TUNING OF THE BISON ALGORITHM CONTROL PARAMETERS

Anezka Kazikova, 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, pluhacek, senkerik}@utb.cz

KEYWORDS

Bison Algorithm, parameter study, swarm algorithm, optimization, metaheuristic.

ABSTRACT

This paper studies the dependency of the Bison Algorithm performance on the control parameter configuration. The Bison Algorithm is a new swarm algorithm based on the protection mechanisms of bison herds. It operates with two groups: the exploiting swarming group and the exploring running group. Even though that adjusting the group size parameters affects both the time requirements and the performance of the algorithm, there was no investigation of the parameter settings carried out yet. This paper describes the Bison Algorithm and then investigates the control parameters for a better understanding of their meaning and influence on the overall optimization process.

INTRODUCTION

A recent trend in the modern optimization is to exploit known biological findings. This is done by simulating instances of typical nature optimization patterns: the Darwinian evolution (Bäck 1996), genomes (Goldberg 1989) or even animal behavior processes (Yang and Deb 2013).

The swarm algorithms model the swarm intelligence – a collective behavior of large animal groups, that can make intelligent decisions without an actual leadership. Based on partial information only, the swarms manage to optimize real-life problems like finding enough food supplies, shortening their travel distance, developing an ultimate hunting strategy, reproducing or escaping predators. The simulations such as the Bees Algorithm (Rajasekhar et al. 2017), Ant Colony Optimization (Dorigo and Stütle 2004; Duan and Ying 2009), Grey Wolf Optimizer (Mirjalili et al. 2014) and Cuckoo Search (Yang and Deb 2009) have been already successfully used in the optimization field.

The Bison Algorithm is a recent swarm optimization algorithm (Kazikova et al. 2017). It simulates two of the most typical bison behavior: protecting the weak by forming a circle around them and the advantages of a running herd.

Various extensions of the Bison Algorithm have been developed since (Kazikova et al. 2018). However, there has not been done any detailed parameter study, having all the previous papers on the subject relying on an early parameter test.

This paper studies the Bison Algorithm with various parameter configurations. It simulates the movement differences and the impact on solution quality. In Section 1 of the paper, the Bison Algorithm is outlined. Section 2 describes the methods used in the parameter study. The outcomes of the experiments are presented in Section 3 and discussed in Section 4. Finally, the meanings of the findings are considered in Section 5.

BISON ALGORITHM

The Bison Algorithm was inspired by the most typical behavior patterns of bison. Bison have two distinctive protective mechanisms: forming a circle around the weak ones and almost inexhaustible running manners (Berman 2008). The algorithm implements both, as described in pseudocode Algorithm 1.

Algorithm 1: Bison Algorithm Pseudocode

```
Initialization:
  Obj. function: f(x) = (x1,...,xd)
  Generate swarming group randomly
  Generate run. group around x_best
  Generate run direction vector Eq.(4)
For every iteration do
  Compute the swarming center Eq.(1,2)
  For every swarmer do
    Compute position candidate Eq.(3)
    if f(x_new) < f(x_old) then move to x_new
  End
  Adjust run direction vector r Eq.(5)
  For every runner do
    Move in run direction vector Eq.(6)
  End
  Copy success. runners to swarmers
  Sort swarming group by f(x) value
End for
```

Proceedings 32nd European Conference on Modelling and Simulation ©ECMS Lars Nolle, Alexandra Burger, Christoph Tholen, Jens Werner, Jens Wellhausen (Editors)
ISBN: 978-0-9932440-6-3/ ISBN: 978-0-9932440-7-0 (CD)

Definition of the Bison Algorithm

The algorithm is defined by two groups, each simulating different behavior. The first group models the swarming pattern. It starts by computing the center of the fittest swarming individuals (sorted by the objective function value). This paper operates with the ranked center computation. This approach sorts the fittest individuals, giving them corresponding weights according to their solution quality (Eq. 1) and then computes the center concerning their weights (Eq. 2). A new position candidate is computed (Eq. 3) and used only if it improves the objective function value of the swarmer.

$$weight = (10, 20, \dots, 10 * s) \quad (1)$$

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

$$x_{new} = x_{old} + (c - x_{old}) \cdot rand(0, v) \quad (3)$$

Where s is the elite group size parameter, defining the number of the fittest bison to compute the center of, c is the center, v is the overstep parameter, x_i is the i^{th} solution, x_{old} is the current solution and x_{new} is the new solution candidate.

The second group simulates the running behavior. During the initialization of the algorithm, a random run direction vector r is generated (Eq. 4) and then slightly altered after every iteration (Eq. 5). The running movement always happens in the run direction vector (Eq. 6), not concerning the quality of the solution. This means that unlike the swarmers the runners move even when it threatens their quality.

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

$$r = r \cdot rand(0.9, 1.1) \quad (5)$$

$$x_{new} = x_{old} + r \quad (6)$$

Where ub and lb are the upper and lower boundaries of the search space, r is the run direction vector and x_{old}, x_{new} are the original and the new solutions.

Since the Bison Algorithm considers the search space a hypersphere, the running movement can be based on the run direction vector throughout the whole optimization process. Whenever the individuals run over the boundaries, they appear on the other side of the exceeded dimensions, exploring the search space thoroughly.

The control parameters of the Bison Algorithm are described in Table 1. The recommended value of the overstep parameter 3.5 means that the swarmers can exceed the center 2.5 times in accordance with Eq. 3. The size of the running group is defined with the help of the swarm group size (Eq. 7).

$$run\ group\ size = NP - swarm\ size \quad (7)$$

Where *NP* is the population number and *swarm size* is the swarm group size parameter.

Table 1: Bison Algorithm Control Parameters

Parameter	Description
Population NP	Population size
Elite group size	Number of the fittest solutions used for center computation
Swarm group size	Number of the swarming group solutions
Overstep v	Maximum length of the swarming movement in relation to the center 0 = no movement 1 = to the center Recommended: 3.5 - 4.1

METHODS

Two parameter scenarios were examined. The first scenario called the *complete set* consists of 12 configurations described in Table 2 in the form, that first notes the swarm group size and then the elite group size. For example, S40E20 means a population of 40 swarmers and 20 elite individuals. The *40S set* examines only the configurations with 40 swarming individuals. Other parameters were set to: $NP = 50$, $v = 3.5$.

Table 2: Tested Sets of Parameter Configurations

Complete set	S20E1, S20E10, S20E20, S30E1, S30E10, S30E20, S30E30, S40E1, S40E10, S40E20, S40E30, S40E40
40S set	S40E1, S40E10, S40E20, S40E30, S40E40

This paper investigates the influence of the control parameter configuration on 1) movement patterns, 2) performance of the algorithm, 3) computation time.

The movement is presented by a 2D simulation of the population distribution on the Rastrigin's Function (Fig. 1). The included models are: S40E1 with one bison being the sole center, S40E40 with the center computed from all the swarmers and S20E10 with many runners.

For the performance experiments, we used the first 15 functions of IEEE CEC 2017 benchmark (Awad et al. 2016), on 30 independent runs, each consisting of $10\,000 \cdot dimension$ evaluations. The solutions were compared with the Friedman rank test ($p<0.05$) in Fig. 2 and 3 for the complete set and in Fig. 4, 5 for the 40S set. Table 3 presents the Friedman P-Values. Table 4 sum the results of the two most successful configurations Wilcoxon rank-sum tests ($\alpha=0.05$) comparing the in 10 and 30 dimensions. Table 5 shows the mean solution and standard deviation of the two approaches. The mean convergences of the 40S set are shown in Fig. 6, 7 and 8.

The time requirements were compared by the Friedman rank test in Fig. 9 and 10. Table 6 shows the mean time needed for solving 10-dimensional problems with the 40S testing set. The significantly better time results according to the Wilcoxon rank-sum test are bold.

RESULTS

Movement Patterns

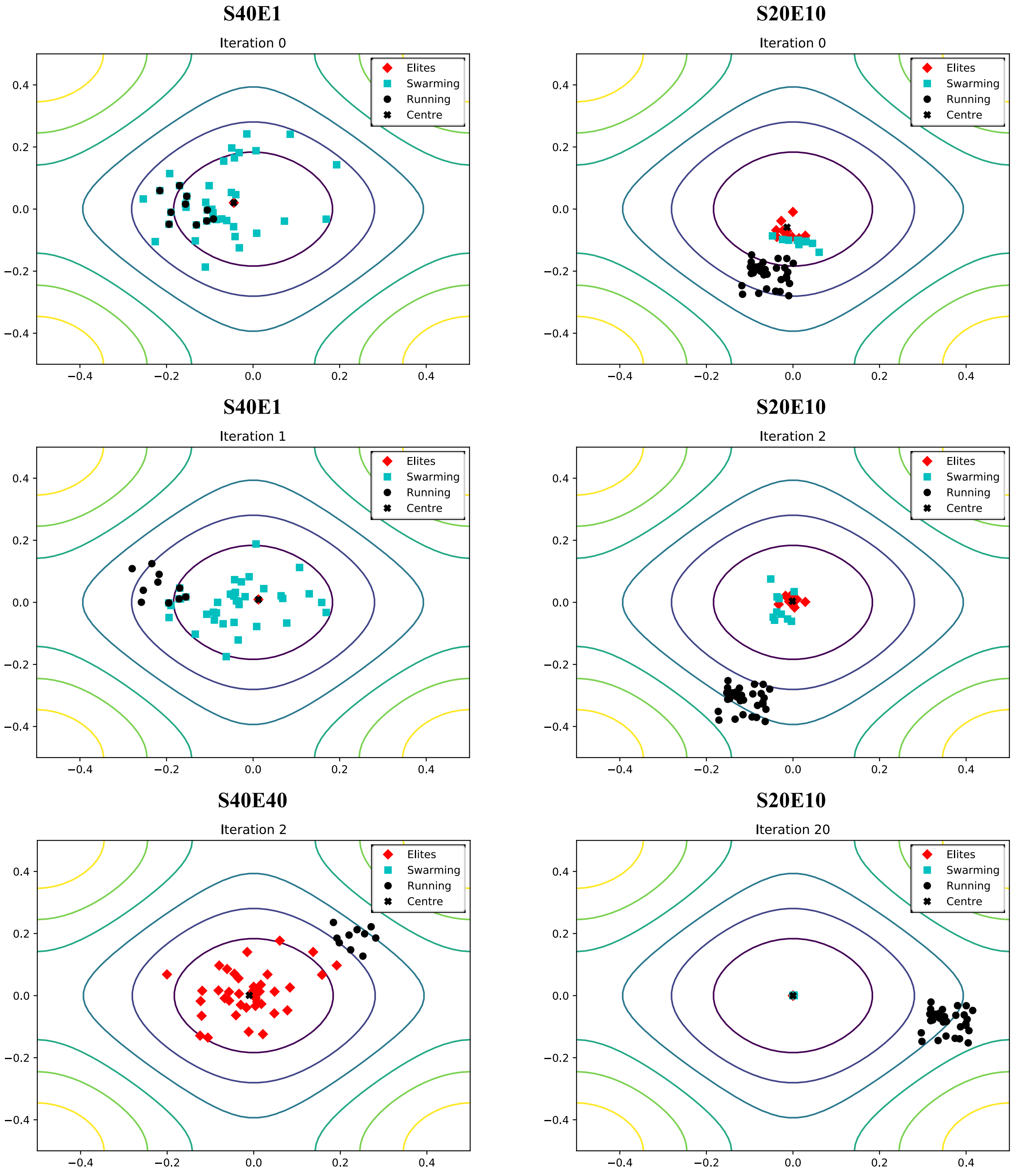

Figure 1: 2D Movement of Parameter Configurations S40E1, S40E40, S20E10

Performance Experiment Results

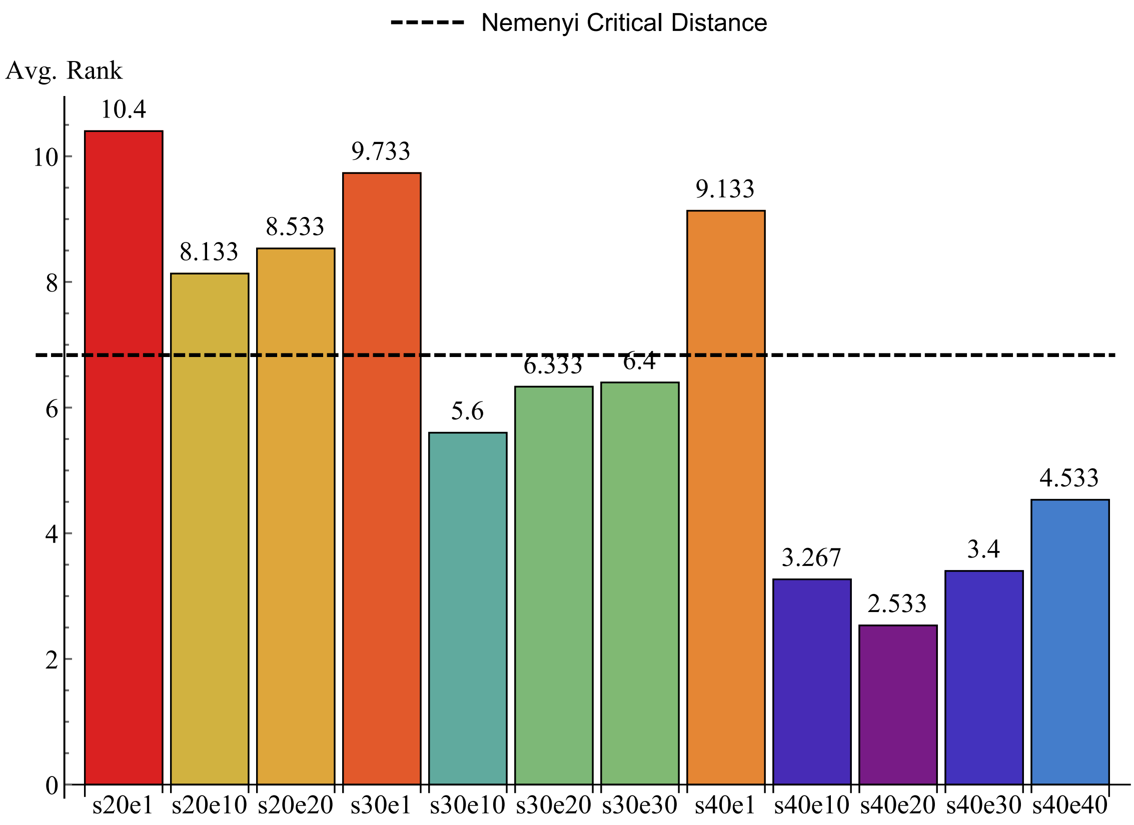

Figure 2: Friedman Rank Test on Complete Set of Parameter Configurations in 10D

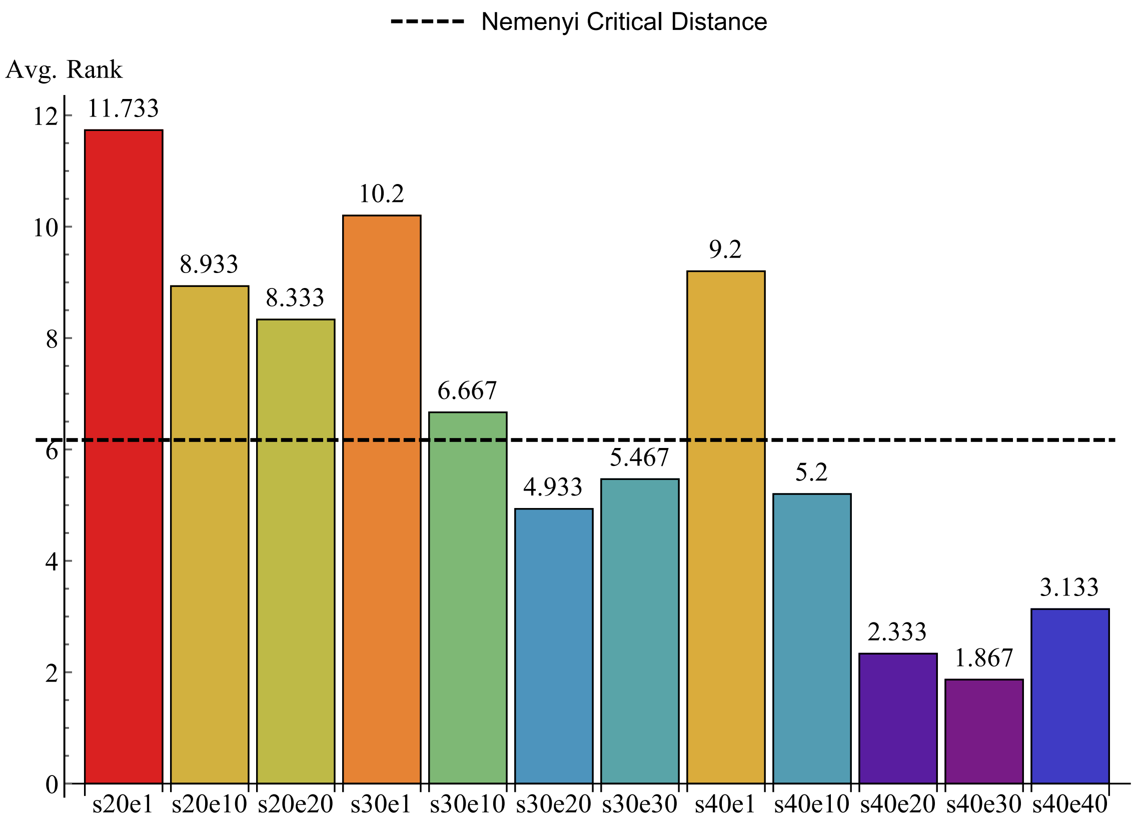

Figure 3: Friedman Rank Test on Complete Set of Parameter Configurations in 30D

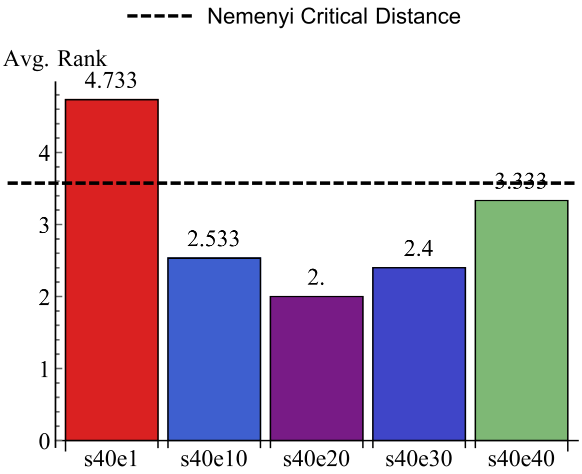

Figure 4: Friedman Rank Test on S40 Set 10D

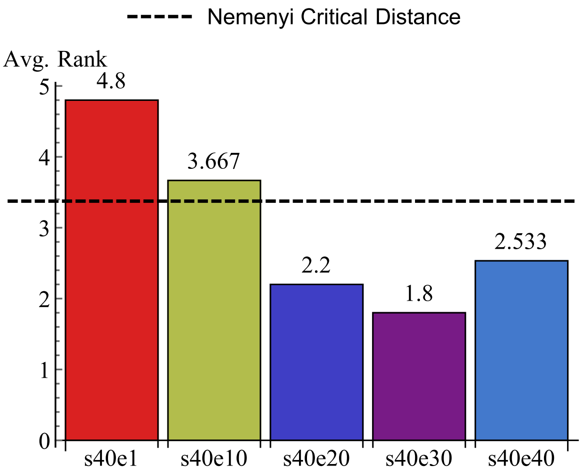

Figure 5: Friedman Rank Test on S40 Set 30D

Table 3: P-Values of the Friedman Rank Tests

	Complete set	S40 set
10 dimensions	2.26E-22	2.79E-07
30 dimensions	1.27E-46	1.45E-10
Time experiments in 10 dimensions	3.91E-74	1.87E-22

Table 4: Significant Wins of the 2 Most Successful Parameter Configurations S40E20 and S40E30 According to the Wilcoxon Rank-Sum Test (α=0.05)

Dimensionality	None	S40E20	S40E30
10 dimensions	11	3	1
30 dimensions	10	2	3

Table 5: Mean Solutions and Standard Deviations of S40E20 and S40E30 in 10 Dimensions

	S40E20 = set 1		**S40E30 = set 2**		**Win**
	avg	*std*	*avg*	*std*	*avg*
f_1	7.29E+02	1.12E+03	5.55E+02	8.29E+02	-
f_2	5.88E-02	2.38E-01	3.33E-02	1.83E-01	-
f_3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	-
f_4	**2.70E-01**	2.13E-01	4.23E-01	9.93E-02	1
f_5	9.15E+00	7.86E+00	1.12E+01	8.81E+00	-
f_6	4.94E-05	2.65E-04	3.24E-06	1.23E-05	-
f_7	2.50E+01	8.27E+00	2.37E+01	8.02E+00	-
f_8	7.21E+00	6.10E+00	7.89E+00	7.63E+00	-
f_9	1.29E-01	4.53E-01	**0.00E+00**	0.00E+00	2
f_{10}	1.01E+03	4.32E+02	1.12E+03	2.52E+02	-
f_{11}	2.83E+00	2.57E+00	2.94E+00	2.01E+00	-
f_{12}	1.06E+04	1.02E+04	8.21E+03	7.32E+03	-
f_{13}	**3.21E+03**	3.06E+03	5.80E+03	4.24E+03	1
f_{14}	**3.41E+01**	6.54E+00	3.77E+01	5.47E+00	1
f_{15}	2.77E+01	1.54E+01	3.51E+01	2.04E+01	-

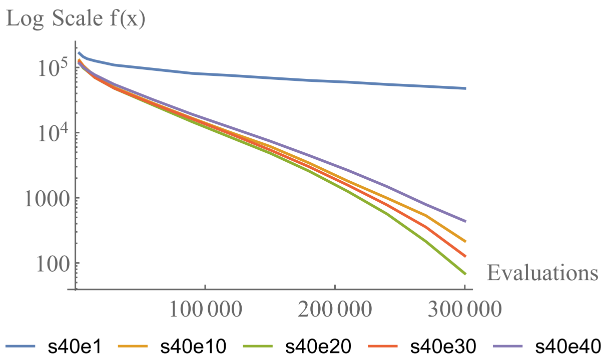

Figure 6: Mean Convergence of 40S Set on F3 in 30D

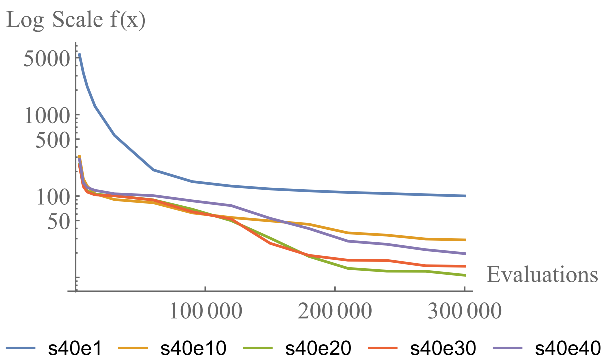

Figure 7: Mean Convergence of 40S Set on F4 in 30D

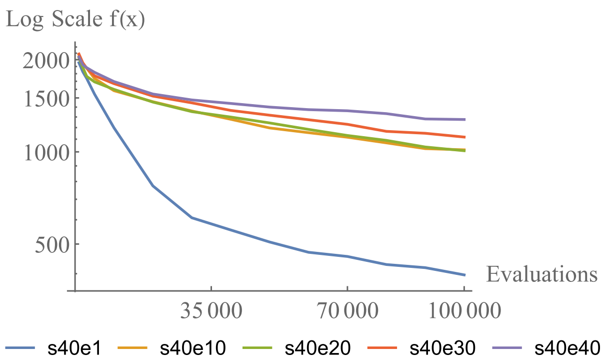

Figure 8: Mean Convergence of S40 Set on F10 in 10D

Computation Time Experiment Results

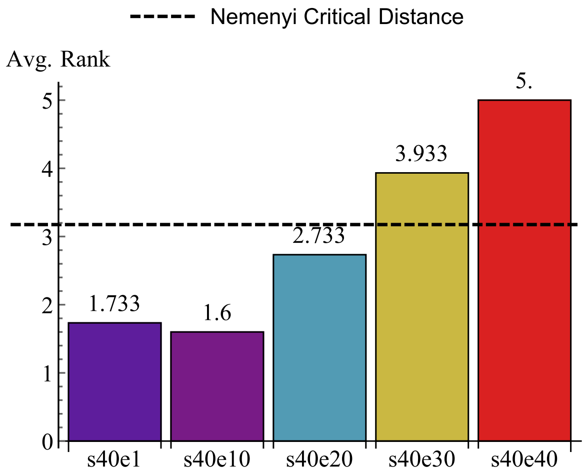

Figure 9: Friedman Rank Test for the Computation Time on S40 Set in 10 Dimensions

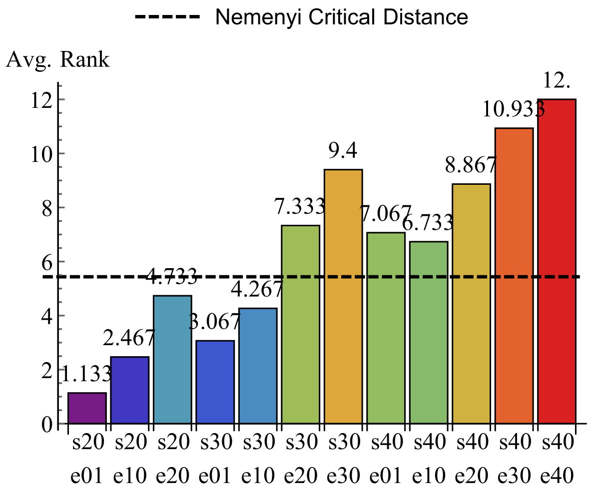

Figure 10: Friedman Rank Test for Computation Time on Complete Set of Parameter Configurations in 10 Dimensions

Table 6: Mean Time Needed for 100 000 Evaluations of 10D Functions in Seconds

	S40E1	S40E10	S40E20	S40E30	S40E40
f_1	5.07	5.02	5.25	5.59	5.75
f_2	5.50	5.42	5.55	5.68	5.83
f_3	5.23	5.10	5.33	5.57	5.81
f_4	5.30	5.13	5.33	5.57	5.79
f_5	5.25	**5.14**	5.38	5.59	5.82
f_6	5.21	5.27	5.53	5.80	6.04
f_7	**4.98**	5.18	5.39	5.64	5.89
f_8	**5.04**	5.18	5.37	5.61	6.11
f_9	**4.95**	5.16	5.37	5.62	5.97
f_{10}	5.75	5.64	5.77	6.02	6.28
f_{11}	**5.13**	5.20	5.39	5.61	5.82
f_{12}	5.20	5.21	5.42	5.63	5.89
f_{13}	5.63	**5.40**	5.51	5.76	5.97
f_{14}	5.84	**5.58**	5.70	5.93	6.08
f_{15}	5.66	5.78	5.58	5.78	5.99
Best times according to the Wilcoxon rank-sum test:					
None	**S40E1**	**S40E10**	**S40E20**	**S40E30**	**S40E40**
8	4	3	0	0	0

DISCUSSION

The results of the performance experiments on the complete testing set indicated the superiority of the configurations with 40 swarming members except for the one with 1 elite bison. However, even the worst ranked S40E1 configuration demonstrated a promising convergence when solving F10 function (Fig. 8) in accordance with the No Free Lunch Theorem (Yang 2012).

A closer investigation of the S40 test set proved the efficiency of the S40E30 and S40E20. Comparing these two configurations proved, that the results were mostly comparable, S40E30 being slightly more successful in 30 dimensions, while S40E20 in 10 dimensions.

The time regarding experiments implied that the elite group size parameter might have a direct influence on the computation time as in most of the cases, the lower the elite group size parameter was, the better time was achieved. Based on these results, there seems to be a conflict between the time and performance requirements.

Even though the difference between the times shown in Table 6 might not seem very wide, it is important to remember, that this experiment included solving 10-dimensional problems only. In higher dimensions, the computation time tends to lengthen just as much as the difference between the time requirements of the parameter configurations.

CONCLUSION

This paper provided interesting findings regarding the parameter configuration of the Bison Algorithm. In most of the functions, the algorithm performed best with the parameter configuration of 40 swarming individuals and 20 or 30 elite members. However, the time experiments preferred the lower amount of both the swarming and the elite individuals.

Since in the real-time optimization are usually requirements for both the quality and the time of the optimization, the obtained results might be useful in the application of the Bison Algorithm on solving real-time problems. For a general optimization, we suggest the S40E20 configuration, as it provided faster evaluations while giving comparable results to the S40E30 configuration.

Since in the convergence analysis even the worst ranked configuration showed remarkable progress in comparison to the others, an adaptive parameter approach of the Bison Algorithm might be considered an exploitable extension of the algorithm and a possible subject of our future research.

ACKNOWLEDGEMENT

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/2018/003. 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

Awad, N. H., Ali, M. Z., Liang, J. J., Qu, B. Y. and P.N. Suganthan. 2016. "Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Bound Constrained Real-Parameter Numerical Optimization, Technical Report". Nanyang Technological University, Singapore.

Bäck, T. 1996. "Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms". Oxford University Press, Oxford, UK.

Berman, R. 2008. "American Bison (Nature Watch)". Lerner Publications, Minneapolis.

Dorigo. M. and T. Stütle. 2004. "Ant Colony Optimization". MIT Press, Cambridge, MA.

Duan, Y. and S. Ying. 2009. "A Particle Swarm Optimization Algorithm with Ant Search for Solving Traveling Salesman Problem". In: *International Conference on Computational Intelligence and Security*, Beijing, 137-141.

Goldberg, D. 1989. "Genetic Algorithms in Search, Optimization, and Machine Learning". Addison-Wesley, Reading, Mass, USA.

Kazikova, A., Pluhacek, M., Senkerik R. and A. Viktorin. 2017. "Proposal of a New Swarm Optimization Method Inspired in Bison Behavior". In: *Recent Advances in Soft Computing (Mendel 2017)*, Advances in Intelligent System and Computing, Springer, In press.

Kazikova, A., Pluhacek, M., Viktorin, A. and R. Senkerik. 2018. "New running Technique for the Bison Algorithm". In: *Lecture Notes in Artificial Intelligence (ICAISC 2018 Proceedings)*, Springer, In press.

Mirjalili, S., Mirjalili, S.M. and A. Lewis. 2014. "*Grey Wolf Optimizer*". Adv. Eng. Softw. 69 (March 2014), 46-61.

Rajasekhar, A., Lynn, N., Das, S. and P.N. Suganthan. 2017. "Computing with the collective intelligence of honey bees–A survey". *Swarm and Evolutionary Computation, 32*, 25-48.

Yang, X.-S. and S. Deb. 2009. "Cuckoo Search via Levy Flights". In: *Proceeding Of World Congress on Nature Biologically Inspired Computing (NaBIC India Dec. 2009)*, IEEE Publications, USA, 210-214.

Yang, X.S. 2012. "Free Lunch or No Free Lunch: That Is Not Just a Question?". Int. J. Artif. Intell. Tools 21 (03)

Yang, X.S. and M. Karamanoglu. 2013. "Swarm Intelligence and Bio-Inspired Computation: An Overview. Swarm Intelligence and Bio-Inspired Computation". 3-23. 10.1016/B978-0-12-405163-8.00001-6.

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.

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.