BORDER STRATEGIES OF THE BISON ALGORITHM

Anezka Kazikova, Zuzana Kominkova Oplatkova, Michal Pluhacek and Roman Senkerik
Faculty of Applied Informatics
Tomas Bata University in Zlin
T.G. Masaryka 5555, 760 01 Zlin, Czech Republic
E-mail: {kazikova, oplatkova, pluhacek, senkerik}@utb.cz

KEYWORDS

Bison Algorithm, boundary strategy, hypersphere.

ABSTRACT

The Bison Algorithm is a recent swarm optimization algorithm based on bison behavior. The algorithm divides the population into two groups, simulating exploitation and exploration patterns separately. The exploration is particularly linked with crossing the search space boundaries. This paper compares several boundary violation protocols: the hypersphere, reflection, random positioning, and clipping strategies on IEEE CEC 2017 benchmark and seeks the most fitting method for the Bison Algorithm.

INTRODUCTION

Classic methods of solving complex optimization tasks often come with a pitfall of unreal time and computational requirements. Where others fail, metaheuristics rise. Spite the fact that they cannot guarantee to find the optimal solution, metaheuristics offer a fair solution in a reasonable time (Talbi 2009).

Many metaheuristics find inspiration in nature since nature has been optimizing for millions and millions of years. There are optimization algorithms built upon the ground of evolution (Back 1996), genetics (Goldberg and Holland 1988), or swarm intelligence such as the flight patterns of birds (Kennedy 2011), hunting tactics of wolves (Mirjalili, Mirjalili and Lewis 2014), bats' echolocation (Yang 2010a), and many others (Karaboga and Basturk 2007; Yang 2010b).

But whatever inspiration a metaheuristic adopts, seeking the global optimum of a multi-dimensional area is always closely linked with borders trespassing. How to deal with roamed solutions? There is a wide variety of bound handling strategies, and each can be suitable for a different kind of metaheuristic or optimization task (Helwig, Branke and Mostaghim 2013; Kadavy et al. 2017b; Kadavy et al. 2018).

The Bison Algorithm is a new swarm metaheuristic developed by Kazikova et al. (Kazikova, Pluhacek, Viktorin, et al. 2018). The algorithm divides the population into the exploiting and exploring groups. While the first one utilizes the fittest solutions, the latter systematically goes through the search space seeking new solutions.

Even though the Bison Algorithm faces the boundaries very often, no study has been done on the border handling methods; all the prior literature uses the hypersphere strategy. This article aims to find the optimal border strategy for the Bison Algorithm.

The paper is structured as follows: Section 1 describes the Bison Algorithm. Section 2 specifies selected boundary violation methods. Section 3 contains the methods, and results of the experiment. And finally, Section 4 concludes the findings and its meaning for future development.

BISON ALGORITHM WITH THE RUN SUPPORT STRATEGY

The Bison Algorithm is a recent swarm optimization algorithm to solve continuous optimization problems (Kazikova, Pluhacek, Viktorin, et al. 2018). The algorithm divides the population into two groups, each performing different characteristics of bison herds:

Algorithm 1: Pseudo code of the Bison Algorithm with the Run Support Strategy

```
Initialization:
  Objective function: f(x) = (x_1,...,x_d)
  Generate: swarming group randomly,
            running group around x_best,
            run direction vector (Eq. 4)
For every iteration i do
  Determine the swarming target:
    If f(runner_{i-1}) < f(swarmer_{i-1}) then
      target = runner_{i-1}
    Else
      target = center of the fittest (Eqs. 1,2)
  For every swarmer do
    Compute sol. candidate x_new (Eq. 3)
    If f(x_new) < f(x_old) then move to x_new
  End
  Adjust run direction vector (Eq. 5)
  For every runner do
    Move in run direction vector (Eq. 6)
  End
  Copy successful runners to swarmers
  Sort the swarming group by f(x) value
End for
```

Communications of the ECMS, Volume 33, Issue 1,
Proceedings, ©ECMS Mauro Iacono, Francesco Palmieri,
Marco Gribaudo, Massimo Ficco (Editors)
ISBN: 978-3-937436-65-4/978-3-937436-66-1(CD) ISSN 2522-2414

the first group is exploiting the search space by swarming closer to the center of the strongest individuals, while the second group systematically runs through the search space and explores new areas. When an explorer finds a promising solution, it is copied to the swarming group and replaces the center of the swarming movement for the next iteration (the last action is called the Run Support Strategy (Kazikova et al. 2019)). Algorithm 1 outlines the main loop of the Bison Algorithm.

Swarming behavior

First, the target of the swarming movement is determined as the center of several strongest solutions by default (Eqs. 1, 2). However, if the running group found a promising solution in the last iteration, the target is changed to the new solution. The swarmers then move towards the target if it improves their quality and can exceed the target by the value of the $overstep$ parameter (Eq. 3).

$$weight = (10, 20, \ldots, 10 \cdot s) \tag{1}$$

$$c = \sum_{i=1}^{s} \frac{weight_i * x_i}{\sum_{j=1}^{s} weight_i} \tag{2}$$

$$x_{i+1} = x_i + (c - x_i) \cdot random(0, v) \tag{3}$$

Where:

- s is the *elite group size* parameter,
- x_i and x_{i+1} represent the current solution and the new solution candidate,
- c is the target of the swarming movement,
- v is the *overstep* parameter.

Running behavior

The running group shifts in the run direction vector (Eq. 6), which is randomly generated during the initialization (Eq. 4) and only slightly altered after each iteration (Eq. 5).

$$r = random\left(\frac{ub-lb}{45}, \frac{ub-lb}{15}\right) \tag{4}$$

$$r = r \cdot random(0.9, 1.1) \tag{5}$$

$$x_{i+1} = x_i + r \tag{6}$$

Where:

- r is the run direction vector,
- ub and lb are the upper and the lower boundaries,
- x_{i+1} and x_i represent the current solution and its previous state.

Table 1. Parameters of the Bison Algorithm

Parameter	Description	Recommended value
Population		50
Elite group size	No. of best solutions for center computation	20
Swarm group size	No. of bison performing the swarming movement	40
Overstep	The maximum length of the swarming movement (0 = no movement; 1= max to the center)	3.5

THE BOUNDARY VIOLATION STRATEGIES

Hypersphere strategy

The hypersphere strategy (also called the periodic method) considers the upper and lower boundaries of the search space to be neighboring. When a solution crosses the borders, it appears on the other side of the dimension (Kadavy et al. 2017b). Fig. 1 shows the hypersphere strategy adopted by the Bison Algorithm.

This method allows the Bison Algorithm to keep the run direction vector only slightly altering throughout the optimization process. This was convenient, as a sudden change of the running direction is quite challenging for the running herd in real life as well. This strategy was implied in (Kazikova et al. 2018a; Kazikova et al. 2019; Kazikova et al. 2019; Kazikova et al. 2018c; Kazikova, et al. 2018b).

$$x_i' = lb + [x_i MOD(ub - lb)] \tag{7}$$

Where:

- ub and lb are upper and lower boundaries of the search space as is in all the following equations (Eqs. 7-10).

Reflection strategy

This strategy reflects the emerged solutions to the feasible space of solutions as shown in Fig. 2.

$$x_i' = \begin{cases} ub - (x_i - ub), if\ x_i > ub \\ lb + (lb - x_i), if\ x_i < lb \\ x_i, otherwise \end{cases} \tag{8}$$

Random positioning strategy

The random positioning strategy is a simple method, which generates a completely new position in the crossed dimension. The method is presented in Fig. 3.

$$x_i' = \begin{cases} rand(lb, ub), if\ x_i > ub\ or\ x_i < lb \\ x_i, otherwise \end{cases} \tag{9}$$

Clipping strategy and flipping the run direction

The original clipping strategy stops the solutions at the borders. However, this approach would only lead to trapping the exploring herd on the borders. In this case the run direction of the whole running group is flipped over in the crossed dimension (Fig. 4).

$$x_i' = \begin{cases} x_i = ub,\ run = -run,\ if\ x_i > ub \\ x_i = lb,\ run = -run,\ if\ x_i < lb \\ x_i, otherwise \end{cases} \tag{10}$$

Where

- run is the run direction vector used by the bison explorers

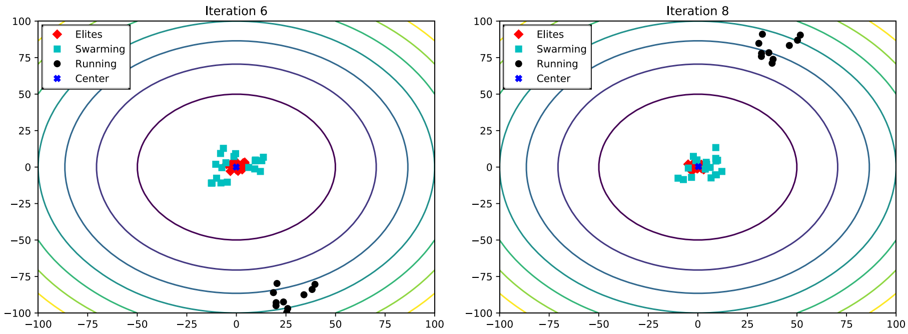

Fig. 1. Hypersphere strategy

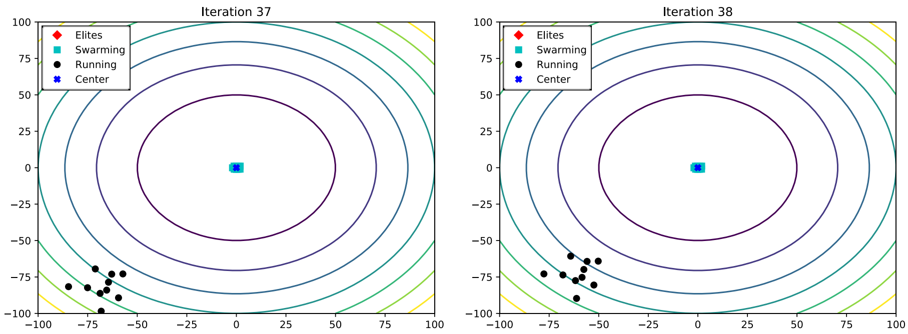

Fig. 2. Reflection strategy

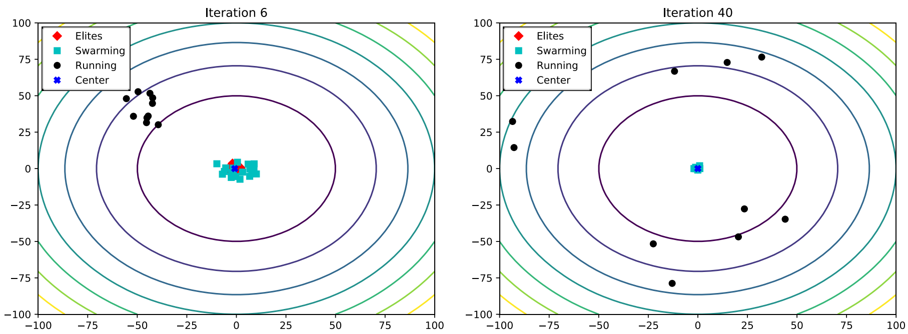

Fig. 3. Random positioning strategy

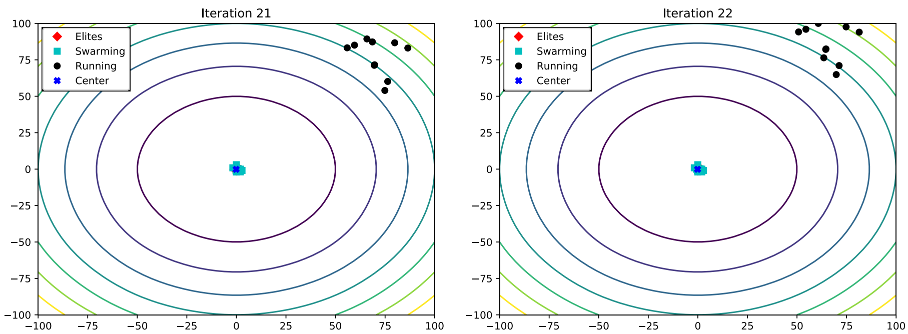

Fig. 4. Clipping strategy with the run direction vector flip

COMPARING THE BORDER STRATEGIES

The mentioned border violation strategies were compared on the 30 functions of IEEE CEC 2017 benchmark (Wu, Mallipeddi and Suganthan 2016) in 10 and 30 dimensions. We carried out 51 independent runs, each consisting of $10\,000 \cdot dimensionality$ evaluations of the objective function. The Bison Algorithm implemented the parameter configuration recommended in Table 1.

First, we compared the frequency of the border crossing. Table 2 shows the number of cases, where one strategy had a significantly lower number of roamed solutions according to the Wilcoxon Rank-Sum test (α<0.05). Fig. 5 compares the roaming quantity with the Friedman Rank test. The Friedman test is valid when $p < 0.05$, which was met in both 10 and 30 dimensions.

According to the Wilcoxon and Friedman tests, the reflection strategy provides the lowest rate of the border crossing. Reversely the highest rate was carried out by the clipping strategy.

Next, we compared the quality of the optimization given different border strategies. The final solution qualities of the CEC 2017 benchmark were compared with the Friedman rank test (Fig. 6) in 10 dimensions and in 30 dimensions (Fig. 7). Table 3 compares the quality of the two most successful strategies with the Wilcoxon rank sum test (α=0.05).

The quality-oriented results show that the best performance was carried out with the hypersphere and random positioning strategies. Interestingly, in the comparison studies performed on the Firefly Algorithm (Kadavy et al. 2018) or the PSO (Kadavy et al. 2017a; Kadavy et al. 2017b), the results were reversed: the algorithms performed the best with the reflection and clipping strategies.

Table 2. Number of functions with a significantly lower amount of getting out of bounds (Wilcoxon α=0.05)

Dimensionality	**Hypersphere**	**Reflection**	**Random**	**Clipping**	**None**
10	0	19	0	0	11
30	1	13	0	0	16

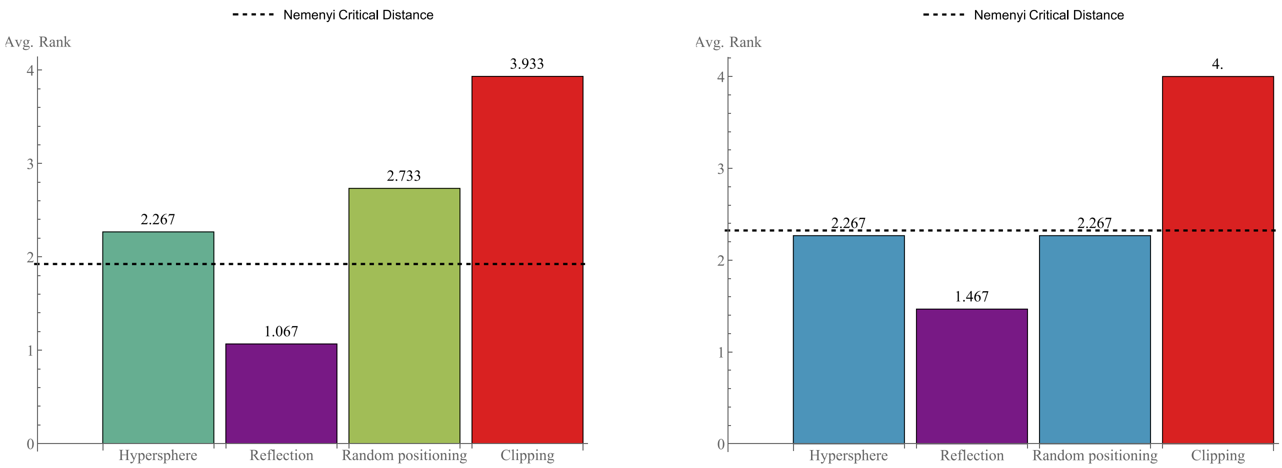

Fig. 5. Friedman rank test comparing the number of boundary violations in 10 dimensions (left) (p=$6.22\ E-35$) and 30 dimension (right) (p=$8.95\ E-22$)

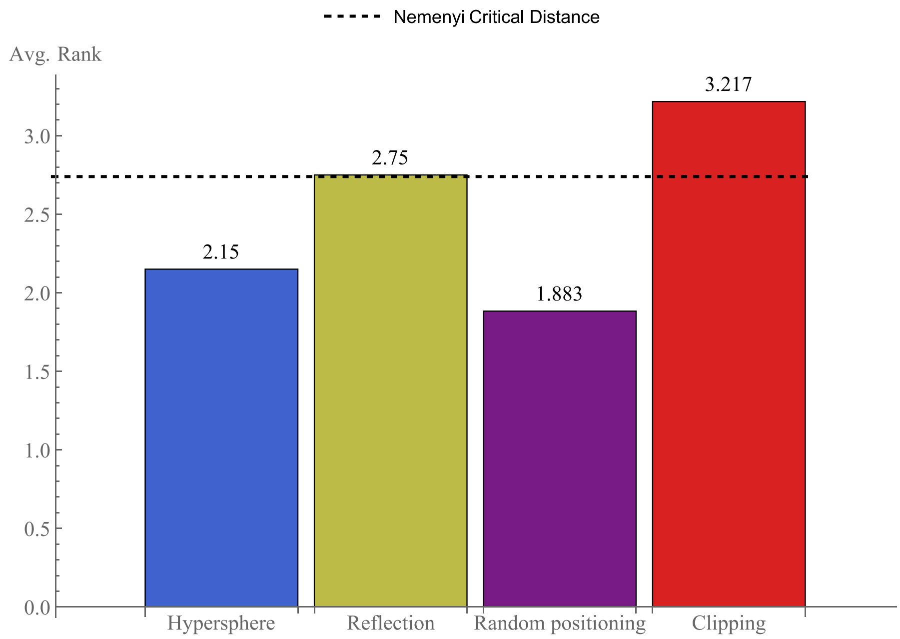

Fig. 6. Friedman rank test comparing the quality of solutions in 10 dimensions (p=6.22 E-5)

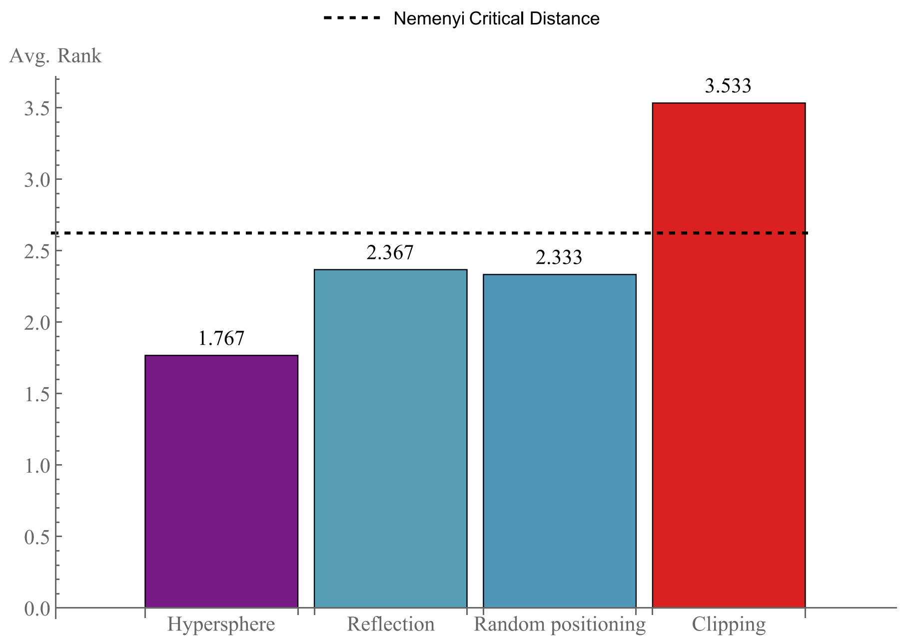

Fig. 7. Friedman rank test comparing the quality of solutions in 30 dimensions (p=1.18 E-7)

Table 3. Number of significantly better results comparing the hypersphere and random border policy (Wilcoxon α=0.05)

Dimensionality	**Hypersphere**	**Random**	**None**
10	1 (F6)	2 (F10, F25)	27
30	3 (F8, F17, F21)	1 (F3)	26

CONCLUSION

We confirmed that different metaheuristics require different border strategies. While the Particle Swarm Optimization and the Firefly Algorithm might prefer the reflection and clipping methods, the Bison Algorithm performed best with the hypersphere and random positioning strategies.

Considering the design of the algorithm, the border crossing problem is mostly encountered by the exploring group of solutions. The success of the random positioning might point to the possibility, that the accomplishments of the running bison group may not lie within the closeness of the herd. Which brings up a new question: what would happen, if the bison runners employed a larger degree of randomness?

ACKNOWLEDGMENT

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic within the National Sustainability Programme Project no. LO1303 (MSMT-7778/2014), further by the European Regional Development Fund under the Project CEBIA-Tech no. CZ.1.05/2.1.00/03.0089 and by Internal Grant Agency of Tomas Bata University under the Projects no. IGA/CebiaTech/2019/002. This work is also based upon support by COST (European Cooperation in Science & Technology) under Action CA15140, Improving Applicability of Nature-Inspired Optimisation by Joining Theory and Practice (ImAppNIO), and Action IC1406, High-Performance Modelling, and Simulation for Big Data Applications (cHiPSet). The work was further supported by resources of A.I.Lab at the Faculty of Applied Informatics, Tomas Bata University in Zlin (ailab.fai.utb.cz).

REFERENCES

Back, T. 1996. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press.

Goldberg, D.E., and J.H. Holland. 1988. "Genetic algorithms and machine learning." *Machine Learning* 3(2):95–99.

Helwig, S., J. Branke, and S. Mostaghim. 2013. "Experimental analysis of bound handling techniques in particle swarm optimization." *IEEE Transactions on Evolutionary computation* 17(2):259–271.

Kadavy, T., M. Pluhacek, A. Viktorin, and R. Senkerik. 2017a. "Comparing border strategies for roaming particles on single and Multi-Swarm PSO." In *Computer Science On-line Conference*. Springer, pp. 528–536.

Kadavy, T., M. Pluhacek, A. Viktorin, and R. Senkerik. 2018. "Comparing boundary control methods for firefly algorithm." In *International Conference on Bioinspired Methods and Their Applications*. Springer, pp. 163–173.

Kadavy, T., M. Pluhacek, A. Viktorin, and R. Senkerik. 2017b. "Hypersphere universe boundary method comparison on HCLPSO and PSO." In *International Conference on Hybrid Artificial Intelligence Systems*. Springer, pp. 173–182.

Karaboga, D., and B. Basturk. 2007. "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm." *Journal of Global Optimization* 39(3):459–471.

Kazikova, A., M. Pluhacek, T. Kadavy, and R. Senkerik. 2019. "Introducing the Run Support Strategy for the Bison Algorithm." In I. Zelinka, ed. *Recent Advances in Electrical Engineering and Related Sciences: Theory and Application*. AETA 2018. Springer. Available at: https://www.springer.com/gp/book/9783030149062 [Accessed March 2, 2019].

Kazikova, A., M. Pluhacek, and R. Senkerik. 2018a. "Performance of the Bison Algorithm on Benchmark IEEE CEC 2017." In R. Silhavy, ed. *Artificial Intelligence and Algorithms in Intelligent Systems*. Cham: Springer International Publishing, pp. 445–454. Available at: http://link.springer.com/10.1007/978-3-319-91189-2_44 [Accessed March 2, 2019].

Kazikova, A., M. Pluhacek, and R. Senkerik. 2018b. "Regarding the Behavior of Bison Runners within the Bison Algorithm." *Mendel Journal Series* 24(1):63–70.

Kazikova, A., M. Pluhacek, and R. Senkerik. 2018c. "Tuning Of The Bison Algorithm Control Parameters." In *ECMS 2018 Proceedings edited by Lars Nolle, Alexandra Burger, Christoph Tholen, Jens Werner, Jens Wellhausen*. 32nd Conference on Modelling and Simulation. ECMS, pp. 156–162. Available at: http://www.scs-europe.net/dlib/2018/2018-0156.htm [Accessed March 2, 2019].

Kazikova, A., M. Pluhacek, A. Viktorin, and R. Senkerik. 2018. "New Running Technique for the Bison Algorithm." In L. Rutkowski, ed. *ICAISC*. Springer International Publishing AG, part of Springer Nature 2018.

Kennedy, J. 2011. "Particle swarm optimization." In Encyclopedia of machine learning. Springer, pp. 760–766.

Mirjalili, S., S.M. Mirjalili, and A. Lewis. 2014. "Grey wolf optimizer." *Advances in Engineering Software* 69:46–61.

Talbi, E.-G. 2009. *Metaheuristics: from design to implementation*. Hoboken, N.J: John Wiley & Sons.

Wu, G., R. Mallipeddi, and P.N. Suganthan. 2016. "Problem Definitions and Evaluation Criteria for the CEC 2017 Competition on Constrained Real-Parameter Optimization." *National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report*.

Yang, X.-S. 2010a. "A new metaheuristic bat-inspired algorithm." In Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, pp. 65–74.

Yang, X.-S. 2010b. "Firefly algorithm, Levy flights and global optimization." In Research and development in intelligent systems XXVI. Springer, pp. 209–218.

AUTHOR BIOGRAPHIES

ANEZKA KAZIKOVA received her master's degree in Engineering Informatics from the Tomas Bata University in Zlin in 2015. She is now a Ph.D. student at the same university and researches the swarm algorithms and competitive behavior. Her e-mail is: kazikova@utb.cz. Web page of all the authors can be found at: www.ailab.fai.utb.cz.

ZUZANA KOMINKOVA OPLATKOVA received her Ph.D. degree in Technical Cybernetics in 2008 at Tomas Bata University in Zlin. She works at the same university since 2004, currently as an associate professor. This title has been given to her at Brno Technical University, Faculty of Information Technologies in June 2013. She serves as a member of journal editorial boards, member of conference international programme committees, journal reviewers and guest editor of books published by Springer. Her research interests include evolutionary computation, artificial neural networks, chaos control, classification techniques, pseudo neural networks and evolutionary symbolic regression methods. Her e-mail is: oplatkova@utb.cz.

MICHAL PLUHACEK received his Ph.D. degree in Information Technologies from the Tomas Bata University in Zlin in 2016. Currently works as a junior researcher at the Regional Research Centre CEBIA-Tech of Tomas Bata University in Zlin. His research focus includes swarm intelligence theory and applications and artificial intelligence in general. His e-mail is: pluhacek@utb.cz.

ROMAN SENKERIK received his Ph.D. degree in Technical Cybernetics from the Tomas Bata University in Zlin in 2008. He is currently an associated professor at the Tomas Bata University in Zlin, Faculty of Applied Informatics. His research interests include interdisciplinary, computational intelligence, optimization, cyber-security, theory of chaos and complexity. His e-mail is: senkerik@utb.cz.