedings', Springer.

Byrski, A., Kisiel-Dorohinicki, M. & Carvalho, M. (2010), 'A crisis management approach to mission survivability in computational multi-agent systems', *Computer Science* **11**, 99–113.

Cantú-Paz, E. (1995), 'A summary of research on parallel genetic algorithms', *IlliGAL Report No. 95007. University of Illinois* .

Cetnarowicz, K. (1996), Evolution in multi-agent world = genetic algorithms + aggregation + escape, *in* '7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'96)', Vrije Universiteit Brussel, Artificial Intelligence Laboratory.

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

Chen, S.-H., Kambayashi, Y. & Sato, H. (2011), *Multi-Agent Applications with Evolutionary Computation and Biologically Inspired Technologies*, IGI Global.

Coello Coello, C. A., Lamont, G. B. & Van Veldhuizen, D. A. (2007), *Evolutionary Algorithms for Solving Multi-Objective Problems, Second edition*, Kluwer Academic Publishers.

Dasgupta, D. & Nino, L. (2008), *Immunological Computation Theory and Applications*, Auerbach.

de Castro, L. N. (2006), *Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications*, Chapman and Hall/CRC Computer and Information Science Series.

de Jong, K. (2002), *Evolutionary Computation*, A Bradford Book.

Deb, K. (2001), *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons.

Digalakis, J. & Margaritis, K. (2002), 'An experimental study of benchmarking functions for evolutionary algorithms', *International Journal of Computer Mathemathics* **79**(4), 403–416.

Dresner, K. & Stone, P. (2008), 'A multiagent approach to autonomous intersection management', *Journal of Artificial Intelligence Research* **31**, 591–656.

Dreżewski, R. (2003), A model of co-evolution in multi-agent system, *in* V. Mařík, J. Müller & M. Pěchouček, eds, 'Multi-Agent Systems and Applications III', Vol. 2691 of *LNCS*, Springer-Verlag, Berlin, Heidelberg, pp. 314–323.

Dreżewski, R. (2006), 'Co-evolutionary multi-agent system with speciation and resource sharing mechanisms', *Computing and Informatics* **25**(4), 305–331.

Dreżewski, R. & Cetnarowicz, K. (2007), Sexual selection mechanism for agent-based evolutionary computation, *in* Y. Shi, G. D. van Albada, J. Dongarra & P. M. A. Sloot, eds, 'Computational Science – ICCS 2007', Vol. 4488 of *LNCS*, Springer-Verlag, Berlin, Heidelberg, pp. 920–927.

Dreżewski, R. & Siwik, L. (2010), A review of agent-based co-evolutionary algorithms for multi-objective optimization, *in* 'Computational Intelligence in Optimization. Aplication and Implementations', Springer-Verlag, Berlin, Heidelberg.

Fogel, D. B. (1998), *Evolutionary Computation: The Fossil Record. Selected Readings on the History of Evolutionary Computation*, IEEE Press.

Fonseca, C. M. & Fleming, P. J. (1995), 'An Overview of Evolutionary Algorithms in Multiobjective Optimization', *Evolutionary Computation* **3**(1), 1–16.

Franklyn, S. & Graesser, A. (1997), Is it an agent, or just a program?: A taxonomy for autonomous agents, *in* 'Intelligent agents III: Agent Theories, Architectures and Languages', Vol. 1193/1997 of *LNCS*, pp. 21–35.

Fusinska, B., Kisiel-Dorohinicki, M. & Nawarecki, E. (2007), Coevolution of a fuzzy rule base for classification problems, *in* 'Rough sets and intelligent systems paradigms: international conference, RSEISP 2007, Warsaw, Poland', Vol. 4585 of *LNCS/LNAI*, Springer, pp. 678–686.

George, J., Gleizes, M., P.Glize & C.Regis (2003), Real-time simulation for flood forecast: an adaptive multi-agent system staff, *in* 'Proceedings of the AISB'03 Symposium on Adaptive Agents and Multi-Agent Systems', University of Wales.

Horst, R. & Pardalos, P. (1995), *Handbook of Global Optimization*, Kluwer.

Jennings, N., Faratin, P., Johnson, M., Norman, T., OBrien, P. & Wiegand, M. (1996), 'Agent-based business process management', *International Journal of Cooperative Information Systems* **5**(2-3), 105–130.

Kisiel-Dorohinicki, M. (2002), Agent-oriented model of simulated evolution, *in* W. I. Grosky & F. Plasil, eds, 'SofSem 2002: Theory and Practice of Informatics', Vol. 2540 of *LNCS*, Springer.

Lobel, B., Ozdaglar, A. & Feijer, D. (2011), 'Distributed multi-agent optimization with state-dependent communication', *Mathematical Programming* **129**(2), 255–284.

Mahfoud, S. W. (1992), Crowding and preselection revisited, *in* R. Männer & B. Manderick, eds, 'Parallel Problem Solving from Nature — PPSN-II', Elsevier, Amsterdam, pp. 27–36.

Mahfoud, S. W. (1995), Niching methods for genetic algorithms, PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, USA.

McArthur, S., Catterson, V. & Hatziargyriou, N. (2007), 'Multi-agent systems for power engineering applicationspart i: Concepts, approaches, and technical challenges', *IEEE TRANSACTIONS ON POWER SYSTEMS* **22**(4), 1743–1752.

Moya, L. J. & Tolk, A. (2007), Towards a taxonomy of agents and multi-agent systems, *in* 'Proceedings of the 2007 spring simulation multiconference - Volume 2', Society for Computer Simulation International, San Diego, CA, USA, pp. 11–18.

Paredis, J. (1995), 'Coevolutionary computation', *Artificial Life* **2**(4), 355–375.

Piętak, K., Woś, A., Byrski, A. & Kisiel-Dorohinicki, M. (2009), Functional integrity of multi-agent computational system supported by component-based implementation, *in* V. Mařík, T. Strasser & A. Zoitl, eds, 'Proceedings of the 4th International Conference on Industrial Applications of Holonic and Multi-agent Systems', Vol. 5696 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 82–91.

Potter, M. A. & De Jong, K. A. (2000), 'Cooperative coevolution: An architecture for evolving coadapted subcomponents', *Evolutionary Computation* **8**(1), 1–29.

Russell, S. J. & Norvig, P. (2009), *Artificial Intelligence: A Modern Approach*, 3rd edition edn, Prentice Hall.

Sánchez-Velazco, J. & Bullinaria, J. A. (2003), Gendered selection strategies in genetic algorithms for optimization, *in* J. M. Rossiter & T. P. Martin, eds, 'Proceedings of the UK Workshop on Computational Intelligence (UKCI 2003)', University of Bristol, Bristol, UK, pp. 217–223.

Sarker, R. & Ray, T. (2010), *Agent-Based Evolutionary Search*, Vol. 5 of *Adaptation, Learning and Optimization*, 1 edn, Springer.

Schaefer, R., Byrski, A. & Smołka, M. (2009), 'Stochastic model of evolutionary and immunological multi-agent systems: Parallel execution of local actions', *Fundamenta Informaticae* **95**(2–3), 325–348.

Schaefer, R. & Kołodziej, J. (2003), 'Genetic search reinforced by the population hierarchy', *Foundations of Genetic Algorithms* **7**, 383–399.

Siwik, L. & Dreżewski, R. (2009), Agent-based multi-objective evolutionary algorithms with cultural and immunological mechanisms, *in* W. P. dos Santos, ed., 'Evolutionary Computation', InTech, Vienna, pp. 541–556.

Siwik, L. & Natanek, S. (2008), Solving constrained multi-criteria optimization tasks using elitist evolutionary multi-agent system, *in* 'Proceedings of 2008 IEEE World Congress on Computational Intelligence (WCCI 2008), 2008 IEEE Congress on Evolutionary Computation (CEC 2008)', IEEE, Research Publishing Services, pp. 3357–3364.

Uhruski, P., Grochowski, M. & Schaefer, R. (2008), 'A two-layer agent-based system for large-scale distributed computation', *Computational Intelligence* **24**(3), 191–212.

Van Veldhuizen, D. A. (1999), Multiobjective Evolutionary Algorithms: Classifications, Analyses and New Innovations, PhD thesis, Graduate School of Engineering of the Air Force Institute of Technology Air University.

Veldhuizen, D. A. V. & Lamont, G. B. (2000), ‘Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art’, *Evolutionary Computation* **8**(2), 125–147.

Wierzchoń, S. (2002), ‘Function optimization by the immune metaphor’, *Task Quaterly* **6**(3), 1–16.

Wolpert, D. H. & Macready, W. G. (1997), ‘No free lunch theorems for optimization’, *IEEE Transactions on Evolutionary Computation* **1**(1), 67–82.

Wooldridge, M. (2009), *An Introduction to Multiagent Systems*, John Wiley & Sons.

Zhong, W., Liu, J., Xue, M. & Jiao, L. (2004), ‘A multiagent genetic algorithm for global numerical optimization’, *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics* **34**(2), 1128–1141.

Zitzler, E. (1999), Evolutionary algorithms for multiobjective optimization: methods and applications, PhD thesis, Swiss Federal Institute of Technology, Zurich.