

The Effect of Neighborhood Functions on Migrating Birds Optimization for Continuous Domain

Mitat Uysal¹, Aysun Güran^{1*}, Mustafa Zahid Gürbüz¹, Celal Barkan Güran²

¹ Dogus University, Acibadem, Kadıköy, 34722 İstanbul, Turkey.

² Istanbul Technical University, Maçka, 34357 İstanbul, Turkey.

* Corresponding author. Tel: +90 (216) 444 79 97; email: adogrusoz@dogus.edu.tr

Manuscript submitted April 4, 2017; accepted July 13, 2017.

doi: 10.17706/jcp.13.4.383-392

Abstract: Migration Birds Algorithm (MBO) is a new optimization algorithm inspired by swarm intelligence. MBO was developed by Duman, Uysal and Alkaya in 2012. They applied it to solve quadratic assignment problem. Some recent studies demonstrate that MBO shows promising results for the problems in continuous domain. Defining an effective neighborhood function is one of the most important phases in heuristic algorithms since it leads a more effective exploration of the solution space. With this motivation, in this work, we analyze the effect of four different neighborhood functions on MBO algorithm. We applied MBO algorithm to 5 different functions on continuous domain. Our contribution in this study is to use different neighborhood functions for MBO. Experimental studies show that while generating neighbors changing all dimensions of the solutions and taking the information of the best solution gives better performance results.

Key words: Migration birds algorithm, neighborhood functions, numerical optimization, swarm intelligence.

1. Introduction

Generally speaking, the optimization problem has the form,

$$\text{Minimize } f(x_1, x_2, \dots, x_n) \quad (1)$$

$$\text{Subject to } \phi_i(x_1, x_2, \dots, x_n) = 0 \quad (i = 1, 2, 3, \dots, l) \quad (2)$$

$$\psi_j(x_1, x_2, \dots, x_n) \leq 0 \quad (j = 1, 2, 3, \dots, m) \quad (3)$$

(2) represents l ($l < n$) equality constraints and (3) represents m inequality constraints. By using vector notation, we may express the general constrained optimization problem as follows:

$$\text{Minimize } f(x) \quad (4)$$

$$\text{Subject to } \phi(x) = 0 \quad (5)$$

$$\psi(x) \leq 0, \quad x \in R_n \quad (6)$$

The problem formulated in (4), (5), and (6) is usually referred to as the general nonlinear programming problem. Any point X that satisfies these equations is called a feasible point.

The difficulties associated with using mathematical optimization on large-scale engineering problems have contributed to the development of alternative solutions [1]. Linear programming and dynamic programming

techniques often fail in solving NP-hard problems [2]. To overcome these problems, researchers have proposed nature-inspired metaheuristic algorithms like Simulated Annealing (SA), Genetic Algorithms (GA), Differential Evolution (DE), Ant and Bee Algorithm (ABA), Particle Swarm Optimization (PSO), Harmony Search (HS), Firefly Algorithm (FA), Artificial Bee Colony (ABC) Algorithm and recently proposed MBO Algorithm.

MBO was developed by Duman, Uysal and Alkaya in 2012 [3]. They applied it to solve quadratic assignment problem. Its performance was compared with SA, tabu search (TS), GA, scatter search (SS), PSO, DE and guided evolutionary simulated annealing (GESA). In this comparison, they stated that MBO outperformed the best performed metaheuristic (SA) by approximately three percent on the average. Duman and Elikucuk [4] applied MBO to solve fraud detection problem. They also proposed a new version of MBO where a different benefit mechanism was used. They tested the original MBO algorithm and its new version on real data and compared their performance with that of genetic algorithm hybridized with scatter search (GASS). Test results showed that the MBO algorithm and its new version performed significantly better than the GASS algorithm. Gao *et al* [5] proposed an enhanced migrating birds optimization (EMBO) algorithm for no-wait flow shop scheduling with total flow time criterion. Their computational results and comparisons showed that the EMBO algorithm is effective for solving the no-wait flow shop scheduling problem with total flow time criterion. Niroomand *et al* [6] introduced a modification of the MBO algorithm to the closed loop layout model with exact distances. To show applicability of the proposed method their results were compared with those of MBO algorithm, SA algorithm as well as Xpress software in the design of closed loop layout. The computational experiments of their study showed that in most of the benchmark problems the results obtained from the proposed MBO method was better than those obtained by other methods which were published in the literature.

MBO algorithm was also tested on continuous functions. Defining an effective neighborhood function is one of the most important phases in heuristic algorithms since it leads a more effective exploration of the solution space. Alkaya *et al* [7] evaluated the performance of the MBO on 30 well-known functions which are tried to be minimized on 2, 10 and 30 dimensional spaces. The objective of their evaluation is to participate in a competition [8] where several meta-heuristics are compared. They developed a novel neighborhood function that uses D dimensional spheres for a more effective exploration plan in the D dimensional solution space. Their results presented that MBO algorithm is effective for solving problems in continuous domain. Makas and Yumusak [9] compared the performance of MBO algorithm with ABC, PSO, DE and GA by using ten multimodal benchmark functions. They also employed MBO algorithm in system identification problems. Their experimental results showed that MBO algorithm performs good optimizations for the benchmark functions, and has a good performance to solve the system identification problems. These two studies motivate us to study on effective neighborhood functions because even though MBO is a recently proposed algorithm the result of these studies show that MBO is also promising for problems in continuous domain. Under these reasons, in this paper, we analyze the effect of four different neighborhood functions on MBO algorithm for continuous domain. The analyzed functions are tested on 5 multidimensional benchmark test functions (De Jong's function Rosenbrock's Valley function, Rastrigin's function, Griewangk's function and Ackley's function) for 2, 5, 10 dimensional spaces. Performance of the functions are compared among themselves and the experimental results are shared.

The remaining parts of the paper are organized as follows: Section II outlines MBO algorithm; Section III points out brief information about neighborhood functions, Section IV presents the test functions, Section V points out the experimental results and finally Section VI gives concluding remarks.

2. Migration Birds Algorithm

MBO algorithm starts with a number of initial solutions corresponding to birds in a V formation. Starting with the first solution (corresponding to the leader bird), and progressing on the lines towards the tails, each solution is tried to be improved by its neighbor solutions. If the best neighbor solution brings an improvement, the current solution is replaced by that one. There is also a benefit mechanism for the solutions (birds) from the solutions in front of them. Here this benefit mechanism is defined as sharing the best unused neighbors with the solutions that follow (here “unused” means a neighbor solution which is not used to replace the existing solution). In other words, a solution evaluates a number of its own neighbors and a number of best neighbors of the previous solution and considered to be replaced by the best of them. Once all solutions are improved (or tried to be improved) by neighbor solutions, this procedure is repeated a number of times (tours) after which the first solution becomes the last, and one of the second solutions becomes first and another loop starts. The algorithm is stopped after a number of iterations. Below, first the notation used and then the pseudocode of the MBO algorithm are given. Let

n = the number of initial solutions (birds)
 k = the number of neighbor solutions to be considered
 x = the number of neighbor solutions to be shared with the next solution
 m = number of tours
 K = iteration limit

The algorithm’s performance is directly affected by the parameters k and x . Parameter k is inversely proportional with the flight speed of the real birds. If small values are chosen for k , the birds are assumed to be flying at higher speeds and this means the MBO algorithm may reduce the total execution time. However, if parameter k increases, the search deepness of the algorithm increases. Therefore, even execution time increases, k may need to be increased for high dimensional problems. Parameter x represents the upwash benefit of trailing tip vortices in a bird flock. Since the bent mechanism of the MBO algorithm is defined as the number of good neighbor solutions obtained from the predecessor solution, high values of x cause solutions to be similar to each other [9].

```

1. Generate  $n$  initial solutions in a random manner and place them on an hypothetical V formation arbitrarily
2.  $i=0$ 
3. while ( $i < K$ )
4. for ( $j=0; j < m; j++$ )
5. Try to improve the leading solution by generating and evaluating  $k$  neighbors of it
6.  $i=i+k$ 
7. for each solution  $sr$  in the flock (except leader)
8. try to improve  $sr$  by evaluating  $(k-x)$  neighbors of it and  $x$  unused best neighbors from the solution in the front
9.  $i=i+(k-x)$ 
10. endfor
11. endfor
12. Move the leader solution to the end and forward one of the solutions following it to the leader position
13. endwhile
14. return the best solution in the flock

```

Fig. 1. Pseudo code of MBO algorithm.

3. The Neighborhood Functions Used For MBO

As it is seen from the pseudocode of the algorithm which is given in Fig. 1, in the first step MBO algorithm generates n initial solutions in a random manner and place them on a hypothetical V formation arbitrarily.

Each solution is a D-dimensional vector. Here, D is the number of optimization parameters. In initialization phase, random values between the lower and the upper boundaries of the parameters are assigned for the parameters of solutions according to (7):

$$x_{ij} = x_j^{\min} + rand(0,1)(x_j^{\max} - x_j^{\min}) \quad (7)$$

where x_{ij} is the position of the i th solution in the j th dimension, x_j^{\min} and x_j^{\max} are the limit values for the j th dimension, and $rand$ is a uniform random number ranging from 0 to 1. After initialization, the flock is evaluated subjected to repeated cycles (the iteration limit (K) and number of tours (m)). During this process for each iteration and each tour k neighbor for leader birds and $(k-x)$ neighbors for the birds except leader should be created.

Defining an effective neighborhood function is one of the most important phases in heuristic algorithms since it leads a more effective exploration of the solution space. With this motivation, in this work we analyze the effect of four different neighborhood functions on MBO algorithm. The following is a list of abbreviations that refers to different neighbor generation functions (NGF1, NGF2, NGF3, NGF4).

- NGF1 creates the neighbors according to (8):

$$\hat{x}_{ij} = x_{ij} + \phi(x_{ij} - x_{kj}) \quad (8)$$

where x_{ij} is the position of the i th solution in the j th dimension, k is a randomly selected index value with the properties ($k \leq n, k \neq i$), x_{kj} is the position of the k th solution in the j th dimension, ϕ is a random number ranging from -1 to 1, and \hat{x}_{ij} is the generated new neighbor position in the j th dimension for the i th solution.

- NGF2 creates the neighbors according to (9):

$$\hat{x}_{ij} = x_{ij} + \phi(x_i - x_k) \quad (9)$$

where x_i is the i th solution, k is a randomly selected index value with the following properties ($k \leq n, k \neq i$), and ϕ is a random number ranging from -1 to 1. The difference between NGF1 and NGF2 is that instead of changing only random one dimension of a randomly selected solution, NGF2 changes all dimensions of that solution.

- NGF3 creates the neighbors according to (10):

$$\hat{x}_{ij} = x_{ij} + \phi(x_{best,j} - x_{kj}) \quad (10)$$

where x_{ij} is the position of the i th solution in the j th dimension, $x_{best,j}$ is the best solution in the j th dimension, ϕ is a random number from the range [-1, 1], k is an integer uniformly chosen from the range [1, n] and is different from i , It can be said that by changing only one dimension, NGF3 uses the information of a randomly selected solution and the current best solution.

- NGF4 creates the neighbors according to (11):

$$\hat{x}_{ij} = x_{ij} + \phi(x_{best} - x_i) \quad (11)$$

where x_i is the i th solution, x_{best} is the best solution, ϕ is a random number from the range [-1, 1]. It can be said that by changing all dimensions of the solutions, NGF4 uses the information of a randomly selected solution and the current best solution.

4. Test Functions

Standard benchmark test functions are used to test the accuracy and robustness of optimization algorithms. Such benchmark functions are necessary especially when the quality of a proposed optimization algorithm needs to be assessed or when several optimization algorithms need to be compared under the same conditions. Examples of such benchmark test functions can be found in [10]. In this study, five functions are selected for test runs (De Jong's function Rosenbrock's Valley function, Rastrigin's function, Griewangk's function and Ackley's function):

- De Jong's function

De Jong's is one of the simplest test benchmark. Function is continuous, convex and unimodal. It has the following definition:

$$f(x) = \sum_{i=1}^n x_i^2 \tag{12}$$

Test area of the function is $-5.12 \leq x_i \leq 5.12, i = 1, \dots, n$. Global minimum $f(x) = 0$ is obtainable for $x_i = 0, i = 1 \dots n$. Fig. 2 shows a two dimensional De Jong's function.

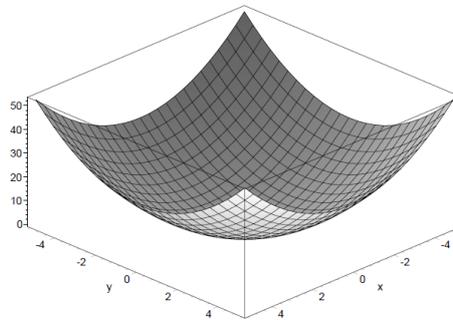


Fig. 2. De Jong's function in 2D.

- Rosenbrock's valley function

Rosenbrock's valley is also known as banana function or the second function of De Jong. It has the following definition:

$$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2] \tag{13}$$

Test area of the function is $-2.048 \leq x_i \leq 2.048, i = 1, \dots, n$. Its global minimum equal $f(x) = 0$ is obtainable for $x_i = 0, i = 1 \dots n$. Fig. 3 shows a two dimensional Rosenbrock's valley function.

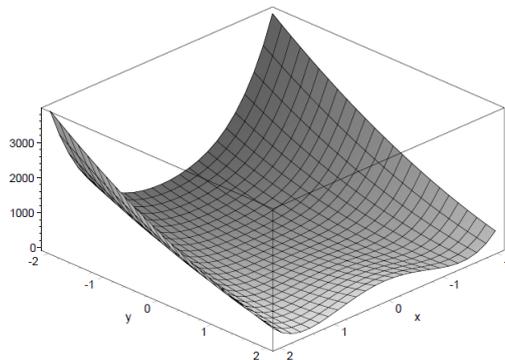


Fig. 3. Rosenbrock's valley function in 2D.

- Rastrigin's function

Rastrigin's function is based on the function of De Jong with the addition of cosine modulation in order to produce frequent local minima. It has the following definition:

$$f(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)] \tag{14}$$

Test area of the function is $-5.12 \leq x_i \leq 5.12, i = 1 \dots n$. Its global minimum equal $f(x) = 0$ is obtainable for $x_i = 0, i = 1 \dots n$. Fig. 4 shows a two dimensional Rastrigin's function.

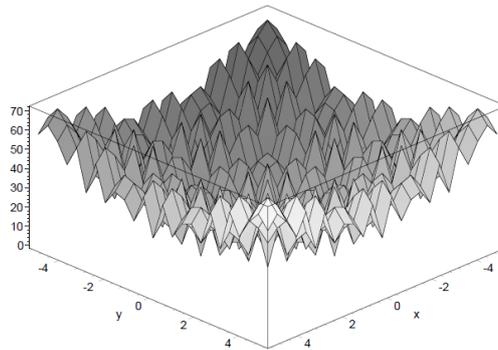


Fig. 4. Rastrigin's function in 2D.

- Griewangk's function

Griewangk's function has the following definition:

$$f(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) \tag{15}$$

Test area of the function is $-600 \leq x_i \leq 600, i=1 \dots n$. Its global minimum equal $f(x) = 0$ is obtainable for $x_i = 0, i=1 \dots n$. Fig. 5 shows a two dimensional Griewangk's function.

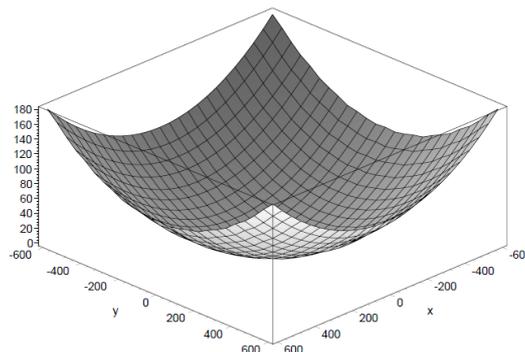


Fig. 5. Griewangk's function in 2D.

- Ackley's Function

Ackley's is a multimodal test function. It has the following definition:

$$f(x) = -20 \exp(-0.2 \sqrt{\frac{2}{D} \sum_{i=1}^D x_i^2}) - \exp\left(\frac{2}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + \theta \tag{16}$$

Test area of the function is $32.728 \leq x_i \leq 32.728, i=1 \dots n$. Its global minimum $f(x) = 0$ is obtainable for $x_i = 0, i=1 \dots n$. Fig. 6 shows a two dimensional Ackley's function.

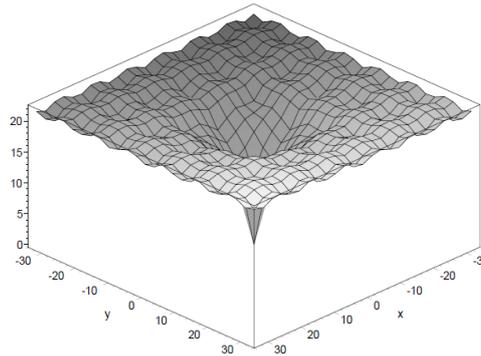


Fig. 6. Ackley's function in 2D.

5. Experimental Results

In order to analyze the effect of NGF1, NGF2, NGF3, NGF4 on MBO algorithm, experiments are repeated 30 times for each benchmark function independently. To have a fair comparison, all experiments use the same settings for the same parameters ($n=51, k=3, x=1, m=10, K=1000$). Through out the experiments, the mean best fitness values are recorded. The experimental results are presented in the following tables for 2, 5 and 10 dimensional spaces.

Table 1. Test Results for D=2

D=2	NGF1	NGF2	NGF3	NGF4
	mean fitness value	mean fitness value	mean fitness value	mean fitness value
De Jong's function	4.82E+00	3.53E-06	1.91E+01	3.72E-02
Rosenbrock's function	4.76E+00	1.74E+00	6.73E+00	1.96E+01
Rastrigin's function	9.77E-01	1.00E-02	1.26E+00	6.06E-01
Griewangk's function	2.30E-01	4.86E-03	3.43E-01	1.32E-02
Ackley's Function	3.84E+00	2.06E-06	3.15E+00	2.13E-01

Table 2. Test Results for D=5

D=5	NGF1	NGF2	NGF3	NGF4
	mean fitness value	mean fitness value	mean fitness value	mean fitness value
De Jong's function	1.76E+03	2.99E+02	3.10E+03	1.44E+02
Rosenbrock's function	1.02E+08	1.71E+05	1.39E+08	3.83E+05
Rastrigin's function	3.08E+01	1.15E+01	3.27E+01	1.12E+01
Griewangk's function	2.05E+01	2.25E+00	1.90E+01	1.64E+00
Ackley's Function	1.63E+01	7.49E+00	1.62E+01	6.79E+00

As it is seen from Table 1, when D is set to 2, NGF2 achieves better performance results than the other neighborhood functions. This means while generating new neighbors of solutions, instead of changing only one dimension, it is better to change all dimensions of the solutions.

When D is set to 5, from Table 2, it is seen that the overall performance results are decreased. In this case, MBO algorithm performs better with NGF4 for most of the benchmark functions (except from the Rosenbrock's function) due to the fact that NGF4 takes advantage of the information of the best solution to

guide the search of neighbor solutions. Since NFG2 performs better than NFG1 and at the same time NFG4 presents higher performance results than NFG3, it can be said that changing all dimensions has better effects in the solution space to discover the global minimum.

Table 3. Test Results for D=10

D=10	NGF1	NGF2	NGF3	NGF4
	mean fitness value	mean fitness value	mean fitness value	mean fitness value
De Jong's function	1.11E+04	2.06E+03	1.26E+04	1.28E+03
Rosenbrock's function	3.40E+09	4.75E+07	2.55E+09	3.90E+07
Rastrigin's function	1.00E+02	4.86E+01	1.02E+02	4.83E+01
Griewangk's function	1.03E+02	2.12E+01	1.06E+02	1.58E+01
Ackley's Function	1.87E+01	1.20E+01	1.91E+01	1.17E+01

As it is depicted from Table 3, when D is set to 10 the MBO algorithm performs better with NGF4 for all of the benchmark functions. From the results it can be said that especially for high dimensional numeric optimization problems using the information of best solution helps to improve the exploitation which means the ability to apply the knowledge of the previous good solutions to find better solutions. Since NFG2 and NFG4 outlines NFG1 and NFG3 respectively, it can be stated that changes in all dimensions rather than change just in one random dimension is better to improve exploration which means the ability to investigate the unknown regions in the solution space to discover the global optimum.

6. Conclusion and Futurework

This paper compares the performance of MBO algorithm under four different neighborhood functions, namely NFG1, NFG2, NFG3 and NFG4. In order to evaluate the algorithm's convergence performance, these neighborhood functions are tested on five benchmark functions namely De Jong's function Rosenbrock's Valley function, Rastrigin's function, Griewangk's function and Ackley's function.

From the experimental results, the neighborhood functions (NGF2 and NGF4) which consider the changes in all dimensions of the solutions outperforms the neighbor generating functions (NGF1 and NGF3) that change only one random dimension of the solutions. Another consequence that can be depicted from the experiments is that taking the advantage of the information of the best solution (NFG3 and NFG4) generally achieves higher performance results especially in higher dimensional spaces. For future studies we'll investigate the control parameters' effect on the performance of MBO algorithm also we'll compare the performance of the MBO algorithm with the other algorithms such as ABC, PSO and GA.

References

- [1] Elbeltagi, E., Hegazy, T., & Grierson, D. (2005). Comparison among five evolutionary-based optimization algorithms. *Advanced Engineering Informatics*, 19(1), 43-53.
- [2] Lovbjerg, M. (2002). *Improving Particle Swarm Optimization by Hybridization of Stochastic Search Heuristics and Self-organized Criticality*. Unpublished masters dissertation, Aarhus Universitet, Denmark.
- [3] Duman, E., Uysal, M., & Alkaya, A. (2012). Migrating birds optimization: A new meta-heuristic approach and its performance on quadratic assignment problem. *Information Sciences*, 217, 65-77.
- [4] Duman, E., & Elikucuk, I. (2013). Solving credit card fraud detection problem by the new

- metaheuristics migrating birds optimization. *Advances in Computational Intelligence Lecture Notes in Computer Science*, 7903, 62–71.
- [5] Gao, K., Suganthan, P., & Chua, T. (2013). An enhanced migrating birds optimization algorithm for no-wait flow shop scheduling problem. *Proceedings of IEEE Symposium on Computational Intelligence in Scheduling* (pp. 9-13).
- [6] Niroomand, S., Hadi-Vencheh, A., Şahin, R., & Vizvári, B. (2015). Modified migrating birds optimization algorithm for closed loop layout with exact distances in flexible manufacturing systems. *Expert Systems with Applications*, 42(19), 6586-6597.
- [7] Alkaya, A. F., Algin, R., Sahin, Y., Agaoglu, M., & Aksakalli, V. (2014). Performance of migrating birds optimization algorithm on continuous functions. *Advances in Swarm Intelligence Lecture Notes in Computer Science*, 8795, 452-459.
- [8] Ying, T., Li, J., & Zheng, Z. (2015). *Introduction and Ranking Results of the ICSI 2014 Competition on Single Objective Optimization* (Report No. 1501.02128). Cornell University Library.
- [9] Makas, H., & Yumuşak, N. (2016). System identification by using migrating birds optimization algorithm: Comparative performance analysis. *Turk Journal of Electrical Engineering and Computer Science*, 24, 1879 – 1900.
- [10] Liang, J., Qu, B., Suganthan, P. N., & Hernández-Díaz, A. (2013). *Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session and Competition on Real-Parameter Optimization* (Report No. 201212). Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Nanyang Technological University.



Mitat Uysal is a professor and dean of the Engineering Faculty at Dogus University (DOU), Turkey. He received his M.S. and Ph.D. degrees from the Mechanical Engineering Department, İstanbul Technical University, Turkey, in 1977 and 1984. In 1987, he worked as associate professor at İstanbul Technical University. In 1993, He worked as professor at İstanbul Technical University. He has been working as professor at Dogus University since 2008. His research interests include heuristics algorithms, optimization techniques, database management systems, computer architecture, data mining.



Aysun Güran is an assistant professor at Dogus University (DOU), Turkey. She received her M.S. and Ph.D. degrees from the Mathematical Engineering Department, Yıldız Technical University, Turkey, in 2007 and 2014. In 2005, she worked as a research assistant at Dogus University. In 2014, she has started to work as assistant professor at Dogus University. Her research interests lie in the area of data mining, text mining, machine learning, natural language processing, big data analytics.



Mustafa Zahid Gürbüz is an assistant professor at Dogus University (DOU), Turkey. He received his M.S. and Ph.D. degrees from the Mathematical Engineering Department, Yıldız Technical University, Turkey, in 2007 and 2015. In 2005, he worked as a software professional at a company. In 2006, he worked as a research assistant at Dogus University. In 2015, he has started to work as assistant professor at Dogus University. His research interests include optimization algorithms, big data analytics and programming languages.



Celal Barkan Güran is currently a lecturer at Istanbul Technical University (ITU), Turkey. After receiving B.S. from Mathematics Engineering in 2005, he received his M.S. and Ph.D. degrees from the Management Engineering Department- Finance Program, Istanbul Technical University, Turkey, in 2008 and 2015. From 2005 to 2008, he worked as a research assistant at Işık University. From 2008 to 2009, he worked as a risk analyst at a bank. His research interests include portfolio optimization, fuzzy logic, stochastic dominance.