Mateusz Różański
Leszek Siwik

EVOLUTIONARY MULTI-AGENT SYSTEM WITH CROWDING FACTOR AND MASS CENTER MECHANISMS FOR MULTI-OBJECTIVE OPTIMIZATION

Abstract

This work presents some additional mechanisms for evolutionary multi-agent systems for multi-objective optimization that attempt to solve problems with population stagnation and loss of diversity. These mechanisms reward solutions located in a less-crowded neighborhood and on the edges of a frontier. Both techniques have been described, and some preliminary results have been shown.

Keywords   evolutionary computation, multi-agent system, multi-objective optimization

1. Introduction

Nearly every field (science, technology, economics, and everyday life) requires us to take various (often contradictory) criteria into account when making decisions. These reasons have motivated researchers to conduct studies on efficient algorithms for solving multi-objective optimization problems (MOOPs). At this point, some of the most efficient and universal MOOP solvers are considered evolutionary algorithms, which are based on Darwin's selection theory. This approach is also characterized by evolutionary multi-agent systems and as has been shown to give promising results [2–4, 17, 21, 22]. Two problems with these algorithms are population stagnation and a lack of diversity. There have been many ideas to improve these aspects based on such mechanisms as genders [9], species [7,10], elitism [18,19], and others [8]. This work presents a different approach to this problem: mechanisms based on dominations and crowding factors as well as a mass center. The presented results show that these methods significantly improve the algorithms' performance.

The paper is organized as follows. In Section 2, basic concepts of multi-objective optimization and the evolutionary approach are presented. In Section 3, evolutionary multi-agent systems are described for both general use and multi-objective optimization. In Sections 4 and 5, the proposed mechanisms are presented and explained. In Section 6, the testing criteria and testing problems are described. In Section 7, the preliminary results are presented and explained. And finally, the conclusions as well as any future possibilities and plans are presented in Section 8.

2. Evolutionary approach for MOOP

The multi-objective optimization problem can be defined as follows [1, 5, 15]:

$$
MOOP \equiv \begin{cases}
\text{Min/Max} & f_m(\bar{x}), & m = 1, 2, ..., M \\
\text{With consideration} & g_j(\bar{x}) \geq 0, & j = 1, 2, ..., J \\
& h_k(\bar{x}) = 0, & k = 1, 2, ..., K \\
& x_i^{(L)} \leq x_i \leq x_i^{(U)}, & i = 1, 2, ..., N
\end{cases} \tag{1}
$$

The constraint set – both *equalities* $h_k(\bar{x})$ and *inequalities* $g_j(\bar{x})$ as well as *lower bounds* $x_i^{(L)}$ and *upper bounds* $x_i^{(U)}$ – defines the searchable space (feasible alternatives) and, therefore, a set of all possible solutions – D.

The basic concept of optimality in the Pareto sense is the *domination relation*. Alternative $\vec{x}^b$ dominates $\vec{x}^a$ if and only if [6]:

$$
\forall m \quad f_m(\vec{x}^a) \leq f_m(\vec{x}^b) \quad \text{and} \quad \exists m \quad f_m(\vec{x}^a) < f_m(\vec{x}^b) \tag{2}
$$

The solution of the multi-objective optimization problem is the set of non-dominated solutions (the so-called Pareto set [12,25]) constituting the Pareto frontier in the objective space.

A natural approach to solve the task of multi-objective optimization (especially during the initial period of the research) was an attempt to apply a known apparatus (both analytical and numerical) developed and used in the context of solving tasks in single-criterion optimization. These methods turned out to be ineffective due to the following [6]:

- models were too complex,
- formulas applied were too complicated,
- numerical instability of available solvers.

Among different contemporary methods for solving poly-optimization problems, evolutionary methods are fairly effective and commonly used. The main reason is that they process whole populations; as a result, they produce a set of solutions that is believed to be an approximation of the Pareto frontier. At the same time, heuristic methods can be used to solve search and optimization problems for which there are no exact methods or whose problems are of unknown characteristics [11, 13].

This approach turned out to be effective and has had many applications, so a new branch of research and development arose; this is known nowadays as evolutionary multi-objective optimization [5, 6].

3. Evolutionary multi-agent system – EMAS

Despite the huge potential of these methods as well as their multiple effective applications for solving difficult optimization and search problems, these methods did not give the expected results in many cases. The reasons for this include the excessive simplification of the evolution process, the inability of agents to acquire and store any knowledge (and thus the lack of an agent's influence on decisions), and the absence of basic natural mechanisms such as interactions with other individuals, competition, cooperation, and building social structures. The agent-based approach is another step in the development of evolutionary MOOP algorithms that are trying to improve these aspects. The individuals represented by agents have gained possibilities that are naturally occurring in the process of evolution: autonomy and the ability to make independent decisions, exchange knowledge, and build social structures. Agents can observe the environment and make specific decisions and take actions according to the problem and actual environment state. As the evolution process is decentralized, the selection mechanism was built on the basis of exhaustible resources. In most cases, only one is used — energy, which is acquired by the agent during meetings with other agents: the better agent takes resources from the worse agent. The assessment of the agent depends in this case on the problem that is being solved. Agents that acquire the appropriate level of energy can use it in the process of migration or reproduction (which itself is the goal of evolution), while those whose energy amounts fall below a certain level die and are removed from the population (see Fig. 1). Some of the energy lost by individuals during migration and dying is also possessed by the environment and can be used to create new random individuals. As the amount of energy in the whole ecosystem is constant, it can be used to regulate the population size.
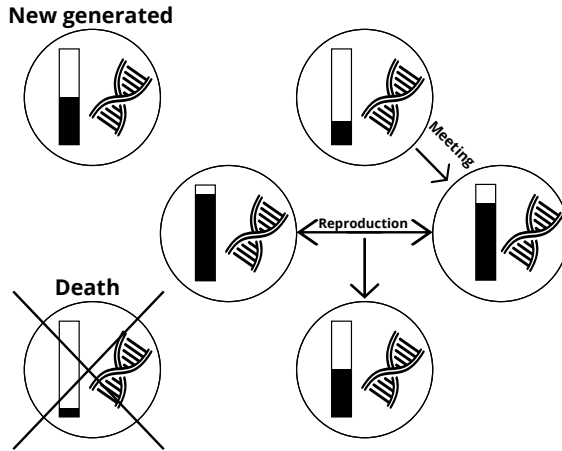
New generated

Death

Meeting

Reproduction

Figure 1. EMAS agent basic interactions

The environment can be divided into islands (see Fig. 2) representing physically separate places, symbolizing niches, or grouping agents into local groups to which access may also be limited in some way [18,19]. The agent can travel between islands during the migration process if it has the appropriate level of energy. A portion of this energy is given to the environment during the process.
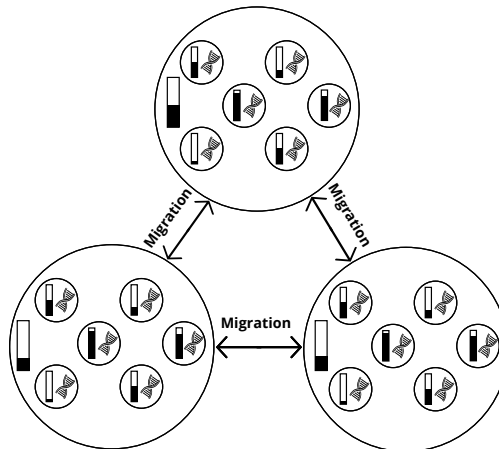
Migration

Migration

Migration

Figure 2. EMAS environment organization

4. Evolutionary multi-agent system for multi-objective optimization

In the case of solving a multi-criteria optimization problem, the agent's evaluation and energetic selection mechanism may be based on the domination relation [20]. If a given individual dominates another during a meeting process, it takes some of the other's energy. In this way, agents that often dominate other individuals and are dominated relatively rarely can acquire the energy needed to survive and create offspring. On the other hand, agents that are dominated often enough and are statistically worse will be gradually removed from the environment. As presented in Listing 1, the life process of the individual consists of a series of strategies and actions that the agent performs iteratively in its life step.

Listing 1. Life step of individual

```
1  lifeStep() {
2     if (energy level below death threshold) {
3        dieAction();
4     } else {
5        migrStrategy();
6        meetStrategy();
7        reproStrategy();
8     }
9  }
```

During each iteration, the energy level is checked first. If it is too low, the death action is performed; if the individual has more energy, it can perform the remaining strategies: migration (if it considers it appropriate in a given situation), meetings, and (most importantly) reproduction (if it has enough energy and can find another individual also wanting to reproduce).

5. EMAS with dominations and crowding factors – fEMAS

In basic EMAS, the agents gain energy by taking it from other agents. As a consequence, the amount of energy collected by an agent results from its relative quality as compared to other agents; however, this aspect sometimes causes problems with gaining energy by relatively good agents. In Figure 3, a sample of population distribution is shown, and two agents (A and B) were selected for the meeting. We assume that we minimize F1 and F2. It can be seen that there is no domination relation between these individuals, so no energy exchange will occur in basic EMAS. It can be seen, however, that Agent B is potentially better than Agent A because there are many solutions that dominate Agent A. Agent B is not dominated by anyone and also is located on the edge of the frontier, so it should definitely reproduce; however, it will have a problem with this because there is only one agent from whom it can take energy – Individual C. Though potentially inferior, Agent A has several individuals that it can dominate; there is a considerable likelihood that it will obtain the energy

needed for reproduction. The described mechanism causes that, when the population drifts towards the frontier, it does not expand to cover the entire frontier; sometimes the opposite occurs: it focuses on the center of the population. The proposed solution to this problem is the dominations factor, which will reward Agent B in the case described above. The dominations factor is defined as the ratio of the number of dominations (how many times an agent has been dominated by other agents) to the total number of meetings. When the number of meetings is equal to zero, then the factor value is also zero.
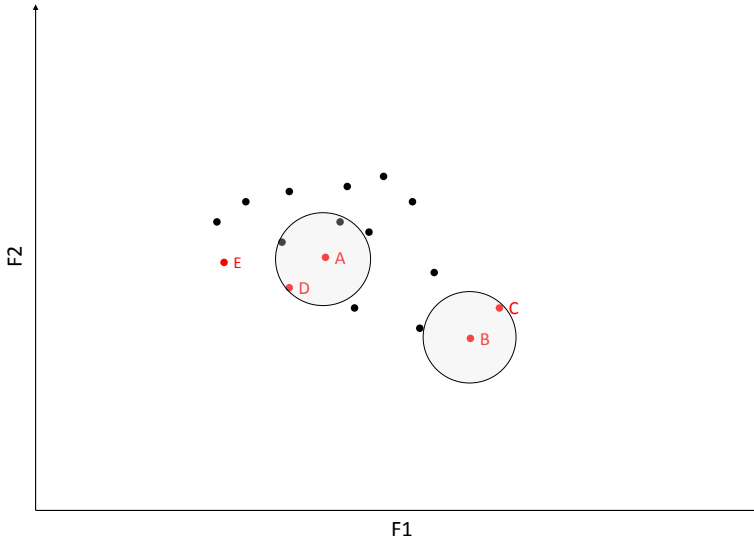

Figure 3. Sample of Population Distribution

In the case presented in Figure 3, there are two other solutions (D and E) that dominate Individual A. If there will be a meeting between Solution A and one of these two individuals, Agent A will have a dominations factor that is greater than 0; because of this, it will be ranked worse when it comes to a meeting with Agent B. If this does not happen and both Agents A and B have a dominations factor equal to zero, there is also the crowding factor [8,16]. This rewards solutions are arranged in less-crowded regions in the absence of a domination relation and equal the dominations factor. The crowding factor is defined as the ratio of the number of meetings in a defined radius to the total number of meetings. When the number of meetings is equal to zero, then the factor value is also zero. In this case, Agent B (which has only one neighbor) should also be ranked better and receive energy from Agent A (who has three agents in its radius).

Including the dominations and crowding factors into EMAS consists of a modification of the agent's meeting strategy. The original procedure in EMAS is presented in Listing 2. The modified strategy in fEMAS is presented in Listing 3.

Listing 2. Meeting Strategy in basic EMAS

```
1  meetStrategy ( ) {
2      if ( Agent A wants to meet ) {
3          B = seekAgentToMeet (A) ;
4      }
5      if ( Agent B accepts meeting ) {
6          if ( Agent A dominates Agent B) {
7              transferEnergy (B, A) ;
8          }
9          if ( Agent B dominates Agent A) {
10             transferEnergy (A, B)
11         }
12     }
13 }
```

Listing 3. Modified Meeting Strategy in fEMAS

```
1  meetStrategy ( ) {
2      if ( Agent A wants to meet ) {
3          B = seekAgentToMeet (A) ;
4      }
5      if ( Agent B accepts meeting ) {
6          if ( Agent A dominates Agent B) {
7              transferEnergy (B, A) ;
8          }
9          if ( Agent B dominates Agent A) {
10             transferEnergy (A, B)
11         }
12         if ( no dominations between Agents A and B) {
13             if ( A. dominationsFactor < B. dominationsFactor ) {
14                 transferEnergy (B, A) ;
15             }
16             if ( A. dominationsFactor > B. dominationsFactor ) {
17                 transferEnergy (A, B) ;
18             }
19             if ( A. dominationsFactor == B. dominationsFactor ) {
20                 if ( A. crowdingFactor < B. crowdingFactor ) {
21                     transferEnergy (B, A) ;
22                 }
23                 if ( A. crowdingFactor > B. crowdingFactor ) {
24                     transferEnergy (A, B) ;
25                 }
26             }
27         }
28     }
29 }
```

It can be seen that the basic comparative criterion is dominance. If such a relation does not take place, the dominations factor is taken into account, followed by the crowding factor.

6. EMAS with mass center mechanism – mcEMAS

Another proposed solution counteracting the fact that the solutions are concentrated in the center of the frontier and also improving the frontier coverage is the mechanism of the mass center. In this mechanism, there are two islands between which agents can migrate. As mentioned before, agents give part of their energy to the environment during the migration process; from this energy, new agents can be generated. In basic EMAS, agents are generated randomly. In the proposed mechanism, this aspect has been modified, with agents being generated on the edges of the frontier. In order to achieve such an effect, the individuals gather information about the position of other individuals during meetings and, on the basis of this, calculate the geometric center of the population. In addition, they also store information about an agent located at the farthest position from this center point. When this agent dies, it sends this information to the environment, and the environment creates a new individual using this information and combining with the mutation operator.

The mass center is based on a weighted average and calculated as follows: The initial value of the mass center point is the position of the given agent. During the meetings, agents exchange information about the calculated mass center and update their values by calculating the average between two points, taking into account the number of meetings (see Listing 4).

Listing 4. Mass center calculation method

```
1  massCenter = (A.massCenter * A.meetingsCounter
2          + B.massCenter * B.meetingsCounter
3          / (A.meetingsCounter + B.meetingsCounter);
```

7. Testing criteria

To check the quality of the proposed solutions, test problems from the ZDT family [23] were used.

All ZDT problems have the same basic schema:

$$ZDT = \begin{cases} \text{Min } F(x) = (f_1(x_1), f_2(x)) \\ \text{With consideration } f_2(x) = g(x_2, ..., x_n) \cdot h(f_1(x_1), g(x_2, ..., x_n)) \end{cases} \tag{3}$$

where $x = (x_1, ..., x_n)$.

ZDT1, the first and simplest of the set of ZDT problems, is characterized by a convex and continuous Pareto front.

It is defined as follows:

$$ZDT1 = \begin{cases} f_1(x) = x_1 \\ g(x_2, ..., x_n) = 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i \\ h(f_1, g) = 1 - \sqrt{f_1/g(x)} \\ \text{where } n = 30, x_i \in [0, 1] \end{cases} \qquad (4)$$

A visualization of the Pareto frontier of the ZDT1 problem is presented in Figure 4.

The ZDT2 problem is a problem introducing the first potential difficulty for the optimization algorithm: the concavity of the Pareto frontier. This problem is defined as follows:

$$ZDT2 = \begin{cases} f_1(x) = x_1 \\ g(x_2, ..., x_n) = 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i \\ h(f_1, g) = 1 - (f_1/g(x))^2 \\ \text{where } n = 30, x_i \in [0, 1] \end{cases} \qquad (5)$$

A visualization of the Pareto frontier of the ZDT2 problem is presented in Figure 5.

The ZDT3 problem is a test problem introducing another difficulty for the optimization algorithm: discontinuity of the Pareto frontier. The problem is defined as follows:

$$ZDT3 = \begin{cases} f_1(x) = x_1 \\ g(x_2, ..., x_n) = 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i \\ h(f_1, g) = 1 - \sqrt{f_1/g(x)} - \frac{f_1}{g(x)} \sin(10\pi f_1) \\ \text{where } n = 30, x_i \in [0, 1] \end{cases} \qquad (6)$$

A visualization of the Pareto frontier of the ZDT3 problem is presented in Figure 6.

The ZDT4 problem is a problem allowing us to evaluate the behavior of a tested algorithm in a situation of solving the multi-modal MOOP problem. The ZDT4 implements 21^9 local Pareto frontiers and is defined as follows:

$$ZDT4 = \begin{cases} f_1(x) = x_1 \\ g(x_2, ..., x_n) = 1 + 10(n - 1) + \sum_{i=2}^{n}(x_i^2 - 10\cos(4\pi x_i)) \\ h(f_1, g) = 1 - \sqrt{f_1/g(x)} \\ \text{where } n = 10, x_i \in [0, 1], x_i \in [-5, 5] \end{cases} \qquad (7)$$

A visualization of the Pareto frontier of the ZDT4 problem is presented in Figure 7.
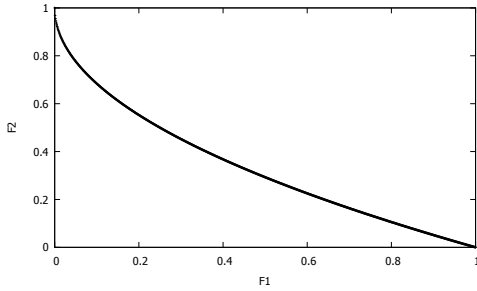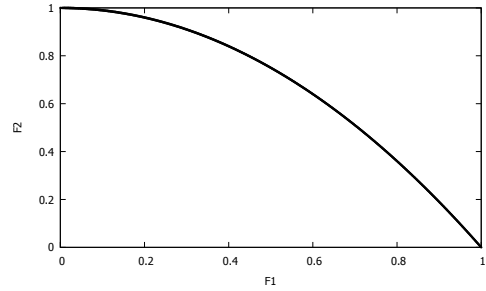
Figure 4. True Pareto frontier for ZDT1

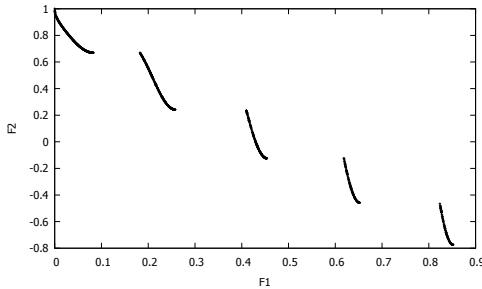**Figure 5.** True Pareto frontier for ZDT2

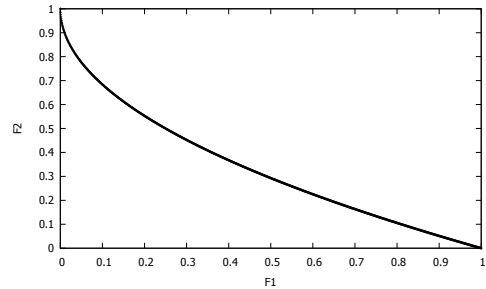**Figure 6.** True Pareto frontier for ZDT3

Figure 7. True Pareto frontier for ZDT4

To evaluate the quality of the proposed solutions, a hypervolume indicator (HV) and hypervolume ratio (HVR) [24,25] were used. The hypervolume measures the area covered by a resulting set of solutions with a reference point. In the HVR, this result is divided by the HV value for the model Pareto frontier for a particular problem:

$$HV = v(\sum_{i=1}^{N} v_i) \tag{8}$$

$$HVR = \frac{HV(PF*)}{HV(PF)} \tag{9}$$

where v_i is hypercube for the i-th solution, $PF*$ represents the result set, and PF is the Pareto frontier.

All algorithms have been implemented in Java using the jMetal framework [14] (evolutionary operators, test problems, and indicators). The energy level was adjusted so that the population oscillated at around 500 individuals. Other parameters are as follows:

- initial energy – 30,
- transfer energy between agents – 10,
- minimal reproduction energy level – 60,
- die energy level – 0.

8. Experimental studies

The measurements were carried out regarding not only the number of iterations but also the number of evaluations; they are presented in the form of tables and various types of diagrams. Since both mechanisms affect different aspects of EMAS, a combination of both methods was also tested: the mechanism for generating new agents from the energy given to the environment in the migration process was taken from mcEMAS, and the way agents were evaluated in the domination relation absence from fEMAS.

8.1. ZDT 1

Some of the most important features of a good multi-objective optimization algorithm are fast convergence and good coverage of the model frontier. To show these aspects, the results for 100, 200, 500, 1000, and 2000 iterations were presented. Table 1 shows the exact results for the HVr metric presented as average values with standard deviation. On the other hand, the diagrams present examples of how the solutions are arranged.

Table 1
HVr values obtained during solving ZDT1 Problem

Iter.	EMAS		fEMAS		mcEMAS		Combined	
	Mean	St. Dev	Mean	St. Dev	Mean	St. Dev	Mean	St. Dev
100	0.00	0.00	0.04	0.04	0.02	0.02	0.03	0.02
200	0.11	0.08	0.40	0.03	0.29	0.11	0.35	0.07
500	0.62	0.05	0.76	0.03	0.71	0.07	0.71	0.05
1000	0.67	0.03	0.91	0.01	0.90	0.03	0.88	0.02
2000	0.66	0.03	0.97	0.01	0.95	0.02	0.96	0.01

In Table 1 as well as in the diagrams, it can be seen that the results for 100 and 200 iterations are non-satisfying regardless of the used algorithm, and the result set is far away from the model Pareto frontier. After 500 iterations, the population is much closer to the model frontier, but the result still cannot be considered good enough. However, it can be noticed that the applied mechanisms start working and significantly improve the effectiveness of the algorithm: the coverage of the frontier is more accurate, and the results are better by 0.09 on average (a 15% improvement) for mcEMAS and the combined solutions to even 0.14 (23%) for fEMAS. After 1000 iterations, the resulting frontier is much closer to the model frontier; also, the coverage of the frontier for fEMAS and mcEMAS is much better than in basic EMAS. The results are better by 0.24 on average (36%) for fEMAS, 0.23 (34%) for mcEMAS, and 0.21 (31%) for the combined methods. Between 1000 and 2000 iterations, the improvement of the results is much smaller; in the case of basic EMAS, there is even

a slight deterioration (this is because the algorithms were launched independently; however, the result is contained within the standard deviation range). From the above results, it can be deduced that a further increase in the iterations for basic EMAS is pointless (cf. Fig. 8). The best result – 0.97 (47% improvement) – comes from fEMAS (cf. Fig. 9); this is very close to the model frontier. McEMAS (cf. Fig. 10) and the combination of both solutions give slightly worse outcomes (0.95 [44%] and 0.96 [45%], respectively), but these can also be considered satisfactory. The combined results are presented in Figure 11.
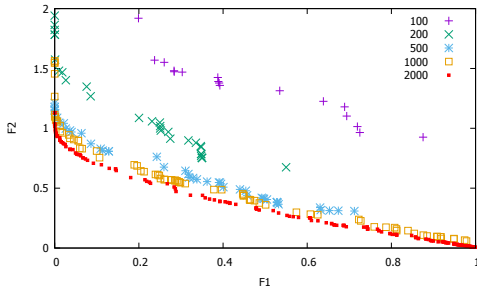

Figure 8. EMAS

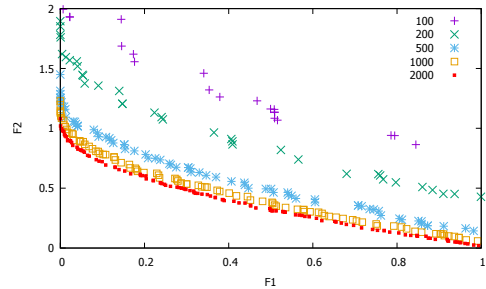**Figure 9.** fEMAS

Figure 10. mcEMAS

Figure 11. Combined

To have a better view of the results, a chart showing the value of the HVr metric has also been presented for each iteration (see Fig. 12). Also in this diagram, it can be seen that the implemented mechanisms have a very similar impact and significantly improve the results. It can be noticed that, in contrast to basic EMAS, they smoothly improve the quality of the result in the proposed algorithms – this happens because new solutions with good fitness are generated and born regularly.

In fEMAS, this occurs because good solutions have a better chance of gaining energy and, as an effect, reproducing. In mcEMAS, the energy returned to the environment by the dominated solutions is given to the newly generated solutions on the edges of the frontier; in most cases, this also improves the HVr value.
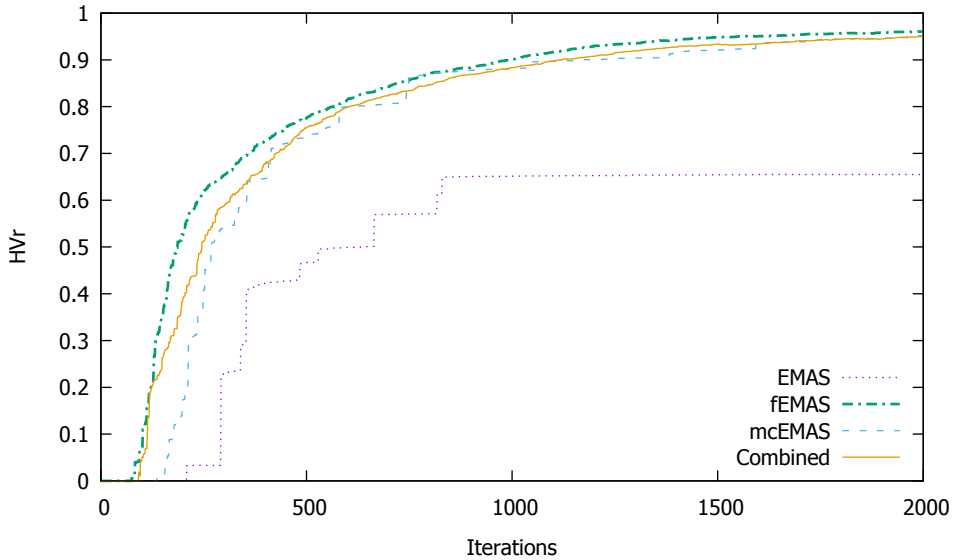
Figure 12. HVr values for each iteration

For complicated problems, calculating the fitness function can be difficult and time-consuming. In such cases, an important factor to take into account is the number of evaluations needed for calculation in the process of obtaining the final result. Table 2 and Figure 13 present a comparison of the number of evaluations for all methods for 1000 and 2000 iterations. It can be clearly seen that mcEMAS has the best results in both cases: for 2000 iterations, this makes about 25% fewer evaluations than in fEMAS. This phenomenon can be explained as follows: in the case of basic EMAS and fEMAS, many average individuals with energies around the reproduction threshold are able to collect enough energy to deliver descendants (which of course need to be evaluated). Many of these new solutions are also average and are to be eliminated in subsequent generations.

Table 2
Number of evaluations for ZDT1 Problem

	EMAS		fEMAS		mcEMAS		Combined	
Iter.	Mean	St. Dev	Mean	St. Dev	Mean	St. Dev	Mean	St. Dev
1000	23,727	1224	18,127	264	17,394	705	19,636	200
2000	25,974	2437	28,792	416	21,680	1043	32,014	406

In mcEMAS, the agents also lose energy in the migration process; because of this, there are fewer of these average new solutions. Also, mcEMAS generates new agents from the free energy, and these new agents appear on the edges of the frontier.

This mechanism, however, is quite aggressive and can also exclude relatively good solutions from reproduction. As an effect, the frontier is not as accurately covered as in the case of fEMAS. Confirmation of this can be seen in the obtained results and charts.
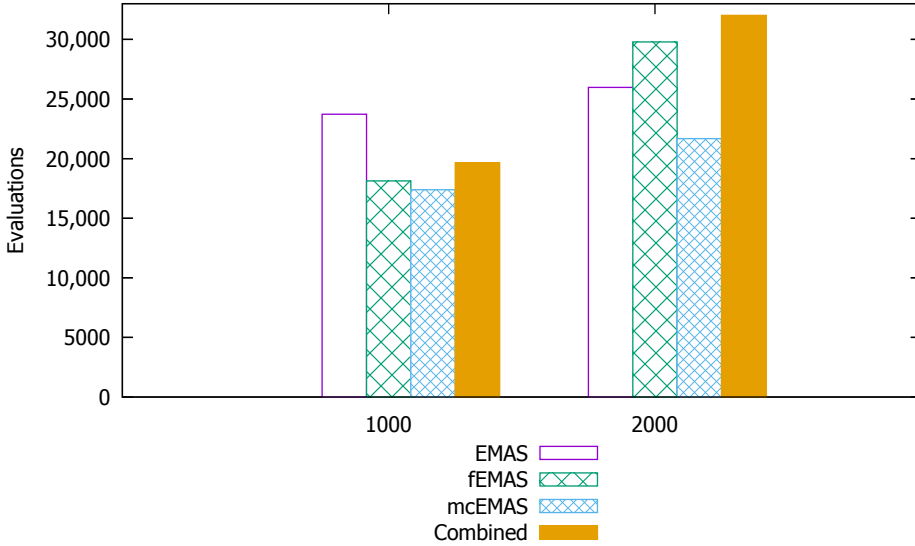

Figure 13. Number of evaluations after 1000 and 2000 iterations

8.2. ZDT 2

Table 3 presents the HVr values for the ZDT2 problem. It can be seen that the results for basic EMAS are completely unsatisfactory for all cases – even for 2000 iterations, the ratio is around 0.19.

Table 3
HVr values obtained during solving ZDT2 Problem

	EMAS		Factors		MassCenter		Combined	
Iter.	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev
100	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
200	0.0	0.0	0.02	0.06	0.0	0.0	0.07	0.08
500	0.11	0.04	0.68	0.04	0.51	0.15	0.75	0.09
1000	0.18	0.03	0.91	0.03	0.78	0.08	0.90	0.04
2000	0.19	0.01	0.95	0.01	0.91	0.04	0.94	0.01

For the other methods, the results for ZDT2 are similar to the previous problem: for 100 and 200 iterations, the result is far away from the model frontier – the HVr is equal to zero in almost all cases. For 500 iterations, it oscillates between 0.51 for mcEMAS (363% better than basic EMAS, see Fig. 14) and 0.75 (519%) for the combined methods. For 1000 and 2000 iterations, the result is very good: 0.95 (400%) for fEMAS (cf Fig. 15) and 0.94 (395%) for the combined methods. For mcEMAS, it is slightly worse (0.91) (a 379% improvement); still, it can be considered satisfactory. It can be noticed that the algorithms with the mass center mechanism have a relatively large standard deviation for the HVr value – this happens because this mechanism can be too aggressive in some cases.

When a new solution is generated on the edge of the frontier, the mutation operator sometimes "works" too well; this new agent is too far away from the end of the frontier. In basic EMAS, this would be very good for the result; but in mcEMAS, there are no mechanisms to fill this gap between the new solution and the rest of the population. This situation can be seen in Figure 16. The combined results are presented in Figure 17.
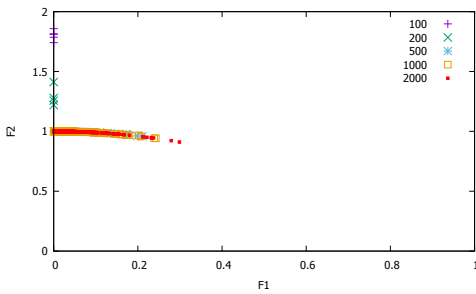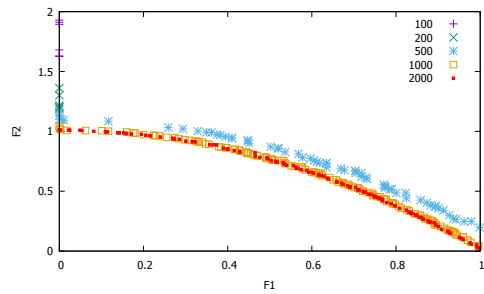
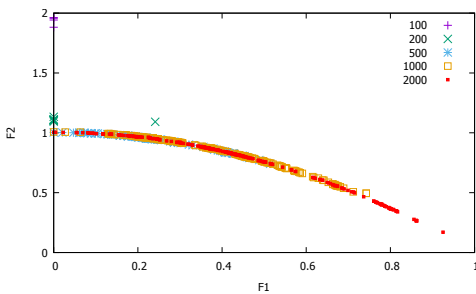

Figure 14. EMAS

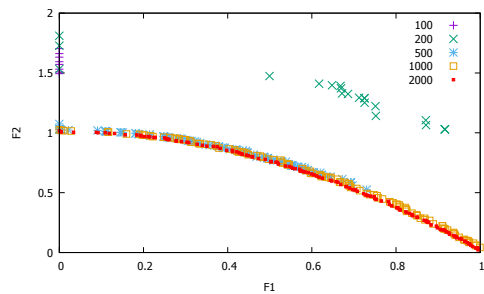**Figure 15.** fEMAS

Figure 16. mcEMAS

Figure 17. Combined

Likewise to the previous problem, Figure 18 presents a sample distribution of the HVr values for each iteration. It can be seen that, in the first stage of the algorithm's operation, mcEMAS has the best results – this can be explained by this aggressive

impact of the mass center mechanism described in the previous paragraph. After 1000 and 2000 iterations, it can be seen that the factor mechanism takes over, and fEMAS and combined solutions gain the best results.
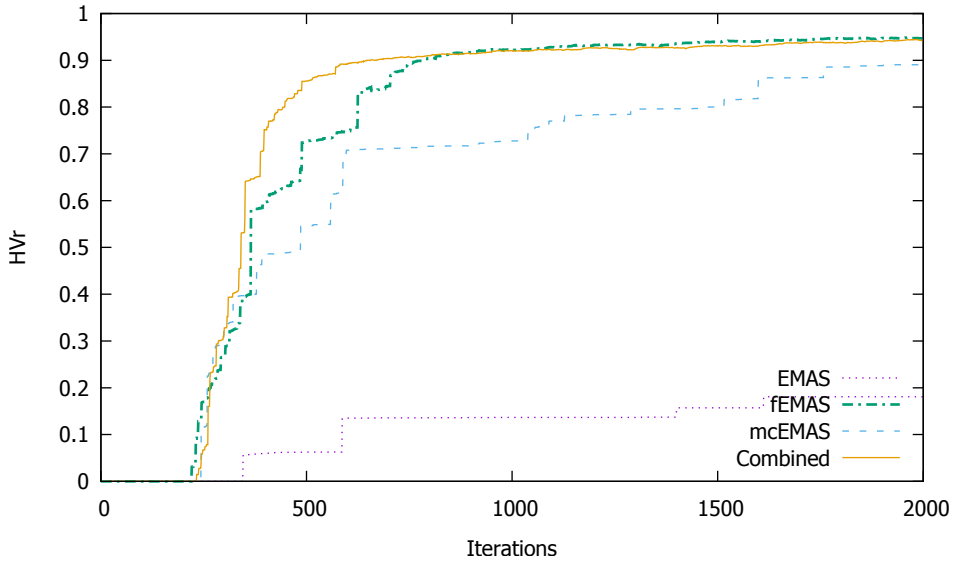

Figure 18. HVr values for each iteration

Table 4 and Figure 19 present the number of evaluations for the considered algorithms. It can be seen that, for 1000 iterations, the results for fEMAS, mcEMAS, and combined methods are similar (oscillating at around 30k). For 2000 iterations, mcEMAS has the best result – it is 10k less than fEMAS and 15k less than the combined methods. Also, it is only around 2000 evaluations more than in the 1000-iteration case. This happens because most agents are non-dominated at this stage, so they do not lose or gain energy. Because of this, not many new agents are generated or borne (or evaluated). Basic EMAS also has the worst results in this measure – the number of evaluations is almost twice as many as in mcEMAS.

Table 4
Number of evaluations for ZDT2 Problem

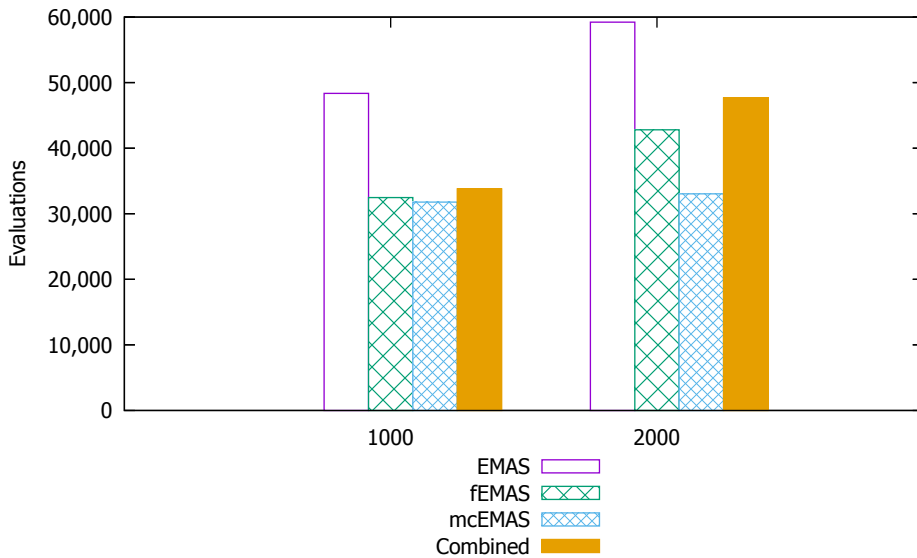|	EMAS		fEMAS		mcEMAS		Combined	
Iter.	Mean	St. Dev	Mean	St. Dev	Mean	St. Dev	Mean	St. Dev
1000	48,360	1450	32,481	4814	31,779	2422	33,787	4250
2000	59,224	1587	42,812	4863	33,038	2375	47,655	3600

Figure 19. Number of evaluations after 1000 and 2000 iterations

8.3. ZDT 3

Table 5 presents the HVr values for the ZDT3 problem. Basic EMAS finished with an HVr value equal to 0.63-0.64 (which again cannot be considered satisfactory). Similar to the other problems, the best results are obtained by fEMAS – they are 0.87 (a 35% improvement) for 1000 iterations and 0.96 (52%) for 2000 iterations. Only a little worse is EMAS with combined mechanisms: consecutively 0.85 (33%) and 0.94 (49%), which is comparable to mcEMAS (0.84 (31%) for 1000 iterations and 0.91 (44%) for 2000 iterations). This can also be seen in Figure 20 – the frontier is only partially covered. Also, in Figures 21 and 29, it can be seen that the results for the proposed solutions are indeed very similar. At the same time better frontier coverage is shown in Figures 22 and 23.

Table 5
HVr values obtained during solving ZDT3 Problem

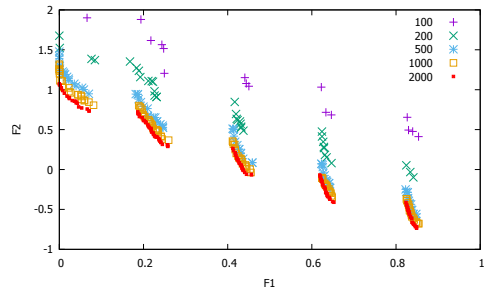|	EMAS		fEMAS		mcEMAS		Combined	
Iter.	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev
100	0.03	0.02	0.12	0.04	0.10	0.09	0.12	0.04
200	0.15	0.10	0.42	0.05	0.34	0.04	0.35	0.04
500	0.55	0.10	0.74	0.03	0.64	0.04	0.70	0.04
1000	0.64	0.10	0.87	0.03	0.84	0.04	0.85	0.02
2000	0.63	0.09	0.96	0.01	0.91	0.03	0.94	0.01

Figure 20. EMAS
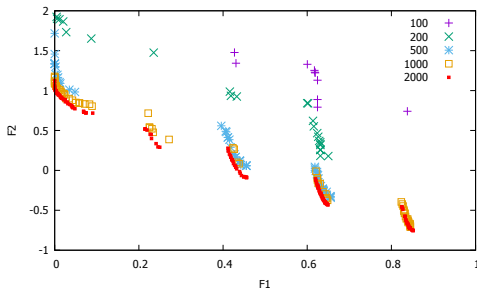
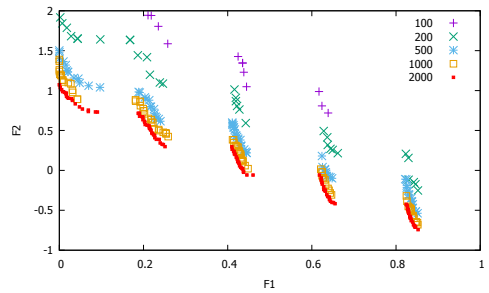

Figure 21. fEMAS

Figure 22. mcEMAS

Figure 23. Combined

In Figure 24, we can see a confirmation of the results from Table 5. It can be seen that fEMAS very quickly reaches the top level of HVr values and has the best results at every stage. Also, in this case, it can be noticed that the proposed algorithms improve the results regularly and do not have problems with population stagnation (which can be seen in basic EMAS).
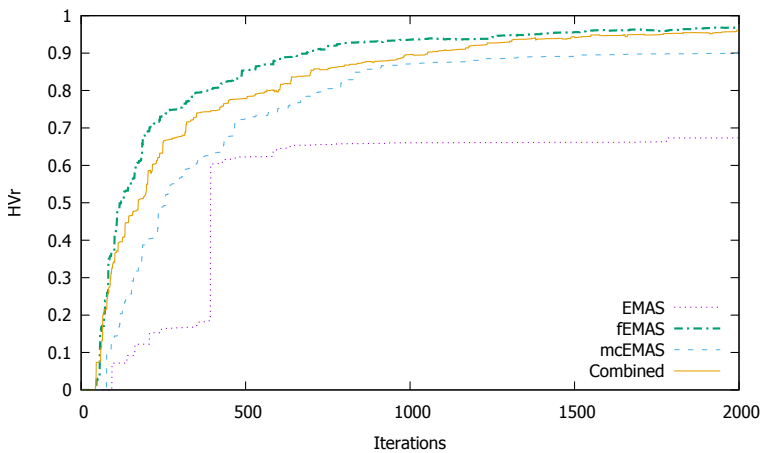

Figure 24. HVr values for each iteration

Table 6 and Figure 25 present the number of evaluations for the considered algorithms. It can be seen that, for 1000 iterations, the fewest evaluations are obtained with mcEMAS: about 15,500. The results for the other methods are worse, but the differences between the algorithms are relatively small. The real advantage of mcEMAS can be seen in the case of 2000 iterations: mcEMAS has only about 20k evaluations, where fEMAS has 28k and the combined mechanisms – 30k. Basic EMAS also has a very good result, which can be a little surprising. This can be explained as follows: the solutions are scattered in the niches of the frontier, so there is little energy exchange; in the end, not many agents can deliver a descendant. In Figure 24, we can see a confirmation of this – the HVr for basic EMAS increases irregularly and stops at the same level for many iterations. So, the relatively small number of evaluations in this case is caused by population stagnation.

Table 6
Number of evaluations for ZDT3 Problem

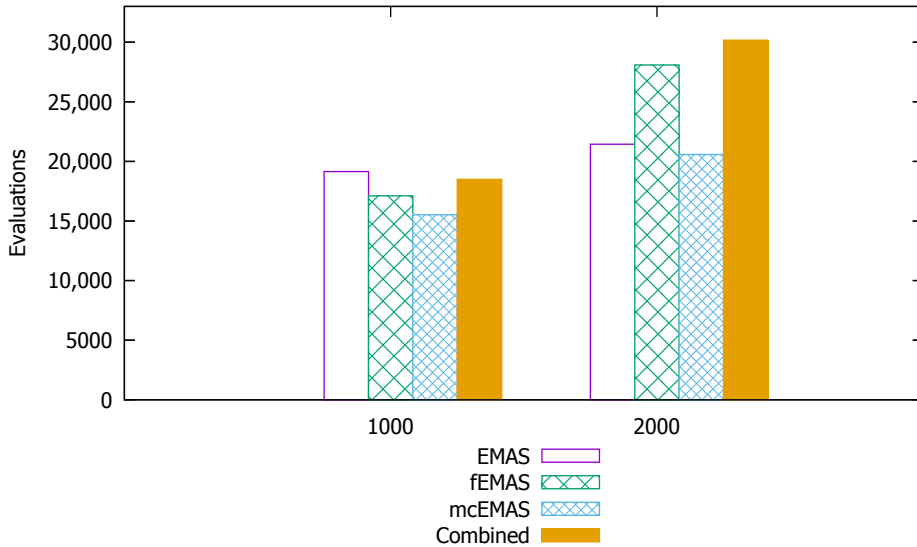|	EMAS		fEMAS		mcEMAS		Combined	
Iter.	Mean	St. Dev	Mean	St. Dev	Mean	St. Dev	Mean	St. Dev
1000	19,148	3664	17,113	268	15,517	979	18,497	453
2000	21,438	4168	28,093	737	20,584	1547	30,170	566

Figure 25. Number of evaluations after 1000 and 2000 iterations

8.4. ZDT 4

Table 7 presents the HVr values for the ZDT4 problem. Also in this test case, the best result was gained by fEMAS – 0.96 for 1000 iterations (39% better than basic EMAS) and 0.98 for 2000 iterations (36%).

Table 7

HVr values obtained during solving ZDT4 Problem

	EMAS		Factors		MassCenter		Combined	
Iter.	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev
100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
200	0.00	0.00	0.00	0.00	0.0	0.0	0.0	0.00
500	0.46	0.18	0.69	0.16	0.42	0.22	0.72	0.24
1000	0.69	0.04	0.96	0.02	0.62	0.14	0.97	0.01
2000	0.72	0.04	0.98	0.01	0.73	0.09	0.98	0.01

The algorithm with both combined mechanisms achieved similar results – accordingly, 0.97 (a 36% improvement) and 0.98 (36%). McEMAS unexpectedly gained relatively bad results – in this case, the HVr value is very similar to basic EMAS. This can be explained as follows: in the case of multimodal problems, the mechanism of the mutation very often makes the individuals fall into the local frontiers (local extrema), which causes the mutated solutions in the space criteria to be very much different. This phenomenon generates two problems: even after hundreds of iterations, the solutions are not located on the frontier but rather create something in the shape of a cloud. The agents marked as the farthest are not on the edges of the frontier; however, they very often are solutions that have very low fitness values and should not be considered as good to replicate. But even if a "good" agent located on an edge is marked as the farthest, the new generated and mutated solution is very often in one of the local frontiers with a low fitness value and will probably die after a couple of iterations. Because of these problems, the newly generated solutions are rarely located at the edges of the main frontier (as the idea of the mass center mechanism presupposes) but more randomly in the space. As a result, it can be recognized that this particular mechanism does not work at all for multimodal problems. Confirmations of this can also be seen in Figure 28 – similar to basic EMAS (see Fig. 26), the model frontier is only partially covered. A clearly better situation can be observed in Figures 27 and 29, confirmed by the observation of HVr metrics values (Fig. 30).

Paradoxically, this randomness in generating new solutions has a small but positive impact on basic EMAS. It can be seen that ZDT Problems 1 and 4 have a similar model frontier (the main difference is the existence of local frontiers), but basic EMAS has results better for about 0.1 for ZDT4 than ZDT1. This happens because there are more newborn agents that, after mutation, are located on the side of the higher F1 value of the criteria space; because of this, the agents located on this side of the frontier have statistically greater chances to dominate those new agents and gain

energy to reproduce (which was the main problem at the beginning). This shows that multimodality is not an impediment but rather a facilitation for some algorithms.
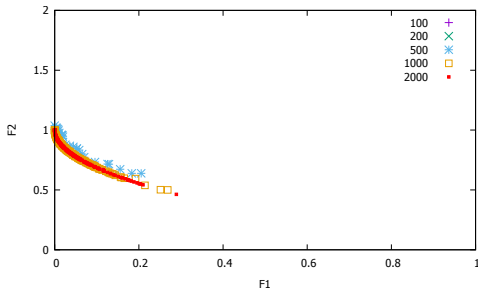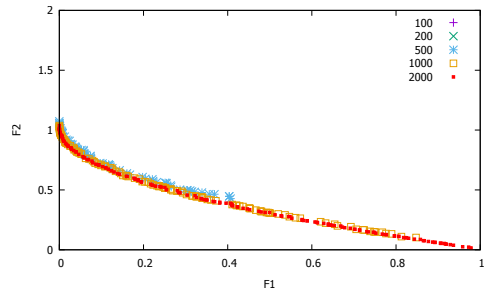

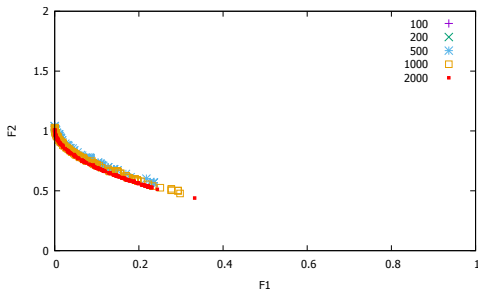**Figure 26.** EMAS

Figure 27. fEMAS

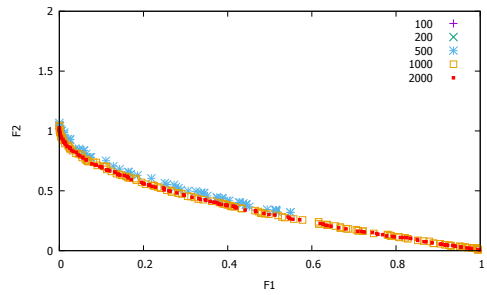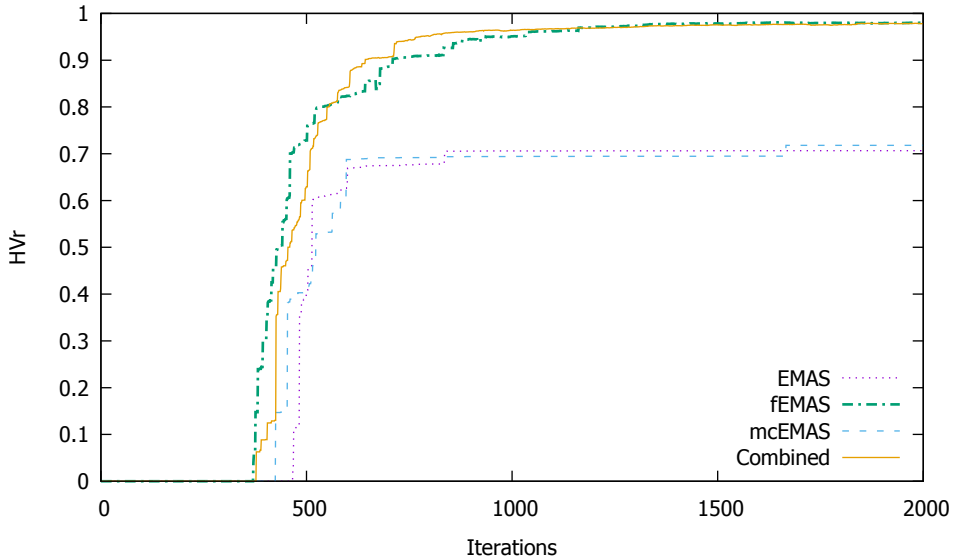
Figure 28. mcEMAS

Figure 29. Combined

Figure 30. HVr values for each iteration

Table 8 presents the evaluations for the proposed algorithms. Basic EMAS and mcEMAS have very similar results (as could be expected from the phenomenon described above) – it is about 40k regardless of the iteration number. After 2000 iterations, fEMAS has about 70k evaluations, which is similar to EMAS with the combined mechanisms; this is again not surprising, as the applied factors (especially crowding) have a dominant role here and very efficiently stimulate energy exchange (which prevents population stagnation). These results can also be seen in the Figure 31.

Table 8
Number of evaluations for ZDT4 Problem

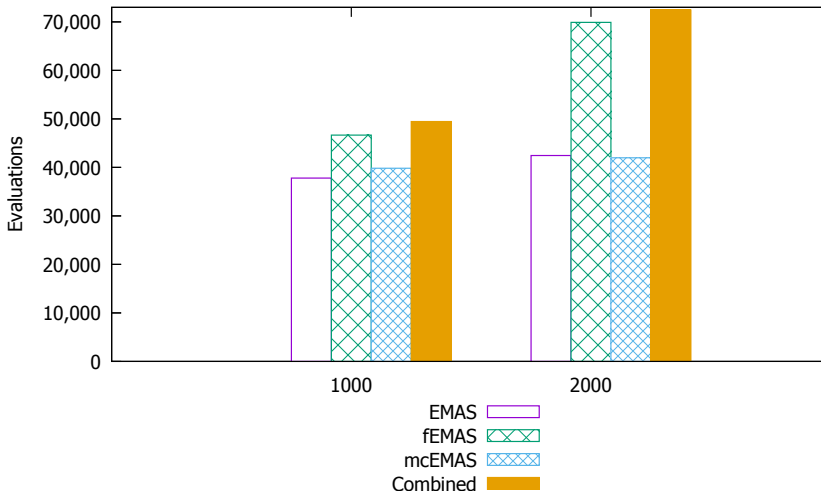| Iter.	EMAS		fEMAS		mcEMAS		Combined	
	Mean	St. Dev	Mean	St. Dev	Mean	St. Dev	Mean	St. Dev
1000	37,802	2134	46,656	1681	39,838	4386	49,447	2173
2000	42,447	1961	69,904	2106	41,992	4305	72,463	1982

Figure 31. Number of evaluations after 1000 and 2000 iterations

Similar tests have been done for another multimodal problem – ZDT6; however, the results did not bring anything new: fEMAS had the highest HVr value, and the outcome for mcEMAS was similar to basic EMAS. Because of this, they have not been presented here.

9. Conclusion

The presented results show that the proposed ideas significantly improve the quality of EMAS performance. A greater advantage of fEMAS is its simplicity and ease of

implementation. On the other hand, we append an additional argument (radius) to the system, which must be manually adapted to the particular problem that is being solved. The mass center mechanism does not have this disadvantage and additionally reduces the number of evaluations, which can be as or even more important than finding the perfect frontier in the case of complicated problems. However, this does not give such good results and is useless for multimodal problems.

Future research will include more-detailed tests of the proposed mechanism for different testing problems.

References

[1] Abraham A., Jain L.C., Goldberg R.: *Evolutionary Multiobjective Optimization Theoritical Advances and Applications*, Springer, 2005.

[2] Byrski A., Dreżewski R., Siwik L., Kisiel-Dorohinicki M.: Evolutionary multi-agent systems, *The Knowledge Engineering Review*, vol. 30(2), pp. 171–186, 2015.

[3] Byrski A., Siwik L., Kisiel-Dorohinicki M.: Designing population-structured evolutionary computation systems. In: *METHods of Artificial Intelligence (AI-METH 2003)*, pp. 91–96, Gliwice, 2003.

[4] Chen S.H., Kambayashi Y., Sato H.: *Multi-Agent Applications with Evolutionary Computation and Biologically Inspired Technologies: Intelligent Techniques for Ubiquity and Optimization*, IGI Global, 2011.

[5] Coello Coello C.A., Lemont G.B., Van Veldhiuzen D.A.: *Evolutionary Algorithms for Solving Multi-Objective Problems*, Springer, New York, 2002.

[6] Deb K.: *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, 2008.

[7] Dreżewski R., Siwik L.: Co-Evolutionary Multi-Agent System with Predator--Prey Mechanism for Multi-Objective Optimization. In: *International Conference on Adaptive and Natural Computing Algorithms*, vol. 4431, pp. 67–76, 2007.

[8] Dreżewski R., Siwik L.: Techniques for Maintaining Population Diversity in Classical and Agent-Based Multi-objective Evolutionary Algorithms. In: Shi Y., van Albada G.D., Dongarra J., Sloot P.M.A. (eds.), *Computational Science – ICCS 2007, 7th International Conference, Beijing, China, May 27–30, 2007, Proceedings, Part II*, *LNCS*, vol. 4488, pp. 904–911, Springer-Verlag, Berlin, Heidelberg, 2007. `https://doi.org/10.1007/978-3-540-72586-2_126`.

[9] Dreżewski R., Siwik L.: Agent-based multi-objective evolutionary algorithm with sexual selection. In: *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2008, June 1–6, 2008, Hong Kong, China*, IEEE, 2008. `https://doi.org/10.1109/CEC.2008.4631296`.

[10] Dreżewski R., Siwik L.: A Review of Agent-Based Co-Evolutionary Algorithms for Multi-Objective Optimization. In: Tenne Y., Goh C.K. (eds.), *Computational Intelligence in Optimization. Application and Implementations*, Springer-Verlag, Berlin, Heidelberg, 2010. `https://doi.org/10.1007/978-3-642-12775-5_8`.

[11] Eiben A.E., Smith J.E.: *Introduction to Evolutionary Computing*, 2nd edition, Springer Verlag, 2015.

[12] Ficici S.G., Pollack J.B.: Pareto Optimality in Coevolutionary Learning. In: *Advances in Artificial Life: 6th European Conference (ECAL 2011)*, vol. 2159, pp. 316–325, Springer-Varlag, 2001.

[13] Jong De K.A.: *Evolutionary Computation – A Unified Approach*, MIT Press, 2016.

[14] Khaos U.d.M.: jMetal. `http://jmetal.sourceforge.net/`.

[15] Kisiel-Dorohinicki M.: *Agentowe architektury populacyjnych systemów inteligencji obliczeniowej*, Wydawnictwa AGH, Kraków, 2013.

[16] Kisiel-Dorohinicki M., Socha K.: Crowding Factor In Evolutionary Multi-Agent System For Multiobjective Optimization. In: *Proceedings of International Conference on Artificial Intelligence*, CSREA Press, 2001.

[17] Schaefer R., Kołodziej J.: Genetic search reinforced by the population hierarchy. In: *Proceedings of the Seventh Workshop on Foundations of Genetic Algorithms, Torremolinos, Spain, September 2–4, 2002*, 2002.

[18] Siwik L., Kisiel-Dorohinicki M.: Elitism in agent-based evolutionary multiobjective optimization, *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, vol. 9, pp. 41–48, 2005.

[19] Siwik L., Natanek S.: Solving Constrained Multi-Criteria Optimization Tasks Using Elitist Evolutionary Multi-Agent System. In: *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pp. 3358–3365, 2008. `https://doi.org/10.1109/CEC.2008.4631252`.

[20] Siwik L., Sikorski P.: Efficient Constrained Evolutionary Multi-Agent System for Multi-objective Optimization. In: *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pp. 3212–3219. 2008. `https://doi.org/10.1109/CEC.2008.4631233`.

[21] Wooldridge M.: *An Introduction to Multiagent Systems*, John Wiley & Sons, 2009.

[22] Wróbel K., Torba P., Paszyński M., Byrski A.: Evolutionary multi-agent computing in inverse problems, *Computer Science*, vol. 14(3), 2013. `https://doi.org/10.7494/csci.2013.14.3.367`.

[23] Zitzler E., Deb K., Thiele L.: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results, *Evolutionary Computation*, vol. 8(2), pp. 173–195, 2000.

[24] Zitzler E., Thiele L.: Multiobjective optimization using evolutionary algorithms – A comparative case study. In: *Parallel Problem Solving from Nature – PPSN V. Lecture Notes in Computer Science*, vol. 1498, pp. 292–301, Springer, Amsterdam, 1998.

[25] Zitzler K., Thiele L.: An evolutionary algorithm for multiobjective optimization: the strength Pareto approach. In: *TIK-Report*, vol. 43, 1998. `https://doi.org/10.3929/ethz-a-004288833`.

Affiliations

Mateusz Różański
    AGH University of Science and Technology, Krakow, Poland, rozanski.mateusz@yahoo.pl

Leszek Siwik
    AGH University of Science and Technology, Krakow, Poland, siwik@agh.edu.pl