

ON THE LEADER SELECTION IN THE SELF-ORGANIZING MIGRATING ALGORITHM

Lukas Tomaszek^{1,✉}, Ivan Zelinka¹, Mohammed Chadli²

¹Department of Computer Science, VSB Technical University of Ostrava, Czech Republic

²MIS Laboratory, University of Picardie Jules Verne, France

lukas.tomaszek@vsb.cz✉, ivan.zelinka@vsb.cz, mchadli@u-picardie.fr

Abstract

In this article, a novel leader selection strategy for the self-organizing migrating algorithm is introduced. This strategy replaces original AllToOne and AllToRand strategies. It is shown and statistically tested, that the new strategy outperforms the original ones. All the experiments were conducted on well known CEC 2014 benchmark functions according to the CEC competition rules and reported here.

Keywords: *self-organizing migrating algorithm, CEC 2014 benchmark, All-ToNBest strategy, swarm algorithms*

Received: 30 April 2019

Accepted: 31 May 2019

Published: 24 June 2019

1 Introduction

In technology, we often deal with optimization tasks. Companies want to improve their processes, and each process can be understood as an optimization task in which we minimize the cost, or maximize the profit. These tasks are often very complex, and sometimes it is impossible to solve them by exact methods in a reasonable amount of time. For such tasks, metaheuristics can be used [6, 14].

Metaheuristics optimization algorithms allow us to find a solution, close to the optimal one, of the optimization tasks in a reasonable amount of time. They are based on the processes in nature like swarm behavioral or Darwinian theory of evolution. There exist many algorithms, and the number is still growing. For example, we can mention genetic algorithms (GA) [4], differential evolution [13], self-organizing migrating algorithm (SOMA) [2], particle swarm optimization (PSO) [10] or firefly algorithm (FA) [16]. In this article, SOMA is used.

SOMA was introduced by Zelinka [17]. Even though it is not so popular like other algorithms, in [1] authors showed, that the SOMA reaches similar results as other algorithms like PSO or FA. Also, only a few attempts have been made to improve its performance. Most of these improvements were based on the combination with another algorithms. For example, Deep combines SOMA with GAs [5], Singh added mutation to SOMA [11, 12] or Coelho combine SOMA with the cultural algorithm [9], but all of these improvements are based on the combination with other algorithm and the standard SOMA were used. SOMA can be also used in discrete domain [3] or for multi-objective optimization [7].

For the SOMA run, a user has to set up the parameters and select a strategy, which describes how a leader will be chosen. This affects the algorithm performance. In this article, we want to merge two SOMA strategies into one and show how the new strategy performs. Specifically, we replace AllToOne and AllToRand strategy by creating new AllToNBest strategy. The new strategy will limit the decisions the user has to make without loss of algorithm performance.

2 Self-Organizing Migrating Algorithm

SOMA [2, 17, 18] belongs to the class of swarm algorithms. These algorithms mimic animals behavioral in nature to be able to find an optimal given problem solution. SOMA models the social behavior of competitive-cooperating individuals. If predators are searching for food, they are cooperating and competing, so if one predator is successful, other animals change their trajectory towards this group member. SOMA attempts to mimic such behavioral. All parts, also with all important details, about the algorithm are described below.

2.1 Parameters Settings

For the SOMA algorithm run, we have to set up the parameters and define a cost function. The cost function is a mathematical model of our optimization problem and determines the parameter called *Dimension*. Other parameters, which are listed in Table 1, must be initiated by the user.

Table 1: SOMA parameters

Parameter name	Recommended range	Remark
<i>PathLength</i>	[2, 5]	Controlling parameter
<i>Step</i>	[0.11, <i>PathLength</i>]	Controlling parameter
<i>PRT</i>	[0, 1]	Controlling parameter
<i>Dimension</i>	Given by problem	Number of arguments in cost function
<i>PopSize</i>	[10, up to user]	Controlling parameter
<i>ML</i>	[10, up to user]	Stopping parameter

The SOMA works with a population of *PopSize* individuals. At each migration cycle, all individuals jump towards the leader. Size of each jump is given by parameter *Step*, and the maximal travel distance is given by parameter *PathLength*. Next parameter, called *PRT*, specifies the individual direction movement. It tells us whether the individual will jump straight toward the leader, or also to the sides. Last parameter *ML* defines the number of SOMA migration loops. Based on this parameter, the algorithm stops its run. This parameter may be replaced by the number of function evaluations or other stopping criteria.

2.2 Initial Population Creation

During the initialization process, the population with *PopSize* individuals is created. Each individual is represented as a vector $\vec{x}_i^{ML} = (x_{i,1}^{ML}, x_{i,2}^{ML}, \dots, x_{i,Dimension}^{ML})$, where \vec{x}_i^{ML} represents individual *i* in migration loop *ML* and the vector component is initialized according to equation

$$x_{i,j}^0 = x_{min,j} + rand_{i,j} \cdot (x_{max,j} - x_{min,j}), \quad (1)$$

where $x_{i,j}^0$ is parameter *j* of individual *i* at the beginning of the algorithm. $x_{min,j}$ and $x_{max,j}$ represent the minimal respectively maximal value of parameter *j* and $rand_{i,j} \in [0, 1]$ is a uniformly distributed random number lying between 0 and 1.

2.3 Migration Loops Performance

During the migration loops, we are selecting the individuals one by one and performing jumps with them. When the individual is selected, it jumps towards a leader according to equation

$$\vec{y}_i^t = \vec{x}_i^{ML} + (\vec{x}_L^{ML} - \vec{x}_i^{ML}) \cdot t \cdot Step \cdot \vec{PRT}_t, \quad (2)$$

where \vec{x}_i^{ML} is a jumping individual, \vec{x}_L^{ML} is a leader, *t* is an integer step number from 1 to *PathLength/Step*, *Step* is a controlling parameter, \vec{PRT}_t is a perturbation vector generated before each jump (explained later), and \vec{y}_i^t represents the position, where jumping individual *i* jumps in step *t*.

In total, each individual performs *PathLength/Step* jumps. At each position \vec{y}_i^t the cost function is evaluated, and greedy selection is performed between original position and all visited positions. In case of minimization problem, the individual is moved for next migration loop to the position \vec{x}_i^{ML+1} according to equation

$$\vec{x}_i^{ML+1} = \vec{y}_i^t \text{ where } f(\vec{y}_i^t) = \min(f(\vec{y}_i^0), \dots, f(\vec{y}_i^{PathLength/Step})), \quad (3)$$

where $f()$ is a cost function. Note that $\vec{y}_i^0 = \vec{x}_i^{ML}$, but we do not want to evaluate the same position twice, so in previous equation the *t* starts from 1.

2.3.1 Step and PathLength Parameters

Step and *PathLength* parameters define the size and number of jumps. Both parameters represent the relative distance between a jumping individual and a leader. With a small *Step* value and a high *PathLength* value, jumping individual performs many steps, and the algorithm searches the space intensely, but we may need many repetitions for finding an optimal solution. On the other hand, with high *Step* value and low *PathLength* value, jumping individuals perform a few steps, the algorithm finishes the search quickly, but it may get stuck in local optima.

Parameters setup is problem dependent. Parameters which perform well on one cost function, may not work on another (no free lunch theorem [15]). The parameters setup is up to the user, but the multiple of the step parameter must not be equal to 1. In this case, a jumping individual may jump to the leader position, and the same individuals may appear in the population. Our recommendation is to set parameters as follow: *PathLength* = 3, and *Step* = 0.11, 0.21 or 0.3.

2.3.2 PRT Vector Creation

A perturbation vector $\vec{PRT} = (PRT_0, PRT_1, \dots, PRT_{Dimension})$ determines whether the individual jumps straight to a leader or also to the sides. This vector is created, based on the parameter PRT , before an individual jump according to equation

$$PRT_i = \begin{cases} 1 & \text{if } rand_i < PRT, \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

where $rand_i$ is uniformly distributed random number lying between 0 and 1, and PRT is a controlling parameter.

The perturbation vector must contain at least one value equals to 1 because if all vector components are equal to 0, the jumping individual would not make a jump, but evaluate the same position twice. If the perturbation vector contains only zeros, it is regenerated. Our recommendation is to set parameters $PRT = 0.1, 0.2$ or 0.3 .

2.3.3 Leader Selection

Another important decision, a user must do, is a leader selection strategy. Let us name four canonical strategies presented by Zelinka [18].

- **AllToOne** - in this strategy, the leader is the best individual in the population (the individual with the lowest cost function value in case of the minimization problem). Note that, the leader itself do not make jumps.
- **AllToRand** - in this strategy, the leader is selected randomly for each jumping individual, so one jumping individual jumps to one randomly selected leader and other jumping individual jumps to another randomly selected leader. The chosen leader must not be the same as the jumping individual.
- **AllToAll** - in this strategy, an individual jumps to all individuals. This means, that firstly the jumping individual performs jumps to one individual, after to a second individual, and so on. The position is updated after all jumps to all individual.
- **AllToAllAdaptive** - this strategy is similar to AllToAll strategy, but after each jumping sequence, the individual changes its position and performs next jumps from the new position, not from the original position.

2.3.4 Constraints

Last important part, we want to mention, is constraints handling. Since each dimension of our cost function has defined boundaries, the individuals may jump out of searching space, so we have to determine how to deal with such positions. Some researchers use the equation $x_{i,j} = \min(x_{max,j}, \max(x_{min,j}, x_{i,j}))$ to restrain solution in the borders [19], but in SOMA this may cause same solution creation, so if a newly created solution is outside the borders, we add or subtract the size of the given dimension. We can express it by equation $x_{i,j} = x_{i,j} + k \cdot (x_{max,j} - x_{min,j})$, where k is a suitable integer value, which causes that the solution will be in the searching range.

3 Proposed Self-Organizing Migrating Algorithm

Before we start the SOMA algorithm, we have to set up the parameters and chose how the leader will be selected. As we have mentioned above, the parameters setup is problem dependent, so it influences the algorithm performance. Also, the chosen strategy has a similar effect. AllToOne strategy may find the search optima fast, but also it may converge to local optima. On the other hand, AllToRand strategy may avoid stacking in local optima, but also the algorithm may run a relatively long time. In the proposed change, we want to eliminate the AllToOne and AllToRand strategy by replacing it with new AllToNBest strategy.

New, AllToNBest, strategy generalizes the old AllToOne and AllToRand strategy into one. When we are selecting a leader, we order the individuals in the population according to their cost value and pick a random individual from N best individuals. If $N = 1$, SOMA will perform like the old AllToOne strategy, and if the $N = PopSize$, SOMA will perform like the old AllToRand strategy.

4 Experiments

In the experimental part, we want to find the optimal N for the SOMA AllToNBest strategy run, and compare it with the old AllToOne and AllToRand strategies. In the following part, you can find the details about the experiments. How we set the parameters, what cost functions we used and the results.

Table 2: Used SOMA parameters

Parameter name	Parameters 1	Parameters 2	Parameters 3	Parameters 4
<i>PathLength</i>	3	3	3	3
<i>Step</i>	0.11	0.11	0.3	0.3
<i>PRT</i>	0.1	0.3	0.1	0.3
<i>PopSize</i>	$Dimension + 20$	$Dimension + 20$	$Dimension + 20$	$Dimension + 20$
<i>MaxFES</i>	$Dimension \cdot 10^4$	$Dimension \cdot 10^4$	$Dimension \cdot 10^4$	$Dimension \cdot 10^4$

4.1 Cost Functions

We chose the CEC 2014 Benchmark functions for the experiments [8]. This benchmark set consists of 30 test functions with different properties. Functions are divided into four categories - unimodal, multimodal, hybrid and composition functions. The searching range of all functions is $[-100, 100]^{Dimension}$. According to the CEC competition rules, the algorithm should not do more than $Dimension \cdot 10^4$ function evaluation, and the error lower than 10^{-8} is taken as 0. We followed these rules, so our stopping criteria were given by maximal function evaluations (MaxFES).

4.2 Parameters

The first experimental part focuses on finding the optimal N for the SOMA AllToNBEST strategy run. These experiments were conducted on the $Dimension = 10$, and four different parameters showed in Table 2. For each of these parameters, we tried to use a different number of best individuals from which the leader was selected. The N was set to 3, 5, 7 and 10.

The second experiment focuses on the comparison of the AllToNBEST strategy with the AllToOne and AllToRand strategies. The experiments were conducted on $Dimension = 10$ and 30. And the parameters were set according to Table 2. The results from the previous experiment showed, that the optimal N is approximately $0.25 \cdot PopSize$, so for the $Dimension = 10$ we set $N = 7$, and for $Dimension = 30$ we set $N = 12$.

4.3 Results

In the following Tables 3 and 4, we can see the results of the first experiment. This experiment should find the optimal number of best individuals from which a leader is selected. We can see the means of errors from 51 independent runs on all functions and for 4 different parameters setup. Also, at the bottom of each table, there is an average Friedman rank. We also performed the Friedman rank test, but the results are not significant at the level $\alpha = 0.05$.

In the next Tables 5 - 8, we can see the results of the second experiment. This experiment should compare the SOMA strategies, namely AllToOne, AllToRand, and novel AllToNBEST strategy. In the tables, we can see the means of errors from 51 independent runs and also the average Friedman rank. After, we performed a Friedman rank test with significance level $\alpha = 0.05$. Friedman test showed that in all except one, there is a significant difference between strategies, so the Nemenyi post-hoc test was applied to the results. The comparison between AllToNBEST and AllToOne and comparison between AllToNBEST and AllToRand can be seen in Table 9. Mark '+++' means, that the AllToNBEST strategy was significantly better.

5 Discussion and Conclusion

Before the run of SOMA, a user has to define parameters and chose the strategy for leader selection. In this article, we presented a novel leader selection strategy called AllToNBEST, which replaces the original AllToOne and AllToRand strategies.

In the first experiment, we showed that the N parameter should be set approximately $0.25 \cdot PopSize$. Even though the difference was not significant the average Friedman rank for this value was the lowest in two cases, and second lowest in another two cases.

After, in the second experiment, we compared the new strategy, with original strategies. It was shown that the AllToNBEST strategy provides similar or better results than original strategies. Also, the statistical tests were used, to showed that the difference is significant in many cases.

One can objectify, that even though the new strategy reduces the leader selection strategy, it added a new parameter into the SOMA, but we showed how to set up the parameter to get similar or better results. Also, in the future, this parameter may be adaptive, to increase the performance of SOMA even more.

Acknowledgement: This work was supported by the ESF in "Science without borders" project, reg. nr. CZ.02.2.69/0.0/0.0/16.027/0008463 within the Operational Programme Research, Development and Education,

Table 3: Means from 51 independent runs for *Dimension* = 10, AllToNBest strategy and different number of best individuals from which a leader is selected

	Parameters 1 (see Table 2)				Parameters 2 (see Table 2)			
	N=3	N = 5	N = 7	N = 10	N=3	N = 5	N = 7	N = 10
1	3.77E+03	4.34E+03	4.95E+03	4.92E+03	1.14E+02	9.59E+01	1.36E+02	1.12E+02
2	8.68E-01	1.94E+00	1.17E+01	2.35E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3	5.81E-01	5.70E+00	5.92E+00	8.23E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
4	4.47E+00	1.57E+00	1.92E+00	2.49E+00	9.20E+00	1.16E+01	9.22E+00	9.99E+00
5	1.89E+01	1.80E+01	1.91E+01	1.74E+01	1.96E+01	1.87E+01	1.90E+01	1.84E+01
6	2.15E-01	1.70E-01	1.91E-01	1.99E-01	8.80E-02	8.06E-02	6.53E-04	6.41E-06
7	4.40E-02	4.11E-02	3.80E-02	2.89E-02	4.87E-02	3.78E-02	2.15E-02	2.05E-02
8	1.95E-02	1.95E-02	0.00E+00	0.00E+00	6.44E-01	3.90E-01	2.54E-01	1.17E-01
9	4.22E+00	4.38E+00	4.13E+00	4.09E+00	3.39E+00	3.07E+00	2.46E+00	2.16E+00
10	5.51E-01	3.15E-01	1.79E-01	1.46E-01	3.11E+00	1.92E+00	1.38E+00	1.55E+00
11	1.45E+02	1.34E+02	1.13E+02	1.26E+02	6.30E+01	6.54E+01	6.11E+01	7.30E+01
12	2.72E-01	2.48E-01	2.48E-01	2.46E-01	2.24E-01	2.75E-01	2.86E-01	2.93E-01
13	1.72E-01	1.75E-01	1.80E-01	1.87E-01	1.30E-01	1.23E-01	1.29E-01	1.29E-01
14	1.47E-01	1.39E-01	1.54E-01	1.53E-01	1.25E-01	1.35E-01	1.35E-01	1.34E-01
15	9.55E-01	9.19E-01	9.22E-01	9.13E-01	8.34E-01	8.29E-01	8.47E-01	8.81E-01
16	1.67E+00	1.68E+00	1.73E+00	1.74E+00	1.15E+00	1.46E+00	1.46E+00	1.69E+00
17	1.07E+04	1.03E+04	1.37E+04	1.63E+04	7.87E+01	6.31E+01	5.66E+01	8.29E+01
18	5.13E+01	1.26E+02	1.19E+02	3.88E+02	1.89E+00	1.46E+00	1.32E+00	9.99E-01
19	1.56E-01	1.29E-01	1.42E-01	1.45E-01	1.36E-01	7.65E-02	5.22E-02	6.29E-02
20	1.93E-01	1.81E-01	1.36E-01	1.51E-01	4.17E-01	3.10E-01	1.50E-01	1.33E-01
21	2.65E+02	3.41E+02	9.21E+01	1.02E+02	4.18E+00	4.34E+00	3.95E+00	8.32E-01
22	2.69E-01	1.95E-01	2.30E-01	1.41E-01	1.32E+00	3.00E+00	5.85E-01	1.92E-01
23	3.29E+02	3.29E+02	3.28E+02	3.22E+02	3.29E+02	3.29E+02	3.29E+02	3.29E+02
24	1.14E+02	1.14E+02	1.14E+02	1.14E+02	1.11E+02	1.10E+02	1.10E+02	1.10E+02
25	1.30E+02	1.31E+02	1.30E+02	1.33E+02	1.26E+02	1.25E+02	1.23E+02	1.28E+02
26	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02
27	3.27E+01	8.12E+01	6.15E+01	5.01E+01	3.42E+01	8.50E+01	3.22E+01	4.33E+01
28	3.69E+02	3.71E+02	3.69E+02	3.62E+02	3.74E+02	3.76E+02	3.71E+02	3.67E+02
29	3.38E+02	3.21E+02	3.21E+02	3.08E+02	2.56E+02	2.59E+02	2.61E+02	2.68E+02
30	6.37E+02	6.51E+02	6.44E+02	6.57E+02	6.05E+02	5.60E+02	6.05E+02	5.85E+02
Avg rank	2.5	2.3	2.2	2.3	2.7	2.4	1.9	2.0

Table 4: Means from 51 independent runs for *Dimension* = 10, AllToNBest strategy and different number of best individuals from which a leader is selected

	Parameters 3 (see Table 2)				Parameters 4 (see Table 2)			
	N=3	N = 5	N = 7	N = 10	N=3	N = 5	N = 7	N = 10
1	3.78E+03	3.54E+03	4.25E+03	5.92E+03	1.03E+02	7.95E+01	1.25E+02	1.80E+02
2	1.57E+00	1.51E+00	5.86E+00	1.10E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3	1.04E+00	6.94E-01	4.65E+00	6.17E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
4	2.55E+00	5.77E+00	4.24E+00	1.32E+00	1.07E+01	1.10E+01	6.61E+00	1.12E+01
5	1.86E+01	1.94E+01	1.90E+01	1.85E+01	1.75E+01	1.91E+01	1.89E+01	1.93E+01
6	1.82E-01	1.64E-01	1.89E-01	1.87E-01	1.19E-01	9.17E-03	6.49E-03	2.58E-03
7	4.59E-02	4.56E-02	3.32E-02	2.81E-02	5.51E-02	4.02E-02	2.32E-02	1.97E-02
8	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.07E-01	2.73E-01	1.76E-01	9.75E-02
9	4.23E+00	4.36E+00	3.93E+00	4.03E+00	3.61E+00	2.64E+00	2.43E+00	2.83E+00
10	6.66E-01	7.20E-01	1.68E-01	1.51E-01	2.78E+00	1.35E+00	9.15E-01	9.46E-01
11	1.67E+02	1.55E+02	1.28E+02	1.18E+02	4.94E+01	3.64E+01	4.78E+01	6.43E+01
12	2.83E-01	2.64E-01	2.62E-01	2.40E-01	2.41E-01	2.62E-01	2.73E-01	2.95E-01
13	1.71E-01	1.90E-01	1.88E-01	1.69E-01	1.27E-01	1.24E-01	1.39E-01	1.39E-01
14	1.39E-01	1.47E-01	1.46E-01	1.57E-01	1.22E-01	1.26E-01	1.44E-01	1.51E-01
15	9.40E-01	9.83E-01	8.70E-01	8.55E-01	8.02E-01	8.54E-01	8.56E-01	8.77E-01
16	1.69E+00	1.71E+00	1.64E+00	1.67E+00	1.22E+00	1.37E+00	1.54E+00	1.53E+00
17	1.12E+04	1.24E+04	1.49E+04	1.41E+04	5.19E+01	4.71E+01	6.74E+01	7.61E+01
18	6.34E+01	7.80E+01	3.67E+02	4.32E+02	1.70E+00	1.29E+00	1.34E+00	1.12E+00
19	1.12E-01	1.06E-01	1.44E-01	1.43E-01	1.05E-01	5.33E-02	6.72E-02	4.16E-02
20	2.39E-01	2.23E-01	1.48E-01	1.78E-01	4.50E-01	2.94E-01	1.94E-01	2.65E-01
21	4.11E+02	4.68E+02	2.23E+02	2.24E+02	3.86E+00	2.93E+00	3.00E+00	1.17E+00
22	8.95E-01	9.29E-01	1.47E-01	1.58E-01	3.73E-01	3.44E-01	2.32E-01	5.41E-01
23	3.29E+02	3.28E+02	3.25E+02	3.25E+02	3.29E+02	3.29E+02	3.28E+02	3.26E+02
24	1.14E+02	1.15E+02	1.14E+02	1.14E+02	1.12E+02	1.11E+02	1.10E+02	1.10E+02
25	1.29E+02	1.30E+02	1.31E+02	1.32E+02	1.27E+02	1.21E+02	1.25E+02	1.26E+02
26	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02
27	8.02E+01	3.35E+01	8.92E+01	5.70E+01	5.18E+01	5.80E+01	8.86E+01	2.52E+01
28	3.73E+02	3.70E+02	3.68E+02	3.71E+02	3.81E+02	3.69E+02	3.73E+02	3.66E+02
29	3.41E+02	3.41E+02	3.16E+02	3.08E+02	2.59E+02	2.66E+02	2.73E+02	2.65E+02
30	6.51E+02	6.54E+02	6.23E+02	6.53E+02	5.82E+02	5.68E+02	5.91E+02	5.84E+02
Avg rank	2.4	2.7	2.2	2.1	2.6	2.1	2.3	2.3

Table 5: Means from 51 independent runs for *Dimension* = 10 and different leader selection strategies. As *N* for AllToNBest strategy we selected 7.

	Parameters 1 (see Table 2)			Parameters 2 (see Table 2)		
	AllToOne	AllToRand	AllToNBest	AllToOne	AllToRand	AllToNBest
1	5.011E+03	1.545E+04	4.950E+03	2.012E+02	5.014E+02	1.365E+02
2	7.370E-01	4.175E+01	1.166E+01	0.000E+00	4.979E-07	0.000E+00
3	4.385E-01	8.395E+01	5.925E+00	0.000E+00	7.939E-09	0.000E+00
4	2.614E+00	9.958E-01	1.921E+00	1.153E+01	2.030E+00	9.224E+00
5	1.857E+01	1.824E+01	1.909E+01	1.935E+01	1.878E+01	1.901E+01
6	2.425E-01	8.624E-01	1.913E-01	9.875E-02	4.521E-03	6.533E-04
7	8.294E-02	3.021E-02	3.804E-02	6.907E-02	2.696E-02	2.148E-02
8	1.951E-02	0.000E+00	0.000E+00	7.413E-01	1.951E-02	2.536E-01
9	5.158E+00	4.925E+00	4.131E+00	3.965E+00	4.203E+00	2.456E+00
10	6.339E-01	1.213E-01	1.794E-01	3.734E+00	3.047E-01	1.379E+00
11	2.287E+02	1.636E+02	1.126E+02	8.085E+01	1.626E+02	6.110E+01
12	3.481E-01	2.390E-01	2.480E-01	2.525E-01	3.348E-01	2.860E-01
13	2.070E-01	1.895E-01	1.803E-01	1.292E-01	1.614E-01	1.288E-01
14	1.593E-01	1.830E-01	1.538E-01	1.347E-01	1.776E-01	1.354E-01
15	1.211E+00	9.301E-01	9.217E-01	9.192E-01	9.679E-01	8.465E-01
16	1.887E+00	1.948E+00	1.729E+00	1.501E+00	1.923E+00	1.457E+00
17	7.119E+03	3.114E+04	1.370E+04	9.768E+01	1.589E+02	5.662E+01
18	3.402E+01	6.254E+02	1.188E+02	1.760E+00	1.213E+00	1.322E+00
19	2.014E-01	2.326E-01	1.418E-01	2.080E-01	1.067E-01	5.225E-02
20	2.514E-01	2.329E-01	1.360E-01	5.845E-01	1.995E-01	1.504E-01
21	6.069E+01	8.089E+02	9.212E+01	6.530E+00	4.191E-01	3.950E+00
22	9.277E-01	2.099E-01	2.299E-01	2.586E+00	1.676E-01	5.851E-01
23	3.103E+02	3.166E+02	3.277E+02	3.230E+02	3.265E+02	3.295E+02
24	1.162E+02	1.155E+02	1.136E+02	1.103E+02	1.121E+02	1.098E+02
25	1.346E+02	1.358E+02	1.302E+02	1.316E+02	1.408E+02	1.225E+02
26	1.002E+02	1.002E+02	1.002E+02	1.001E+02	1.002E+02	1.001E+02
27	8.307E+01	3.971E+01	6.149E+01	2.578E+01	5.420E+01	3.223E+01
28	3.711E+02	3.773E+02	3.685E+02	3.866E+02	3.706E+02	3.715E+02
29	4.316E+02	3.229E+02	3.211E+02	2.552E+02	2.867E+02	2.607E+02
30	6.740E+02	6.272E+02	6.442E+02	5.910E+02	6.011E+02	6.047E+02
Avg rank	2.3	2.1	1.5	2.1	2.3	1.5

Table 6: Means from 51 independent runs for *Dimension* = 10 and different leader selection strategies. As *N* for AllToNBest strategy we selected 7.

	Parameters 3 (see Table 2)			Parameters 4 (see Table 2)		
	AllToOne	AllToRand	AllToNBest	AllToOne	AllToRand	AllToNBest
1	1.124E+04	1.216E+04	4.252E+03	2.961E+01	3.229E+02	1.251E+02
2	4.082E-01	4.698E+01	5.863E+00	0.000E+00	3.133E-10	0.000E+00
3	5.618E-01	6.185E+01	4.646E+00	0.000E+00	0.000E+00	0.000E+00
4	9.873E+00	4.569E-01	4.239E+00	1.719E+01	5.141E+00	6.609E+00
5	2.001E+01	1.878E+01	1.902E+01	1.974E+01	1.901E+01	1.889E+01
6	2.744E-01	4.987E-01	1.886E-01	3.729E-01	3.943E-04	6.489E-03
7	9.463E-02	2.427E-02	3.317E-02	8.711E-02	1.359E-02	2.316E-02
8	1.924E+00	1.366E+00	0.000E+00	2.669E+00	4.682E-01	1.756E-01
9	8.525E+00	4.841E+00	3.934E+00	6.196E+00	4.519E+00	2.428E+00
10	1.453E+01	2.596E-01	1.678E-01	1.653E+01	2.474E-01	9.154E-01
11	4.067E+02	1.822E+02	1.275E+02	2.013E+02	2.074E+02	4.785E+01
12	5.049E-01	2.550E-01	2.623E-01	4.316E-01	3.543E-01	2.728E-01
13	1.898E-01	1.953E-01	1.882E-01	1.414E-01	1.608E-01	1.391E-01
14	1.788E-01	1.887E-01	1.460E-01	1.438E-01	1.835E-01	1.445E-01
15	1.403E+00	9.396E-01	8.702E-01	1.031E+00	9.776E-01	8.558E-01
16	2.097E+00	1.899E+00	1.645E+00	1.612E+00	1.972E+00	1.535E+00
17	8.103E+03	3.160E+04	1.488E+04	1.141E+02	2.244E+02	6.740E+01
18	2.898E+02	8.185E+02	3.670E+02	4.247E+00	1.241E+00	1.340E+00
19	3.171E-01	2.269E-01	1.439E-01	4.765E-01	1.511E-01	6.721E-02
20	8.331E-01	2.307E-01	1.484E-01	1.064E+00	3.911E-01	1.939E-01
21	4.707E+02	1.007E+03	2.234E+02	1.086E+01	3.822E-01	3.001E+00
22	3.035E+00	1.514E-01	1.466E-01	8.945E+00	1.288E-01	2.316E-01
23	3.293E+02	3.250E+02	3.251E+02	3.295E+02	3.295E+02	3.281E+02
24	1.169E+02	1.153E+02	1.138E+02	1.150E+02	1.113E+02	1.103E+02
25	1.369E+02	1.353E+02	1.307E+02	1.361E+02	1.409E+02	1.249E+02
26	1.002E+02	1.002E+02	1.002E+02	1.001E+02	1.002E+02	1.001E+02
27	1.081E+02	6.482E+01	8.920E+01	7.729E+01	6.680E+01	8.856E+01
28	3.743E+02	3.750E+02	3.685E+02	4.087E+02	3.682E+02	3.728E+02
29	5.457E+02	3.349E+02	3.155E+02	2.807E+02	2.912E+02	2.727E+02
30	7.267E+02	6.451E+02	6.229E+02	6.645E+02	5.980E+02	5.914E+02
Avg rank	2.5	2.1	1.3	2.4	2.0	1.4

Table 7: Means from 51 independent runs for $Dimension = 30$ and different leader selection strategies. As N for AllToNBest strategy we selected 12.

	Parameters 1 (see Table 2)			Parameters 2 (see Table 2)		
	AllToOne	AllToRand	AllToNBest	AllToOne	AllToRand	AllToNBest
1	5.371E+06	6.383E+06	4.464E+06	9.182E+05	4.346E+06	9.950E+05
2	1.264E-06	7.090E-04	4.432E-06	0.000E+00	2.530E-05	0.000E+00
3	1.043E-01	1.451E+02	4.156E+00	0.000E+00	1.466E-07	0.000E+00
4	7.757E+01	9.478E+01	8.088E+01	9.147E+01	9.100E+01	8.026E+01
5	2.038E+01	2.034E+01	2.036E+01	2.073E+01	2.073E+01	2.073E+01
6	1.368E+01	1.341E+01	1.220E+01	6.767E+00	1.756E+01	4.063E+00
7	1.352E-03	8.744E-06	9.426E-08	8.446E-03	1.099E-03	3.811E-03
8	2.731E-01	0.000E+00	1.951E-02	1.296E+01	5.658E-01	3.629E+00
9	7.360E+01	6.041E+01	5.612E+01	3.814E+01	9.083E+01	7.210E+01
10	5.668E+00	4.400E-01	1.519E+00	7.430E+01	3.063E+00	1.228E+01
11	2.990E+03	2.547E+03	2.625E+03	3.813E+03	5.074E+03	4.836E+03
12	5.301E-01	4.604E-01	4.678E-01	1.180E+00	1.243E+00	1.208E+00
13	3.285E-01	3.209E-01	3.104E-01	2.886E-01	2.643E-01	2.450E-01
14	2.415E-01	2.333E-01	2.375E-01	2.485E-01	2.425E-01	2.524E-01
15	8.731E+00	7.989E+00	7.390E+00	9.875E+00	1.063E+01	9.579E+00
16	1.042E+01	1.026E+01	1.017E+01	1.099E+01	1.176E+01	1.162E+01
17	1.637E+06	1.610E+06	1.388E+06	3.003E+05	8.090E+05	4.716E+05
18	2.932E+03	2.419E+03	1.181E+03	3.451E+03	5.024E+02	1.065E+03
19	6.613E+00	7.653E+00	6.758E+00	5.860E+00	6.681E+00	6.111E+00
20	6.361E+03	4.803E+03	4.172E+03	8.560E+01	1.434E+03	2.568E+02
21	2.202E+05	1.703E+05	1.714E+05	4.012E+04	7.782E+04	4.729E+04
22	3.374E+02	2.249E+02	2.163E+02	1.980E+02	1.488E+02	1.367E+02
23	3.152E+02	3.152E+02	3.152E+02	3.152E+02	3.152E+02	3.152E+02
24	2.249E+02	2.255E+02	2.248E+02	2.284E+02	2.243E+02	2.247E+02
25	2.079E+02	2.075E+02	2.067E+02	2.081E+02	2.102E+02	2.095E+02
26	1.003E+02	1.004E+02	1.003E+02	1.003E+02	1.003E+02	1.002E+02
27	4.074E+02	3.958E+02	4.064E+02	4.016E+02	4.029E+02	4.011E+02
28	9.554E+02	9.702E+02	9.129E+02	9.253E+02	9.725E+02	8.713E+02
29	1.662E+03	1.366E+03	1.350E+03	1.169E+03	1.516E+03	1.164E+03
30	2.349E+03	2.650E+03	2.179E+03	1.782E+03	2.291E+03	1.463E+03
Avg rank	2.5	2.0	1.4	1.8	2.3	1.6

Table 8: Means from 51 independent runs for $Dimension = 30$ and different leader selection strategies. As N for AllToNBest strategy we selected 12.

	Parameters 3 (see Table 2)			Parameters 4 (see Table 2)		
	AllToOne	AllToRand	AllToNBest	AllToOne	AllToRand	AllToNBest
1	5.759E+06	7.014E+06	4.812E+06	6.297E+05	2.065E+06	2.397E+05
2	0.000E+00	1.254E-06	0.000E+00	0.000E+00	0.000E+00	0.000E+00
3	5.692E-01	1.570E+02	4.936E+00	0.000E+00	0.000E+00	0.000E+00
4	8.185E+01	8.218E+01	7.527E+01	6.578E+01	8.661E+01	5.680E+01
5	2.040E+01	2.036E+01	2.035E+01	2.073E+01	2.072E+01	2.072E+01
6	1.330E+01	1.277E+01	1.139E+01	9.264E+00	1.120E+01	4.576E+00
7	2.607E-03	3.077E-09	0.000E+00	1.152E-02	1.302E-03	4.534E-03
8	7.475E+00	3.698E+00	4.109E+00	1.699E+01	3.821E+00	7.731E+00
9	7.749E+01	6.134E+01	5.663E+01	4.110E+01	9.618E+01	5.056E+01
10	5.933E+01	2.548E+00	3.062E+00	1.114E+02	4.835E+01	3.704E+00
11	3.260E+03	2.691E+03	2.778E+03	4.155E+03	5.336E+03	5.174E+03
12	5.974E-01	4.700E-01	4.871E-01	1.284E+00	1.252E+00	1.259E+00
13	3.186E-01	3.104E-01	3.155E-01	2.933E-01	2.555E-01	2.461E-01
14	2.375E-01	2.441E-01	2.434E-01	2.549E-01	2.364E-01	2.434E-01
15	8.995E+00	7.824E+00	7.456E+00	8.310E+00	1.135E+01	9.564E+00
16	1.041E+01	1.022E+01	1.019E+01	1.105E+01	1.172E+01	1.155E+01
17	1.833E+06	1.871E+06	1.567E+06	2.799E+05	7.841E+05	4.613E+05
18	2.397E+03	9.187E+02	1.185E+03	2.689E+03	1.779E+02	7.687E+02
19	6.143E+00	7.194E+00	6.363E+00	1.063E+01	6.438E+00	5.185E+00
20	6.572E+03	5.427E+03	5.164E+03	8.318E+01	1.449E+03	2.064E+02
21	3.171E+05	2.093E+05	2.193E+05	4.631E+04	9.032E+04	6.558E+04
22	4.442E+02	2.319E+02	2.411E+02	2.592E+02	1.692E+02	1.664E+02
23	3.152E+02	3.152E+02	3.152E+02	3.152E+02	3.152E+02	3.152E+02
24	2.249E+02	2.252E+02	2.249E+02	2.305E+02	2.243E+02	2.252E+02
25	2.075E+02	2.071E+02	2.067E+02	2.095E+02	2.083E+02	2.099E+02
26	1.003E+02	1.004E+02	1.003E+02	1.003E+02	1.003E+02	1.003E+02
27	4.077E+02	4.126E+02	4.069E+02	4.078E+02	3.986E+02	4.021E+02
28	9.299E+02	9.499E+02	9.049E+02	9.613E+02	8.748E+02	8.798E+02
29	1.760E+03	1.385E+03	1.393E+03	1.065E+03	1.392E+03	1.071E+03
30	2.334E+03	2.834E+03	2.381E+03	2.228E+03	1.910E+03	1.691E+03
Avg rank	2.3	2.1	1.4	2.1	1.9	1.6

Table 9: Results from comparison of AllToNBest strategy using Nemenyi post-hoc and significance level $\alpha = 0.05$

		AllToOne				
SOMA	AllToNBest	Dimension = 10	Parameters 1	Parameters 2	Parameters 3	Parameters 4
		Dimension = 30	+++	=	+++	+++
			+++	=	+++	=
		AllToRand				
SOMA	AllToNBest	Dimension = 10	Parameters 1	Parameters 2	Parameters 3	Parameters 4
		Dimension = 30	=	+++	+++	=
			=	+++	=	=

and by SGS SP2019/137 VSB-TU Ostrava, during the research internship at MIS-UPJV Amiens France.

References

- [1] Bujok, P., Tvrđík, J., Poláková, R. 2019. Comparison of nature-inspired population-based algorithms on continuous optimisation problems. *Swarm and Evolutionary Computation*, In Press. DOI: 10.1016/j.swevo.2019.01.006
- [2] Davendra, D. and Zelinka, I. 2016. *Self-Organizing Migrating Algorithm: Methodology and Implementation*. Springer International Publishing. DOI: 10.1007/978-3-319-28161-2
- [3] Davendra, D., Zelinka, I., Bialic-Davendra, M., Senkerik, R., and Jasek, R. 2013. Discrete self-organising migrating algorithm for flow-shop scheduling with no-wait makespan. *Mathematical and Computer Modelling* 57, 1–2, pp. 100–110.
- [4] Davis, L. 1991. *Handbook of genetic algorithms*. Van Nostrand Reinhold, New York, USA.
- [5] Deep, K. et al. 2008. A self-organizing migrating genetic algorithm for constrained optimization. *Applied Mathematics and Computation* 198, 1, pp. 237–250.
- [6] Glover, F. W. and Kochenberger, G. A. 2006. *Handbook of metaheuristics*. Springer Science & Business Media.
- [7] Kadlec, P. and Raida, Z. 2011. A Novel Multi-Objective Self-Organizing Migrating Algorithm. *Radioengineering* 20, 4, pp. 804–816.
- [8] Liang, J. J., Qu, B. Y., and Suganthan, P. N. 2013. *Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization*. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore.
- [9] Coelho, L. S. and Mariani, V. C. 2010. An efficient cultural self-organizing migrating strategy for economic dispatch optimization with valve-point effect. *Energy Conversion and Management* 51, 12, pp. 2580–2587.
- [10] Shi, Y. and Eberhart, R. 1998. A modified particle swarm optimizer. In *Evolutionary Computation Proceedings (1998)*. IEEE, pp. 69–73. DOI: 10.1109/ICEC.1998.699146
- [11] Singh, D. and Agrawal, S. 2014. Hybridization of self organizing migrating algorithm with mutation for global optimization. In *Proceedings of the international conference on mathematical sciences (ICMS)*. Elsevier, pp. 605–609.
- [12] Singh, D. and Agrawal, S. 2015. Hybridization of self organizing migrating algorithm with quadratic approximation and non uniform mutation for function optimization. In *Proceedings of Fourth International Conference on Soft Computing for Problem Solving*. Springer, pp. 373–387. DOI: 10.1007/978-81-322-2217-0_32
- [13] Storn, R. and Price, K. 1997. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization* 11, 4, pp. 341–359.
- [14] Talbi, E.-G. 2009. *Metaheuristics: from design to implementation*. John Wiley & Sons.
- [15] Wolpert, D. H., Macready, W. G. et al. 1997. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation* 1, 1, pp. 67–82.
- [16] Yang, X.-S. 2010. Firefly algorithm, stochastic test functions and design optimisation. arXiv:1003.1409. Retrieved from <https://arxiv.org/abs/1003.1409>
- [17] Zelinka, I. 2004. SOMA – self-organizing migrating algorithm. In *New optimization techniques in engineering*. Springer, pp. 167–217. DOI: 10.1007/978-3-540-39930-8_7
- [18] Zelinka, I. 2016. SOMA – Self-organizing Migrating Algorithm. In *Self-Organizing Migrating Algorithm*. Springer, pp. 3–49.
- [19] Zhang, H., Li, H., Tam, C. M. 2006. Particle swarm optimization for resource-constrained project scheduling. *International Journal of Project Management* 24, 1, pp. 83–92.