POLİTEKNİK DERGİSİ

JOURNAL of POLYTECHNIC

ISSN: 1302-0900 (PRINT), ISSN: 2147-9429 (ONLINE)
URL: http://dergipark.gov.tr/politeknik

Analysis, test and management of the meta-heuristic searching process: an experimental study on SOS

Meta-sezgisel arama sürecinin analiz, test ve yönetimi: SOS üzerine deneysel bir çalışma

***Yazar(lar) (Author(s))**: Hamdi Tolga KAHRAMAN[1], Sefa ARAS[2], Yusuf SÖNMEZ[3], Uğur GÜVENÇ[4], Eyüp GEDİKLİ[5]*

ORCID[1]: 0000-0001-9985-6324

ORCID[2]: 0000-0002-4043-3754

ORCID[3]: 0000-0002-9775-9835

ORCID[4]: 0000-0002-5193-7990

ORCID[5]: 0000-0002-7212-5457

Bu makaleye şu şekilde atıfta bulunabilirsiniz (To cite to this article): Kahraman H. T., Aras S., Sönmez Y., Güvenç U. ve Gedikli E., "Meta-sezgisel arama sürecinin analiz, test ve yönetimi: sos üzerine deneysel bir çalışma", ***Politeknik Dergisi***, 23(2): 445-455, (2020).

Erişim linki (To link to this article): http://dergipark.gov.tr/politeknik/archive

DOI: **10.2339/politeknik.548717**

Meta-Sezgisel Arama Sürecinin Analiz, Test ve Yönetimi: SOS Üzerine Deneysel Bir Çalışma

Araştırma Makalesi / Research Article

Hamdi Tolga KAHRAMAN[1], Sefa ARAS[1], Yusuf SÖNMEZ[2*], Uğur GÜVENÇ[3], Eyüp GEDİKLİ[1]

[1]Teknoloji Fakültesi, Yazılım Müh. Bölümü, Trabzon Karadeniz Teknik Üniversitesi, Türkiye

[2]Gazi Meslek Yüksek Okulu, Elektrik Tek. Bölümü, Ankara Gazi Üniversitesi, Türkiye

[3]Teknoloji Fakültesi, Elektrik-Elektronik Müh. Bölümü, Düzce Üniversitesi, Türkiye

(Geliş/Received : 03.04.2019 ; Kabul/Accepted : 22.05.2019)

ÖZ

Bir arama sürecinde, yerel minimum tuzağına düşmek ya da küresel minimum noktasını atlamak tıpkı yapay zeka yöntemlerinde olduğu gibi meta-sezgisel algoritmaların en büyük sorunlarından biridir. Bu çalışmada, bu sorunların nedenleri araştırılmış ve yeni çözüm yöntemleri geliştirilmiştir. Bu amaçla, meta-sezgisel algoritmaları test ve analiz etmek için yeni bir çerçeve geliştirilmiştir. Ayrıca yeni ve güçlü bir meta-sezgisel yöntem olan Simbiyotik Organizmalar Arama (SOS) Algoritması için analiz ve test çalışmaları yapılmıştır. Çalışmanın amaçlarından biri, simbiyotik operatörlerin doğal ekosistem taklit başarısını ölçmektir. Böylece, arama sürecindeki problemler keşfedilmiş ve geliştirilen test ve analiz yönteminin bir örneği olarak operatörlerin tasarım hataları ortaya çıkmıştır. Dahası, kesin bir komşuluk arayışını gerçekleştirme ve yerel minimumdan kurtulma yolları (çeşitliliği arttırmak) araştırılmıştır. Araştırma sürecindeki operatörlerin performansını arttıran önemli bilgiler deneysel çalışmalarla sağlanmıştır. Ayrıca, bu çalışmada geliştirilen ve sunulan yeni deneysel test yöntemlerinin, tasarım ve test için meta-sezgisel algoritma çalışmalarına katkıda bulunması beklenmektedir.

Anahtar Kelimeler: Simbiyotik organizmalar arama, test problemi, algoritma analizi, komşuluk araması, çeşitlilik.

Analysis, Test and Management of the Meta-Heuristic Searching Process: An Experimental Study on SOS

ABSTRACT

In a search process, getting trapped in a local minimum or jumping the global minimum problems are also one of the biggest problems of meta-heuristic algorithms as in artificial intelligence methods. In this paper, causes of these problems are investigated and novel solution methods are developed. For this purpose, a novel framework has been developed to test and analyze the meta-heuristic algorithms. Additionally, analysis and test studies have been carried out for Symbiotic Organisms Search (SOS) Algorithm. The aim of the study is to measure the mimicking a natural ecosystem success of symbiotic operators. Thus, problems in the search process have been discovered and operators' design mistakes have been revealed as a case study of the developed testing and analyzing method. Moreover, ways of realizing a precise neighborhood search (intensification) and getting rid of the local minimum (increasing diversification) have been explored. Important information that enhances the performance of operators in the search process has been achieved through experimental studies. Additionally, it is expected that the new experimental test methods developed and presented in this paper contributes to meta-heuristic algorithms studies for designing and testing.

Keywords: Symbiotic organisms search, benchmark problem, algorithm analyze, intensification, diversification.

1. INTRODUCTION

The term meta-heuristic is used for artificial intelligence (AI) methods, which search the best solution among all possible candidates. The major handicap of these algorithms, they do not guarantee that the best solution is also the global optima. Two most important concepts are intensification and diversification in meta-heuristics [1-2]. Numerous meta-heuristics have been developed for solving combinatorial and numeric optimization problems. These can be sorted depending on the historical development as; genetic algorithm (GA, 1970s and 1980s) [3, 4], ant colony optimization (ACO, 1992) [5-7], particle swarm optimization (PSO) [8,9], Cuckoo Optimization Algorithm (COA) [10], artificial bee colony (ABC) [11,12], gravitational search algorithm [13,14], teaching learning-based optimization (TLBO) [15], symbiotic search algorithm (SOS) [16,17], criss cross search particle swarm optimizer (CSPSO) [18], modified ant colony optimization (ACO) [19], Hybrid Evolutionary Immune Algorithm [20], ant colony optimization for continuous functions by using novel pheromone updating [21], backtracking search optimization algorithm for numerical optimization problems [22], Dynamic Virtual Bats Algorithm [23], Weighted Superposition Attraction (WSA) [24], the multi-objective vortex search algorithm [25], a multi-objective artificial algae algorithm [26], a hybrid firefly and PSO algorithm [27], OnLine Learning Algorithm using Worst-Violators (OLLAWV) [28] and queuing search algorithm [29] are the important heuristic algorithms which have developed from past to present.

*Sorumlu yazar (Corresponding Author)
e-posta : ysonmez@gazi.edu.tr

Measuring the effects of parameters on search performance is another research topic [30]. Moreover, a detailed review study, which is presented on an extensive survey of work concerning parameter control in evolutionary algorithms topic by Karafotias et al [31] can be investigated. In the Large-Scale Global Optimization problems, heuristic algorithms should have effective and efficient search strategies. Trunfia et al. [30], in the solution of high-dimensional optimization problems, investigates the impact of populations dimension and subcomponents dimensionality on performance of a cooperative co-evolutionary optimizer.

The biggest problem encountered in the testing process of meta-heuristic algorithms is to get caught in a local optimum trap [32]. Search process of the global minimum in solution space results in failure, in this situation. The search process of heuristic algorithms should be analyzed to identify the problem source. The search process of the algorithm can be managed to solve the problem. Making a manageable search process is a challenge task [30]. This case explains that why so much work has been done on heuristic algorithms. An important factor in the success of meta-heuristics is to perform parallel search. These population or colony-based algorithms scan the search/problem space with solution candidates more than one. In this process, there are three main issues that affect the search success of meta-heuristic algorithms. First of all, local minimum traps encountered in the search process. One of the basic reasons for trapping in local minimum is not to ensure diversification effectively. The algorithm should not stick in local minimum for a successful search. It is essential to ensure significant diversity among alternative candidates. A successful diversity operation depends on how the algorithm creates new solution candidates. Sun et al. [32] have benefited from genetic algorithm operators to solve the "premature convergence" and "computationally expensive" problems of Gravitational Search Algorithm. Similarly, the PSO [8] algorithm converges to the local minimum in solution of multi-modal problems. Meng et al. [18] have developed operator "vertical crossover" to increase swarm diversity in Crisscross Search PSO (CSPSO). Basic questions to be answered are to analyze a meta-heuristic search method for diversification: (i) Which operators provide the diversity? (ii) Can diversity be achieved successfully? (iii) How can diversity be managed? Answers of these question related to diversity will be investigated as "local minima (lm)" in "case 1, 2 and 3" in "Experimental Studies" section. The second issue is that meta-heuristic algorithms should perform the neighborhood search successfully. The success criterion in this respect is the degree of convergence to the optimum point in search space of the algorithm. Meng et al. [18] have developed the operator "horizontal crossover" to improve the global convergence ability in CPSO algorithm. In order to analyze a meta-heuristic algorithm for intensification or global convergence ability, basic questions to be answered are: (i) Which operators perform the intensification? (ii) Intensification can be successfully achieved while diversification is being performed? (iii) How can intensification be managed? The other reason for trapping in local minimum is not to ensure intensification effectively. This characteristic will be investigated as "neighborhood search (ns)" in "Experimental Studies" section. The last issue is that meta-heuristic algorithms should scan a multi-dimensional search space quickly [30] and find optimum solutions in a short computation time. The two properties mentioned above are a measure of the robustness of the algorithm. Novel algorithms have to prove that they produce faster and/or more effective results than before depending on these criteria [16, 17, 31]. Benchmark problems are used as a general approach in performance tests of meta-heuristics. The success of algorithm is measured by degree of global solution convergence and discovering time for an acceptable answer (finding the global optima in a finite time interval) [33]. The last issue is that meta-heuristic algorithms should scan a multi-dimensional search space quickly [30] and find optimum solutions in a short computation time [34]. This characteristic will be investigated as "execution time (et)"in "Experimental Studies" section.

In this paper test and analysis of a meta-heuristic search algorithm for three basic issue (lm, ns, et) explained above. A framework has been designed for this purpose and various methods have been developed for analyze process. Moreover, in this paper, parameter control and its effects on lm, ns and et have been investigated. As a meta-heuristic search method, Symbiotic Organisms Search (SOS) algorithm is preferred. The data required for the analysis has been obtained from experimental studies. The searches have been realized on the operators of SOS that constitute the search process. In experimental studies, make search symbiotic operators (mutualism, commensalism and parasitism) individually and together. That is, experimental results for 7 different operator combinations (all possible test cases) have been analyzed. In order to investigate effects of SOS operators and benefit factors (BF1 and BF2) on intensification and diversification, 26 benchmark problems at different levels of complexity have been used. According to experimental results, function and ability of SOS operators have been uncovered in terms of lm, ns and et. Other sections of this paper is composed as follow. In Section 2, detailed information of SOS is explained. Experiments are given detailed in Section 3. At last, in Section 4, conclusions have been given, at last.

2. SOS ALGORITHM

SOS is a novel and strong searching algorithm [16]. It performs a simulation of the process of symbiosis between organisms. SOS algorithm makes the discovery of a huge number of solution candidates in the search area. Solution candidates are called organisms in the SOS algorithm. A community of organisms is called as ecosystem. The interactions between organisms reveal the success of SOS algorithm. Scientists have identified the interaction of organisms in nature in three categories. These are mutualism, commensalism and parasitism. To model these interactions among the organisms, three operators are defined in SOS. The mutualism operator provides mutual utilization between the two species. The commensalism operator simulates a symbiotic relationship, while one organism profit from a relationship with another organism, the other organism neither profit nor is damaged. Parasitism operator simulates an interaction among two species of organisms. SOS algorithm consists of five steps. Organisms are created in the first step. An organism corresponds to a solution nominee for a problem. Ecosystem is initialized with creating a set of organisms in the second step. In the third step, fitness values of all organisms are evaluated. The best organism is chosen based on fitness values of organisms in ecosystem. Ecosystem life cycle starts at the next stage. Mutualism (M), commensalism (C) and parasitism (P) (symbiosis relationship) are applied between organisms in the fourth step. When life cycle of ecosystem is finalized the best organism corresponds to optimum solution. The main feature that distinguishes the SOS from other search methods is symbiotic relationships. These relationships are actuated by operators M, C, P. Experimental study is carried out through these three operators.

3. EXPERIMENTAL STUDY

In the preparation phase of the experimental work, mathematical expressions and working principles of SOS operators are examined. In the SOS algorithm, the search process is done by the operators M, C and P. These operators are applied independently and in sequence to all organisms in the ecosystem. The operating principle of SOS operators is given below.

i. Mutualism operator (M):
   a. A randomly selected organism is defined as X_j, where $X_j \neq X_i$.
   b. Mutual relationship vector (MV) and benefit factors (BF1, BF2) are determined.

   $$MV = (X_i + X_j) / 2 \quad (1)$$

   c. X_i and X_j organsims are modified by using Eq. 2 and Eq. 3:

   $$X_{inew} = X_i + rand(0, 1) * (X_{best} - MV * BF1) \quad (2)$$

   $$X_{jnew} = X_j + rand(0, 1) * (X_{best} - MV * BF2) \quad (3)$$

   d. Fitness values of X_{inew} and X_{jnew} are calculated. If the modified organisms are more sutitable than the previous the modifications are adopted. Otherwise previous organisms are kept.

ii. Commensalism phase (C):
   a. An X_j organism is selected randomly, where $X_j \neq X_i$.
   b. Organism X_i is modified by using as follow.

   $$X_{inew} = X_i + rand(-1, 1) * (X_{best} - X_j) \quad (4)$$

   c. Fitness of the X_{inew} is computed. If the modified organism is more suitable, new organism is accepted instead of X_i. Otherwise, the modification is rejected and previous organism X_i is kept.

iii. Parasitism phase (P):
   a. An X_j organism is selected randomly, where $X_j \neq X_i$.
   b. A Parasite Vector (PV) is created from X_i
   c. X_i's fitness is computed. If the fitness value of PV is more suitable than X_j, X_j is replaced with PV. Otherwise, do not make replacement operation and keep X_j delete Parasite_Vector.

In the subsection below, the application phase of the experimental work is explained. At this stage, experimental work is introduced and the algorithms are analyzed by evaluating the data obtained from these studies.

3.1. Definition of Experimental Work

In experimental studies, answers to the following questions are sought: in order to make a deep analysis of the SOS algorithm on escape from local minima (lm) and neighbour hood search (ns) issues.

3.2. Setup for Experimental Work

Experimental studies have been conducted on 26 benchmark problems. These benchmark problems are the same as those reported in studies that represent SOS [16]. The organisms number of the ecosystem is determined as 50 and the number of epochs are 100 thousand. Each experimental study has been run for 100 times to increase validation of the test results, and average and best values have been recorded. For this purpose, the pseudo-code of test process for operators is given in Algorithm 1. Test cases of operators can be summarized as follows:

Algorithm 1. Pseudo-code for test procedure

1: select a case study (case 1, 2 or 3)
2: initialization *(select a benchmark problem)*
3: repeat *(100000 times)*
   (i) choose symbiotic relationship(s)
4: until: termination criterion (epoch number and/or fitness value) is met
5: stop the searching process and save the experimental results (best organism X_{best} and the mean and the average fitness values in all of the ecosystems)

Operators shown as choose symbiotic relationship are: M, C and P. In the test process, each operator performed the search process separately and in double combinations. In this way, selected operators create 7 different test cases (M, C, P, MC, MP, CP, MCP). Here, MCP is the original SOS algorithm.

Case-1 (Analyzing of operator combinations): The operator combinations M, C, P, MC, MP, CP, MCP have been implemented on benchmarks and results have been compared with each other. As a result of these comparisons The effects of the symbiotic operators' co-operation or independent work on the search process have been investigated. The information of all the test studies and the results obtained has been recorded. By analyzing the recorded results, important information about operators' functions and works have been discovered. In this respect, important information about "lm", ns" and "et", which are the research topics of the article, has been reached.

Case-2 (Analyzing of parameter effects): Eqs. 1, 2 and 3 are used when constructing a new organism with the M operator in the SOS algorithm. BF1 and BF2 are benefit factors. Cheng and Prayogo [16] randomly select the benefit factor of M operator as 1 or 2 in the SOS algorithm. However, how the change of the benefit factor affects the search process is a matter to be investigated. In the experimental study, it has been aimed to find optimum value for the benefit factor. The effect on problem solving has been investigated by assigning random values to the benefit factor at the determined intervals. Ten different test cases have been created as (0-0.5), (0.5-1), (1-1.5), (1.5-2), (0-1), (0.5-1.5), (1-2), (0-2) and 1, and also by including the state of the original SOS algorithm. Experimental studies have been carried out by applying SOS algorithm with 10 different test cases for all benchmark problems. Depending on the results obtained, the effect of the benefit factor (BF) on solving the problem has been examined. Thus, the optimum value of the benefit factor has been investigated. The effects of "BF1" and "BF2" on "lm", "ns" and "et" have been also examined.

Case-3 (Analysis of solution candidates): In this case, effects of the solution candidates / organisms produced by the operators M, C and P on lm (intensification) and ns (diversification) have been investigated. The X_{best} organism that is the best organism for the problem of the SOS has been used to carry out this research. In other words, X_{best} represents the current minimum point (for minimization problems) in the SOS algorithm. As in other meta-heuristics, the SOS also realizes neighborhood search (current) by searching around best solution candidate (X_{best}). For example, in the Artificial Bee Colony algorithm [16], the neighborhood search is performed by using the environment size parameter. Depending on the environmental size parameter, the new solution candidate makes a neighborhood search around the best solution candidate (elite bee). Similarly, the [4] neighborhood search in the Genetic Algorithm is performed with the help of the crossover operator. The crossover is carried out with child individuals placed around the parental individual in the population. All this information shows that the neighborhood search in the SOS algorithm is carried out through organisms located around X_{best}. So the subject that needs to be investigated is: How much change is taking place after any organism has been exposed to the operators M, C, P? How this organism is positioned according to X_{best} or How far is it to X_{best}? While the increase in the amount of distance increases the diversity (lm), decrease in distance increases the intensification (ns).

In order to realize Case-3, distances between organisms must be measured. To accomplish this, let's first look at how an organization is represented. In the SOS algorithm, an ecosystem consists of organisms. Each organism corresponds to a solution candidate for problem. Suppose a problem consists of n-parameters and the optimum value of these parameters is sought. In such a problem, the parameters are represented by a vector <p1, p2, p3, ..., pn>. The solution candidates set (ecosystem) to be created to find optimum values of these parameters represented by E. In this case, each organism (solution candidate) in *E* set consists of a vector representing the optimum parameter values for the problem. Within this vector, there will also be a parameter that represents the fitness value of each organism *f* for the problem. According to this information, an *E* consisting of *m* organisms can be described as follow:

$$E_{[m,n]} \equiv \begin{bmatrix} e_{p[1,1]} & \cdots & e_{p[1,n]} & f_{[1]} \\ \vdots & \ddots & \vdots & \\ e_{p[m,1]} & \cdots & e_{p[m,n]} & f_{[m]} \end{bmatrix} \quad 5$$

The X_{best} organism is the organism having the best fitness value (*f*) in the ecosystem given in Eq. 5. As given in Algorithm 1, after X_{best} is determined in SOS algorithm, M, C and P operators are applied to this organism, respectively. In order to perform Case-3, the operators M, C, and P must be applied without changing X_{best}. Thus, the effects of these operators on the same X_{best} can be examined in the most accurate way. So now, the thing is to be done is to measure how much the organisms resemble the X_{best} organism after the interaction process of M, C, and P. As a measure of similarity, vector distance information has been used. The measurement procedure can be determined as given in Eq. 6 below by taking into account the Euclidian distance (EU) between the organism, which is derived from the interaction process, and the X_{best}.

$$EU: d\left[X_{best}, X_{(M,C,P)}\right] = \sqrt{\left(X_{best[1]} - X_{(M,\ C,\ P)[1]}\right)^2 + \left(X_{best[2]} - X_{(M,\ C,\ P)[2]}\right)^2 + \cdots + \left(X_{best[n]} - X_{(M,\ C,\ P)[n]}\right)^2} \quad 6$$

Eq. 6 shows the obtaining of the distance vector between the two organisms. Accordingly, this distance vector is three dimensions. These can be represented as given in Eq. 7.

$$d[X_{best}\ X_{M,C,P}] \equiv \begin{bmatrix} d_M \\ d_C \\ d_P \end{bmatrix} \qquad 7$$

Based on the values of the distances given in Eq. 7, it reveals that which one of organisms X_M, X_C, X_P is closest to X_{best}. Thus, after the X_{best} has been exposed to the operators M, C, and P, It reveals that which operator has undertaken the task of searching for a neighbor. It is discovered also that which operator creates organisms (which increases diversity) that are the farthest from X_{best}. Concepts and research topics of experimental study have been described in detail in Cases-1, 2 and 3, and experimental studies have been carried out based on these research topics. In the next subsection, the results obtained from experimental studies are examined in detail.

3.3. Experimental Results

Results obtained from experiments carried out according to Case-1, 2 and 3, which are explained above, are given.

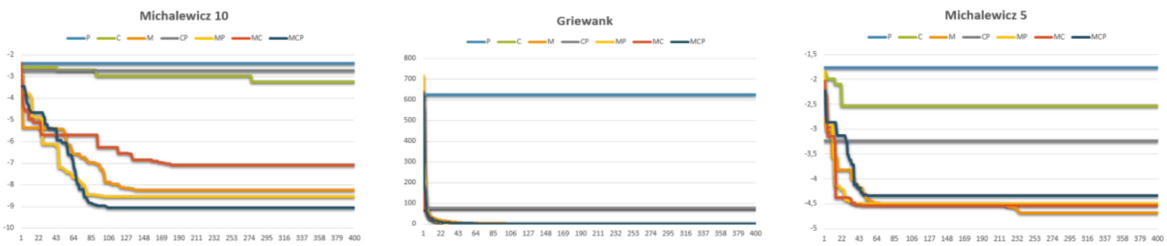
3.3.1. Results of case 1: analysis of operator combinations

In this experimental study, it has been aimed to investigate the effect of operators in finding the best solution of SOS algorithm according to the conditions defined in Case-1. In other words, the functions of the operators in terms of lm (diversification) and ns (intensification) and the performances of the operators in terms of et have been revealed. First, the effect of symbiotic operators on finding the global minimum (7 different combinations / test cases) has been examined. Average and best fitness values have been measured for each combination of operators and for all benchmarks. Only the number of iterations has been selected as the criterion for termination for Case-1 experimental studies to discover search success of each operator. Each operator (test case) has been run 100 times and average fitness and the best fitness values have been recorded. The results obtained for 26 different benchmark problems and 7 different SOS combinations are given in Table 1.

MCP (SOS): MCP is the operator used in the original SOS algorithm. As given in Table 1, the original SOS algorithm did not find the global min value at 8 out of 26 benchmark problems. This situation shows that the original SOS algorithm has problems in lm and ns. The original SOS combination (M, C, P) showed a better search performance than the others except MC, when compared to other test cases / combinations. However, it should be noted that in the two benchmark problems (numbers 9 and 23) the MC test case yields better results than the original MCP. This proves that the P operator has a disruptive effect.

M: When compared with MCP, the search process for 6 benchmark problems (1, 20, 21, 22, 24, 26) has failed. Care must be taken that M operator cannot find a global minimum when working alone. This indicates that the M operator gets trapped to the local minimum. That is, the M operator performs the ns process but cannot escape from the local minimum because it lives the lm problem. M operator cannot achieve diversification while satisfactorily performing intensification. As can be seen in Table 1, this problem has removed with MC. It can be easily seen that the operator C allows escaping from the local minimum, when it is included in the search process. In other words, C operator partially solves the lm problem by making large jumps (splashes) in search space.

C: It has performed unsuccessfully alone. The results given in Table 1 should be considered together with the work process of C operator and M operator (see Eqs. 1, 2 and 3). The mimicry proportion of an organism exposed to the C operator (see Eq. 4) to the X_{best} organism is less than the organism exposed to the M operator (see Eqs. 1, 2 and 3). In other words, the C operator does not bring the organisms in the ecosystem close to X_{best} as the M operator. This explains why the C operator is not as effective as the M operator in ns duties. In Fig. 7, the distances between the organisms exposed to the operators M, C and P and the X_{best} organism are given. All this information also explains how the C operator gets rid of the local minimum traps. Because getting trapped in a local minimum is caused by making a close call to the best solution candidate. The C operator can make large and random search in the solution space compared to the M operator. One reason for this is the effect that is created by the operator rand (-1,1) given in Eq. 4. This operator can reduce the effect of X_{best} to zero in the creation of the new organism (completely destroy). However, C operator is more effective than M in lm tasks. This



(a) (b) (c)

Fig. 1. Variation of fitness values produced by the operator combinations for three different benchmark problems

Table 1. The result obtained by applying 7 test cases (MCP, M, C, P, MC, MP, CP) to 26 benchmark problems

Algorithm / Function	Global Min	SOS (MCP)		M		C		P		MC		MP		CP	
		Mean	**Best**	**Mean**	**Best**	**Mean**	**Best**	**Mean**	**Best**	**Mean**	**Best**	**Mean**	**Best**	**Mean**	**Best**
Beale	0	0	0	0,03810	0	0,00148	1,23E-05	0,44334	0,02199	0	0	0	0	0,00194	1E-05
Easom	-1	-0,99	-1	-1	-1	-0,2160	-0,9992	-0,0003	-0,0334	-1	-1	-1	-1	-0,1881	-0,9998
Matyas	0	0	0	7,6E-189	1,1E-195	9,5E-05	0	0,20803	0,00225	0	0	5E-127	7,1E-135	0	0
Bohachevsky1	0	0	0	0	0	5,32774	0	331,162	0,55412	0	0	0	0	24,8360	0
Bohachevsky2	0	0	0	0	0	13,8529	0	371,557	0,78026	0	0	0	0	0,06376	0
Bohachevsky3	0	0	0	0	0	0,03683	0	362,038	4,16026	0	0	0	0	0	0
Booth	0	0	0	0	0	0,07600	0,00033	8,04570	0,19231	0	0	0	0	0,06459	8,4E-05
Michalewic2	-1,8013	-1,8013	-1,8013	-1,8013	-1,8013	-1,3732	-1,7972	-1,1413	-1,7951	-1,8013	-1,8013	-1,8013	-1,8013	-1,3693	-1,7927
Michalewic5	-4,6876	-4,5010	-4,6876	-4,4911	-4,6876	-2,52271	-3,13146	-2,0079	-3,01954	-4,45805	-4,68766	-4,45697	-4,68766	-2,4519	-3,3190
Michalewicz10	-9,6601	-8,1378	-9,4131	-8,07331	-9,38186	-3,41808	-4,01192	-3,08595	-4,505	-8,06859	-9,27263	-7,96508	-9,32461	-3,3657	-4,5444
Schaffer	0	0	0	1,25E-14	0	0,00079	0	0,30981	0,04507	0	0	6,75E-08	0	0,00556	0
S.H.C.B.	-1,0316	-1,0316	-1,0316	-1,0316	-1,0316	-1,0315	-1,0316	0,0677	-0,9713	-1,0316	-1,0316	-1,0316	-1,0316	-1,0005	-1,0316
Shubert	-186,73	-186,62	-186,73	-186,73	-186,73	-159,71	-186,71	-181,52	-186,72	-186,73	-186,73	-186,68	-186,73	-186,43	-186,73
Colville	0	5,7E-05	9,1E-07	2,76E-06	3,28E-09	39,4743	1,23202	3030,02	16,3341	1,18E-05	5,84E-08	0,00011	3,44E-08	39,8829	1,56494
Zakharov	0	0	0	2,69E-98	2,1E-102	0	0	712,442	48,7634	0	0	7,54E-65	9,7E-68	2,13741	0
Step	0	0,02309	6,4E-07	0,00033	1,1E-07	9,32940	7,5	172,390	118,525	0,01434	3,11E-06	0,00308	5,16E-08	14,0811	7,5
Sphere	0	0	0	3,8E-134	1,1E-136	1999,56	0	64621,2	43821,4	0	0	2,51E-88	2,54E-90	1241,07	0
SumSquares	0	0	0	3,9E-135	3,9E-138	207,848	0	9357,76	5933,91	0	0	5,47E-89	9,65E-92	302,495	0
Quartic	0	0,03856	0,00086	0,03107	0,00017	0,03403	0,00046	443,024	222,634	0,03817	0,00018	0,03151	0,00047	0,03777	0,00090
Schwefel 1.2	0	0	0	2,26E-44	8,25E-50	691,701	0	106360	48834,1	0	0	8,29E-29	3,71E-32	1133,26	0
Schwefel 2.2	0	0	0	1,98E-68	7,48E-70	1,21905	0	115,388	96,1462	0	0	2,62E-45	3,41E-46	5,57258	0
RosenBrock	0	0,40003	1,8E-08	2,57E-08	4,67E-11	5,03885	2	66788,7	142,572	0,34	1,72E-09	6,78E-07	3,42E-09	2	2
Dixon-Price	0	0,80035	0,00012	0,10586	2,08E-06	18591,1	464	1686738	994061	0,86549	5,55E-05	0,47819	1,96E-05	39367,3	464
Rastrigin	0	0	0	0	0	9,20769	0	432,482	358,666	0	0	0	0	12,7528	0
Griewank	0	1,02102	1,00003	1,00248	1,00001	80,9720	76	596,158	373,607	1,04747	1,00000	1,00572	1,00000	80,3316	76
Ackley	0	0,08628	0,08628	0,08628	0,08628	0,08628	0,08628	20,5341	20,1546	0,08628	0,08628	0,08628	0,08628	0,08628	0,08628

indicates that the operator C can be used to solve the lm problem.

P: It has performed unsuccessfully alone. It has also no positive effect in other combinations. It is obvious that there is a problem in the design of this operator. One of the organisms exposed to the operator P must benefit. It is clear from Table 1 and Fig. 1 that such a situation does not take place.

MC: It is the most successful test case. Compared to M operator, it escapes from local minimum traps (operator C is the way to escape from a local minimum trap). Therefore, it partially removes the lm problem. In addition, when t is compared to the operator C, it performs the neighborhood search more successfully.

MP: When compared with the simple M operator and the MCP, the search process seems to have failed due to the disruptive effect of the P operator. The validity of these evaluations can be easily understand from Table 1 and Fig. 1.

CP: It is understood that it could not make a neighborhood search and that the global minimum points have been missed. It should be noted that operator M has the function of searching neighborhood. Therefore, it is understandable that the failure of the ns process when the operator M is not used.

3.3.2. Results of case 3: analysis of solution candidates

In this subsection, the effects of the solution candidates (organisms) produced by the operators M, C and P in terms of lm (diversification) and ns (intensification) have been experimentally revealed. In order to perform the experimental study, the relationship between the solution candidates (new organisms) produced by the operators and the X_{best} organism has been investigated. In the experimental study, Griewank Function has been chosen as the test problem. Mathematical expression of this benchmark can be given as follow.

$$f(x) = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \qquad 8$$

As can be observed from Fig. 2, Griewank has a large number of local minimum points. In addition, this benchmark problem makes the solution difficult because the local minimum points are close to each other. Hence, this benchmark problem allows the heuristic algorithms to be tested effectively in terms of ns and lm.

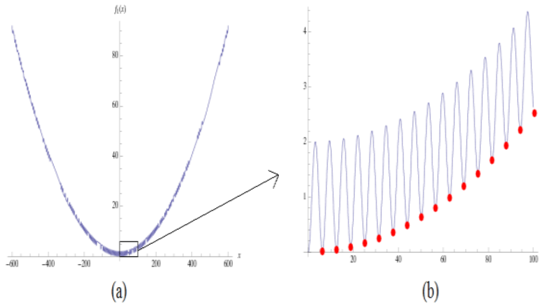


Fig. 2. Curve of Griewank Function (a) and an image of a range of values taken over this curve (b)

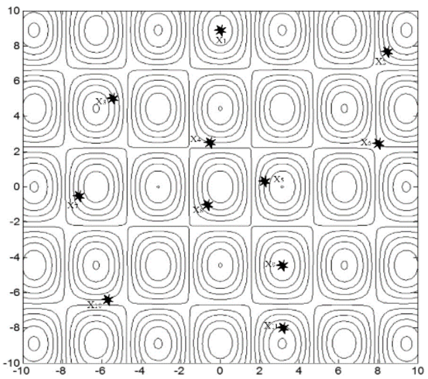


Fig. 3. Distribution of organisms in solution space

The global minimum point of the Griewank benchmark problem given in Fig. 3 is the solution where X_i values are 0. Let's examine the performance of the SOS algorithm operators (M, C, P) in terms of lm and ns on this benchmark problem. First, the number of ecosystem size / organisms should be determined. In the experimental study, the relationship between these organisms and X_{best} will be examined. The ecosystem size has been determined as 11 in the experiment. Organisms in ecosystem size have been created randomly. The search space for the problem has been bounded in the range [-10, 10].

Random distributions of organisms in search space have been plotted on a contour plots chart (please see Fig. 3). In the counter plots graphic, the increase in the number of rings means that the value of fitness increases. In cost problems, the smallest fitness value is the global minimum. The cavity point of the rings indicate the regional minimum points. While the one having the smallest value of these points is the global minimum point, others are local minimums. The global minimum of the Griewank benchmark problem in two-dimensional space solution of (0, 0). Cavities other than this cavity correspond to local minimum points. As can be seen in Fig. 3, the closest organisms to the global minimum are X_4, X_5 and X_8. The closest of them will be the X_{best} organism. In the following subsections, the X_{best} organism is determined first. Then, it is examined how close the other organisms exposed to the operators M, C, P are to X_{best} and how different from are they from X_{best}. Mathematical calculations has been given to illustrate the movement limits of organisms exposed to operators. These calculations have been carried out by putting limit values and random values in the Eqs. belonging to the operators. Thus, the effect of each operator has been revealed both with experimental results and with mathematical results. In addition, the values obtained from experiments or mathematical calculations have been shown on the graph. Thus, organism movements can be monitored easily.

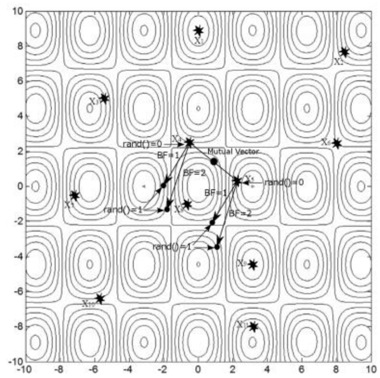


Fig. 4. Effect of Mutualism operator on X_4 and X_5 organisms

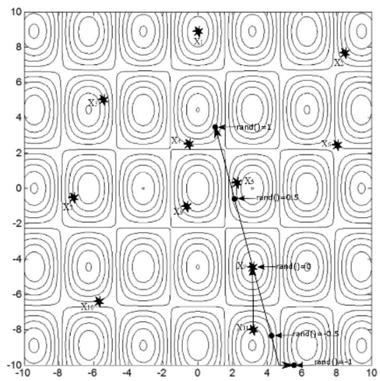


Fig. 5. Effect of Commensalism operator on X_9

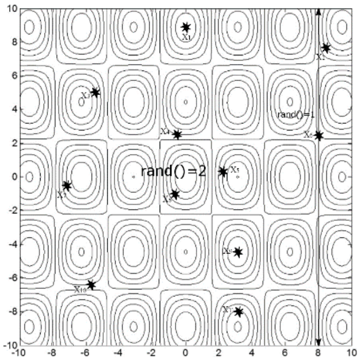


Fig. 6. Effect of Parasitism Operator on X_6 Organism

The effect of the Mutualism operator on organisms (X_4 and X_5) according to the obtained values is given in Fig. 4. The number generated ranndomly by the Rand variant expresses the size / distance of the movement of the organism. If Rand produces 0, the organism remains in the same position, but if rand produces 1, the organism will go to the farthest position where it can go. For numbers generated between 0 and 1, the movement distance will be directly proportional to this number. In the condition of BF coefficient is 1 or 2, the movement direction of the organism is defined. In all cases, however, the organisms move to the X_8 direction, that is X_{best}. This proves that the Mutualism operator makes a neighborhood search. In other words, M operator carries out ns duty.

The effect of the Commensalism operator on organism X_9 is shown in Fig. 5. The random number generated by the Rand variant expresses the movement magnitude and direction of the organism. If the Rand variant produces positive, the direction is parallel to the line drawn from X_{11} organism to X_8 (X_{best}) organism. But, if the Rand produces negative, the direction is parallel to the line drawn from X_8 organism to X_{11} organism. The magnitude of the Rand variant is proportional to the magnitude of the motion. The organism does not always move in the X_8 direction. The X_9 organism is at the local minimum. It escapes from this point with the commensalism operator as can be seen from Fig. 5.

Fig. 6 shows how the parasitism operator affects the X_6 organism. The number produced by the Rand variable indicates the area of motion. If Rand variant produces 1, then the organism can move in a one dimensional space; if Rand produces 2, it can move in a two dimensional space. Here the solution space is two-dimensional, so it shows that the rand variant produces 2 and can go anywhere in the solution space. With Parasitism operator effect, if a better organism is obtained from the X_6 organism than the X_3 organism, this organism replaces him by killing the X_3 organism. Parasitism operator was created to provide diversity. However, it has been determined with experimental studies that the parasitism operator does not function. In the previous experiment, the effects of M, C, P operators on the Griewank benchmark problem have been investigated. With this investigation, it has been determined that the Mutualism operator makes a neighborhood search, the Commensalism operator provides the diversity / escaping from local minimum traps and the Parasitism operator doesn't perform its functions.

This experiment has been applied by extending to other benchmark problems. Thus, the organizational effects of M, C, P operators on all benchmark problems have been investigated experimentally. The distances given in Eq. 7 have been calculated for this process. In this experiment, it is desired how much operators approach organisms to X_{best}. Operators have been tested on 25 benchmark problems. Because 1 benchmark problem produced very high values, it caused an incomprehensible graphic. So it has not been included in the test process. The distances in the experiment have been calculated as given in Eq. 6. For definitions related to this experiment, see back to Case-3 in section "3.2 Experimental set-ups".

The results obtained from this experiment are given in Fig. 7. The X_{best} organism is represented on the 0-axis. It can be clearly seen from Fig. 7, it is clear that the organism, which research the organism closest to the X_{best}, is created by the Mutualism operator in all of the benchmark problems. In other words, the organism exposed to the M operator makes searching in the neighborhood of the X_{best} organism. This is evidence supporting the M operator makes neighborhood search.

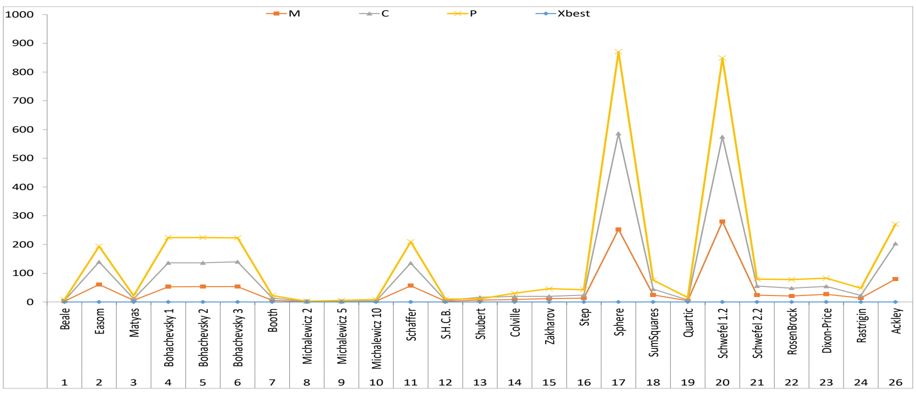


Fig 7. Distances of solution candidates produced by M, C, P operators to X_{best}

The parasitism operator creates organisms that are not similar to the X_{best} organism. This is evidence that Parasitism's operator aims to provide diversity.

4. CONCLUSIONS AND DISCUSSIONS

In this paper, several experimental methods and approaches have been developed to design, test and manage meta-heuristic algorithms. The developed methods have been successfully applied on the SOS algorithm. Thus, important information about the four research questions for the SOS algorithm has been obtained. Discussions and conclusions on the mentioned research topics are given below.

One of the important results obtained from the experimental work is the discovery that the P operator does not fulfill the aimed function. Namely, the Parasitism is based on the fact that one of the organisms is damaged while the other benefits. However, in the experimental studies, none of the organisms exposed to P-operator have benefited. This is clearly shown in Table 1 or in Fig. 1. As a result, SOS algorithm cannot imitate nature in terms of P operator. Therefore, the P-operator needs to be redesigned in accordance with its function in the nature.

Another important result obtained in the article study is the discovery of the functions of the M and C -operators. When Table 1, Figs. 4 and 7 are examined, it is understood that M operator makes a neighborhood search. It is believed that making the neighborhood search more precise (fine tuning) is possible by increasing BF1 and BF2 coefficients given in Eqs. 2 and 3. This feature shows that the neighborhood search in the SOS algorithm can be partially managed through benefit factors. Besides, the biggest disadvantage of M is that it cannot escape from the local minimum. The C-operator is able to get rid of the local minimum traps because it searches with big jumps when he was working alone. However, when C-operator works alone, it also jumps the global minimum. In addition, when used in conjunction with operator M, in many problems (see Table 1), it ensures escaping from local minimums and finding optimum values. Finally, things to do for operators M and C are:

i) It must be ensured that M realizes the neighborhood search with an adaptive strategy. M should be able to a precise local (neighborhood) search with a small step, when necessary. Self-renewing operators are needed just as viruses gain resistance against antibiotics. In order to achieve this, an approach can be developed that changes the BF coefficients depending on the fitness value of the organisms during the search process. It should be paid attention that M's task is to perform “ns” in a controlled way.

ii) The operator C should be able to escape the local minimum traps, but not jump the global solution. Current state of the operator C is static. That is, it does not have any parameters that allow it to control the search process in terms of lm and ns. Making this static structure dynamic will contribute to the manageability of the process. A control parameter can be added to the operator C given in Eq. 4 for this purpose. The amount of jump can be changed via this control parameter. Changing this coefficient dynamically with an adaptive strategy can also contribute to the manageability of the search process.

iii) The operator P must be reconstructed. In other words, the operator P must be redesigned in a way that to produce a damaged and a beneficial organism in accordance with the function of nature. It is essential that one of the solution candidates produced by the operator P must be absolutely more successful. This is a requirement of the situation in the nature because the main task of the operator P is to ensure diversity in the ecosystem. It should be possible to produce successful new organisms naturally while providing diversity.

The three items mentioned above briefly show that the SOS algorithm is not flexible in terms of design and coding. This situation restricts the manageability of the search process, especially in terms of -lm and -ns. In many other heuristic algorithms, it can be seen that neighborhood search and getting rid of local minima have been made manageable. For example, a number of crossover methods have been developed for GA [4]. The crossover operator can apply these methods using the user-defined strategies. In this way, searches can be made in the neighborhood of parental individuals. So it is possible to say that neighborhood search in GA is a manageable process in this way. In addition, the crossover rate similarly provides manageability to process in terms of ns. In addition to that, it can be said that the manageability of the process is also available in terms of lm, in GA. Therefore, a user-defined mutation coefficient is used in GA. Similarly, in the ABC [11] algorithm, neighborhood searches are performed through scout bees sent to the elite bee environment. In the ABC algorithm, environmental size is the value that makes the neighborhood search manageable. By changing this value, the user gains control chance in terms of ns. The management of the lm process in the ABC algorithm is partly user-controlled. The lm process in the ABC is carried out by onlooker bees. The user can only change the number of onlooker bees in this process. Namely, it is not possible to talk about a dynamic strategy. In spite of all these, the SOS gives better results than GA and ABC. Main reason of this is that optimum values of the user-defined control parameters in the GA and ABC algorithms are unknown. Additionally, optimum values of control parameters may vary within the search process. This change can only be achieved by making the parameter values dynamic. This requires the search process to be managed with an adaptive strategy. The values given to these user-defined parameters are extremely effective in the success of the search process. Thus, ensuring manageability of an algorithm with a static approach may not always produce good results. The level of expertise of users is highly effective in describing control parameters values. Perhaps the most important factor in the success of the SOS algorithm or in producing stable results is that it is not user manageable. In addition to this, the success of the bacteria in the nature and its benefits on the SOS algorithm should not be forgotten. As a result, the successful design of the M and C operators, in part, enables the SOS to find optimal solutions. In future studies, it is expected that the SOS algorithm can provide more optimal solutions with (i) designing operators M and C with optimum coefficients (ii) making the solution candidates produced by these operators manageable automatically with dynamic and adaptive strategies (iii) designing in a way that will allow the intervention of expert designers when necessary.

REFERENCES

[1] Blum C. and Roli A., "Metaheuristics in combinatorial optimization: Overview and conceptual comparison", ***ACM Computing Surveys***, 35(3): 268-308, (2003).

[2] Yang X.S., ***Scholarpedia***, 6(8):1147, (2011).

[3] Lin, A., Sun, W., Yu, H., Wu, G., & Tang, H., "Global genetic learning particle swarm optimization with diversity enhancement by ring topology", ***Swarm and evolutionary computation***, 44: 571-583, (2019).

[4] Ali, A. F., Tawhid, M. A., "A hybrid particle swarm optimization and genetic algorithm with population partitioning for large scale optimization problems", ***Ain Shams Engineering Journal***, 8(2): 191-206, (2017).

[5] Dorigo M., "Optimization, Learning and Natural Algorithms", PhD thesis, Politecnico di Milano, (1992).

[6] Chen, Z., Zhou, S., & Luo, J., "A robust ant colony optimization for continuous functions", ***Expert Systems with Applications***, 81: 309-320, (2017).

[7] Dorigo M. and Stützle T., "Ant Colony optimization", ***MA: MIT Press,*** Cambridge, (2004).

[8] Jana, B., Mitra, S., & Acharyya, S., "Repository and Mutation based Particle Swarm Optimization (RMPSO): A new PSO variant applied to reconstruction of Gene Regulatory Network", ***Applied Soft Computing***, 74: 330-355, (2019).

[9] Mason, K., Duggan, J., & Howley, E., "A meta optimisation analysis of particle swarm optimisation velocity update equations for watershed management learning", ***Applied Soft Computing,*** 62: 148-161, (2018).

[10] Rajabioun R., "Cuckoo Optimization Algorithm", ***Applied Soft Computing***, 11:5508-5518, (2011).

[11] Harfouchi, F., Habbi, H., Ozturk, C., & Karaboga, D., "Modified multiple search cooperative foraging strategy for improved artificial bee colony optimization with robustness analysis", ***Soft Computing***, 1-24, (2017).

[12] Karaboga, D., Kaya, E., "An adaptive and hybrid artificial bee colony algorithm (aABC) for ANFIS training", ***Applied Soft Computing***, 49: 423-436, (2016).

[13] Sun, G., Ma, P., Ren, J., Zhang, A., & Jia, X., "A stability constrained adaptive alpha for gravitational search algorithm", ***Knowledge-Based Systems***, 139: 200-213, (2018).

[14] Kahraman H. T., Sagiroglu S., and Colak I., "The development of intuitive knowledge classifier and the modeling of domain dependent data", ***Knowledge Based Systems***, 37: 283-295, (2013).

[15] Rao R.V., Savsani V.J, and Vakharia D.P., "Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems", ***Computer-Aided Design***, 43: 303–315, (2011).

[16] Cheng M.Y., and Prayogo D., "Symbiotic Organisms Search: A new metaheuristic optimization algorithm", ***Computers and Structures***, 139: 98–112, (2014).

[17] Trana D.H., Cheng M.Y., and Prayogo D., "A novel Multiple Objective Symbiotic Organisms Search (MOSOS) for time–cost–labor utilization tradeoff problem", ***Knowledge-Based Systems***, 94: 132–145, (2016).

[18] Meng A., Li, Z., Yin, H., Chen, S., and Guo, Z., "Accelerating particle swarm optimization using crisscross search", ***Information Sciences***, 329: 52–72, (2016).

[19] Wang Z., Xing H., Li T., Yang Y., Qu R., and Pan, Y., "A Modified Ant Colony Optimization Algorithm for Network Coding Resource Minimization", ***IEEE Transactions on Evolutionary Computation***, 20(3): 325-342, (2016).

[20] Lin Q., Chen J., Zhan Z.H., Chen W.N., Coello C.A.C., Yin Y., Lin C.M., and Zhang J., "A Hybrid Evolutionary Immune Algorithm for Multiobjective Optimization Problems", ***IEEE Transactions on Evolutionary Computation***, 20(5): 711-729, (2016).

[21] Seçkiner S.U., Eroğlu Y., Emrullah M., and Dereli T., "Ant colony optimization for continuous functions by using novel pheromone updating", ***Applied Mathematics and Computation***, 219(9): 4163-4175, (2013).

[22] Civicioglu P., "Backtracking search optimization algorithm for numerical optimization problems", ***Applied Mathematics and Computation***, 219(15): 8121-8144, (2013).

[23] Topal A. O., and Altun O., "A novel meta-heuristic algorithm: Dynamic Virtual Bats Algorithm", ***Information Sciences***, 354: 222-235, (2016).

[24] Baykasoğlu A., and Akpinar Ş., "Weighted Superposition Attraction (WSA): A swarm intelligence algorithm for

optimization problems–Part 1: Unconstrained optimization", ***Applied Soft Computing***, 56: 520-540, (2017).

[25] Özkış A., and Babalık A., "A novel metaheuristic for multi-objective optimization problems: The multi-objective vortex search algorithm", ***Information Sciences***, 402: 124-148, (2017).

[26] Babalik A., Ozkis A., Uymaz S.A., and Kiran, M.S., "A multi-objective artificial algae algorithm", ***Applied Soft Computing***, 68: 377-395, (2018).

[27] Aydilek İ.B., "A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems", ***Applied Soft Computing***, 66: 232-249, (2018).

[28] Melki G., Kecman V., Ventura S., and Cano A., "OLLAWV: OnLine Learning Algorithm using Worst-Violators", ***Applied Soft Computing***, 66: 384-393, (2018).

[29] Zhang J., Xiao M., Gao L., and Pan Q., "Queuing search algorithm: A novel metaheuristic algorithm for solving engineering optimization problems", ***Applied Mathematical Modelling***, 63: 464-490, (2018).

[30] Trunfio G.A., Topa P., Was J., "A new algorithm for adapting the configuration of subcomponents in large-scale optimization with cooperative coevolution", ***Information Sciences***, 372: 773–795, (2016).

[31] Karafotias G., Hoogendoorn M., and Eiben A.E., "Parameter Control in Evolutionary Algorithms: Trends and Challenges", ***IEEE Transactions on Evolutionary Computation***, 19(2): 167-187, (2015).

[32] Sun G., Zhang A., Yao Y., and Wang Z., "A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding", ***Applied Soft Computing***, 46: 703–730, (2016).

[33] Sahin, O., Akay, B., "Comparisons of metaheuristic algorithms and fitness functions on software test data generation", ***Applied Soft Computing***, 49: 1202–1214, (2016).

[34] Gupta, S., & Deep, K. "A novel random walk grey wolf optimizer", ***Swarm and evolutionary computation***, 44: 101-112, (2019).