A NEW MULTI-OBJECTIVE ARTIFICIAL BEE COLONY ALGORITHM FOR MULTI-OBJECTIVE OPTIMIZATION PROBLEMS

Züleyha YILMAZ ACAR[1,*], Fikri AYDEMİR[2], Fatih BAŞÇİFTÇİ[1, +]

[1]Selçuk University, Faculty of Technology, Department of Computer Engineering, Konya Turkey

[2]Information Technologies R D Center Kuveyt Turk Participation Bank, Konya Turkey

[*]zuleyhayilmaz@selcuk.edu.tr

Abstract

Since real-world problems have multi-objective optimization problems, algorithms that solve such problems are getting more important. In this study, a new multi-objective artificial bee colony algorithm is proposed for solving multi-objective optimization problems. With the proposed algorithm, non-dominated solutions are kept in the fixed-sized archive. It has benefited from the crowding distance during the selection of elite solutions in the archive. Moreover, the onlooker bees are selected from the archive members with the proposed algorithm. It is aimed to improve the archive members with this modification. To evaluate the performance of the proposed algorithm, ZDT1, ZDT2 and ZDT3 from ZDT family of benchmark functions were used as multi-objective benchmark problems and the results were compared with MOPSO and NSGA-II algorithms. The results show that the proposed algorithm is an alternative method for multi-objective optimization problems.

Keywords: Optimization, multi-objective optimization, artificial bee colony algorithm, swarm intelligence

1. Introduction

Many of the problems in the real-world are defined as problems with more than one and often conflicting goals [1]. Since achieving these goals is an optimization process; such a problem is called a multi-objective optimization problem (MOP). A general MOP can be expressed by (1)–(3):

[+] This paper has been presented at the ICENTE'18 (International Conference on Engineering Technologies) held in Konya (Turkey), October 26-28, 2018.

$$F(x) = \{f_1(x), f_2(x), \dots, f_m(x)\}, \quad (1)$$
$$x = (x_1, x_2, \dots, x_d) \;\&\; m > 1$$

Constraints:

$$h_i(x) < 0 \; for \; i = 1, 2, \dots, I \quad (2)$$
$$g_i(x) = 0 \; for \; i = 1, 2, \dots, J \quad (3)$$

where x is decision vector with d dimensions; $F(x)$ is a set of objective functions; $h(x)$ and $g(x)$ are inequality and equality constraints of the problem.

Moreover, there are some concepts in multi-objective optimization: *Pareto-dominance:* It is a method used to compare two solutions. To say that the solution a dominates the solution b, the solution a must not be worse for all objective functions and be good from the solution b for at least one objective function. *Pareto-optimal:* If there is no solution that dominates the solution a, the solution a is called pareto-optimal solution. *Pareto-optimal set:* This set consists of pareto-optimal solutions.

In this process, optimization algorithms which provide many alternative solutions to decision-makers are used to solve these problems. In this study, artificial bee colony (ABC) algorithm is used to solve MOP. ABC algorithm is a popular algorithm proposed for numerical problems in 2005 [2]. Due to its easy applicability and low parameters, it has become a frequently used algorithm for solving optimization problems [3]. The ABC algorithm showed superior performances when compared with other algorithms known for solving single-objective problems. Along with single-objective problems, the literature suggests that the ABC algorithm is proposed for MOPs [4-10].

In this work, a new multi-objective ABC (MOABC) algorithm is proposed for MOPs. The proposed algorithm is applied on ZDT1, ZDT2 and ZDT3 from ZDT family benchmark functions and the results are compared with MOPSO [11] and NSGA-II [12] algorithms from other multi-objective optimization algorithms.

2. Materials and Methods

2.1. Multi-Objective Artificial Bee Colony Algorithm (MOABC)

ABC algorithm has been proposed by Karaboğa in 2005 [2]. This algorithm, which consists of three artificial bees, includes employed, onlooker and scout. The employed bees bring nectar to their hives from food sources and share the obtained information about the sources with the other bees in the hive. The onlookers select a food in the light of this information. In the algorithm, the exhaustion status of the sources is kept in the trial counter. If the counter of a source used by employed bee has reached the predetermined limit value, the employed bee is called as the scout bee and search for new source.

In the MOABC algorithm proposed in this work, initial solutions are generated firstly. Among these solutions, non-dominated solutions are kept in an archive. Improvement of the current solutions is provided by employed bee stage. New solutions are found in (4):

$$v_{ij} = x_{ij} + \delta * (x_{ij} - x_{kj}) \quad (4)$$

where x_{ij} is current solution, x_{kj} is neighbor solution $(i \neq k)$ and v_{ij} is new candidate solution. δ is a random value in the range $[-1,1]$.

A new solution is selected by applying greedy selection method between the candidate solution and current solution. If the selected solution is the current solution, the trial counter is incremented. If it is the new solution, the counter is reset to zero. The obtained solutions are compared with each AR member from the archive and the archive is updated. This ensures that up-to-date solutions are retained in the archive. The employed bee stage and archive update procedure are shown by $Algorithm\ 1$ and $Algorithm\ 2$, respectively.

$For\ i = 1\ to\ PopulationNumber/2$
  $Select\ randomly\ a\ dimension\ j\ and$
   $neighbor\ k\ from\ the\ population\ (i \neq k)$
  $v_{ij} = x_{ij} + \delta * (x_{ij} - x_{kj})$
   $If\ v_i\ dominates\ x_i$
    $x_i = v_i$
    $trial_i = 0$
   $Else\ if\ x_i dominates\ vi$
    $trial_i = trial_i + 1$
   $Else\ if\ v_i and\ x_i are\ non-dominated\ solutions$
    $If\ rand < 0.5$
     $x_i = v_i$
     $trial_i = 0$

$Else$
$trial_i = trial_i + 1$
$End\ If$
$End\ If$
$UpdateArchive(v_i, AR)$
$End\ For$

$Algorithm\ 1. Employed\ Bee\ Stage$

$For\ i = 1\ to\ length(AR)$
$If\ v\ dominates\ AR_i$
$Add\ v\ to\ the\ archive$
$Else\ if\ AR_i dominates\ v$
$Do\ nothing$
$Else\ if\ ARi\ and\ v\ are\ nondominated\ solutions$
$Add\ v\ to\ the\ archive$
$End\ If$
$End\ For$

$Algorithm\ 2. Archive\ Update\ Procedure$

The onlooker bees in the MOABC are accepted as archive members in contrast to the basic ABC algorithm, and another archive member is used to improve an archive member. This process is represented by (5):

$$v_{ij} = x_{ij} + \delta * (x_{ij} - x_{kj}) \tag{5}$$

where the candidate solution v_{ij} in the onlooker bee stage is produced by using AR_{ij} which is an archive member, and AR_{kj} is a neighbor archive member $(i \neq k)$. When a neighbor archive member is selected, the crowding distance (CD) values of all archive members are calculated [12] and the member AR_{kj} with the lowest CD value is selected. The δ is a random value in the interval $[-1,1]$.

The current solution is an archive member, and the candidate solution is produced by using the archive members as in (5). The archive update procedure shown in $Algorithm$ 1 is used between these two solutions. Along with this process, it is aimed to increase the local search ability of the algorithm in the archive members. The onlooker bee stage is shown in $Algorithm$ 3.

$For\ i = 1\ to\ length(AR)$
$Select\ randomly\ a\ dimension\ j$
$Select\ neighbor\ k\ from\ the\ archive$
$(i \neq k) according\ to\ crowding\ distance\ values$
$v_{ij} = AR_{ij} + \delta * (AR_{ij} - AR_{kj})$
$UpdateArchive(v_i, AR)$
$End\ For$

$Algorithm\ 3. Onlooker\ Bee\ Stage$

A fixed-sized archive is used in the MOABC algorithm. When the archive is updated, the archive size is controlled. If the size reaches a predetermined value, elite archive members are kept in archive. CD values are used in the selection of elite members.

As in the basic ABC, in the MOABC algorithm, the trial counters of the food sources are controlled in the scout bee stage. If there is a food source that reaches the predetermined limit value, the new position is determined for that. There can be only one scout bee in every cycle. The stopping criterion is the number of evaluations. When the stopping criterion is satisfied, operation of the algorithm is terminated, and the current archive is returned as a result. The scout bee stage is represented by $Algorithm\ 4$:

$Determine\ only\ one\ x_i\ solution\ with\ maximum\ trial\ in\ the\ population$
$If\ trial_i \geq limit$
$Generate\ initial\ position\ for\ x_i$
$trial_i = 0$
$End\ If$

$Algorithm\ 4. Scout\ Bee\ Stage$

3. Experiments

The proposed MOABC algorithm is compared with MOPSO and NSGA-II algorithms based on the test functions using performance metric results.

3.1. Test Functions

In this study, ZDT1, ZDT2 and ZDT3 functions were used as test functions. ZDT test functions [13] are widely used in evaluating the performance of multi-objective optimization algorithms. The problems used are MOPs with unconstraint two objectives.

3.2. Performance Metric

The Inverted Generational Distance (IGD) [14] metric was used to evaluate the performance of the MOABC algorithm. With this metric, both the diversity and the convergence of the algorithm are examined. The metric calculates the distance between the true pareto-front and the obtained pareto-front by the MOABC algorithm. The low IGD value indicates the success of the algorithm. IGD value is expressed as (6):

$$IGD(OPF,TPF) = \frac{\sum_{i \epsilon TPF} d(i,OPF)}{|TPF|} \quad (6)$$

where TPF is true pareto-front, OPF is obtained pareto-front by the MOABC algorithm. $d(i,OPF)$ is minimum Euclidean distance. $|TPF|$ is number of TPF solutions.

3.3. Parameter Settings

The decision variable number for the test functions is set to 30. The number of population is 50, and the maximum number of evaluations is 1,0E+5. The results obtained 10 independent runs. Additionally, the limit value for the food sources is set to 5 in the MOABC algorithm. In this study, the results of the MOPSO and NSGA-II algorithms used for comparison were obtained from the PlatEMO platform (which can be downloaded from link: http://bimk.ahu.edu.cn/index.php?s=/Index/Software) [15].

4. Results and Discussion

The results obtained from the MOPSO, NSGA-II and MOABC algorithms proposed in this study for ZDT1, ZDT2 and ZDT3 test functions are shown in Table 1. Average IGD values and standard deviation values are included in the table.

Table 1. IGD and Standard Deviation Values for ZDT1, ZDT2 and ZDT3

Test Functions	Error Values	MOPSO	NSGA-II	MOABC
ZDT1	Mean	4,96E+1	1,09E-2	7,63E-3
	Std	1,37E+1	6,65E-4	7,22E-4
ZDT2	Mean	6,18E+1	1,12E-2	1,05E-1
	Std	9,80E+0	8,67E-4	1,53E-1
ZDT3	Mean	5,59E+1	1,83E-2	9,50E-3
	Std	1,15E+1	1,15E-2	1,27E-3

Table 1 shows performance comparisons for MOPSO, NSGA-II and the proposed MOABC algorithms for ZDT1, ZDT2 and ZDT3 functions. As can be seen, the MOABC algorithm achieved better results than MOPSO and NSGA-II algorithms for ZDT1 and ZDT3. For ZDT2, while the proposed MOABC algorithm yields better results than the MOPSO algorithm, the algorithm that achieves the best result is NSGA-II. The distributions of the solutions obtained by MOABC algorithm on the true pareto-front are shown in Figures (1)–(3).

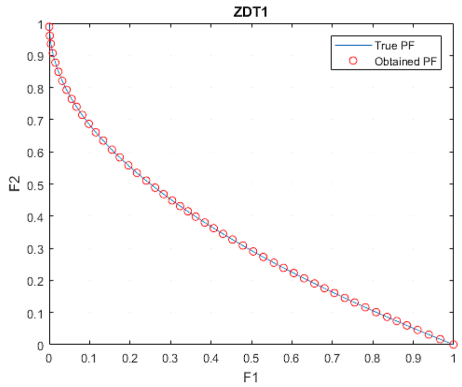

Figure 1. Pareto-front of MOABC algorithm on ZDT1 function

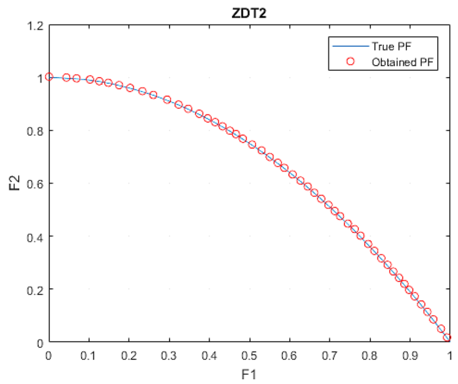

Figure 2. Pareto-front of MOABC algorithm on ZDT2 function

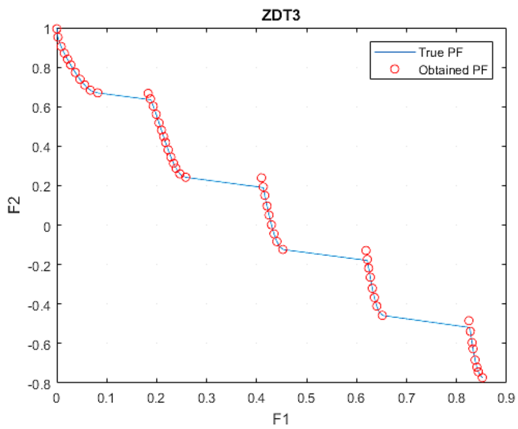

Figure 3. Pareto-front of MOABC algorithm on ZDT3 function

As shown in Figures (1)–(3), the MOABC algorithm showed a good distribution on the true Pareto-front with the solution variability. It seems that the non-dominated solutions obtained by the MOABC algorithm cover the true pareto-front.

5. Conclusion

In this study, MOABC was proposed to solve MOPs. In the proposed algorithm, an improvement has been made in the onlooker be stage, and the results were compared with the MOPSO and NSGA-II algorithms. Three functions of the ZDT benchmark family were selected as the test function. IGD metric was used as performance metric. It is seen that the proposed MOABC algorithm is an alternative solution method to solve MOPs. In future works, the performance of MOABC algorithm can be evaluated in other test and engineering problems.

References

[1] Konak A., Coit D.W., and Smith A.E., Multi-objective optimization using genetic algorithms: A tutorial, Reliability Engineering & System Safety 2006; 91: 992-1007.

[2] Karaboga D., An idea based on honey bee swarm for numerical optimization, Technical report-tr06, Erciyes university, engineering faculty, computer engineering department 2005.

[3] Karaboga D., Gorkemli B., Ozturk C., and Karaboga N., A comprehensive survey: artificial bee colony (ABC) algorithm and applications, Artificial Intelligence Review 2014; 42: 21-57.

[4] Gong D., Han Y., and Sun J., A novel hybrid multi-objective artificial bee colony algorithm for blocking lot-streaming flow shop scheduling problems, Knowledge-Based Systems 2018; 148: 115-130.

[5] Sanchez-Gomez J.M., Vega-Rodríguez M.A., and Pérez C. J., Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach, Knowledge-Based Systems 2017; 159: 1-8.

[6] Saad A., Khan S.A., and Mahmood A., A multi-objective evolutionary artificial bee colony algorithm for optimizing network topology design, Swarm and Evolutionary Computation 2018; 38: 187-201.

[7] Pérez C.J., Vega-Rodríguez M.A., Reder K., and Flörke M., A Multi-Objective Artificial Bee Colony-based optimization approach to design water quality monitoring networks in river basins, Journal of Cleaner Production 2017; 166: 579-589.

[8] Kishor A., Singh P.K., and Prakash J., NSABC: Non-dominated sorting based multi-objective artificial bee colony algorithm and its application in data clustering, Neurocomputing 2016; 216: 514-533.

[9] Dwivedi K., Ghosh S., and Londhe N.D., Low power FIR filter design using modified multi-objective artificial bee colony algorithm, Engineering Applications of Artificial Intelligence 2016; 55: 58-69.

[10] Ding M., Chen H., Lin N., Jing S., Liu F., Liang X., and Liu W., Dynamic population artificial bee colony algorithm for multi-objective optimal power flow, Saudi Journal of Biological Sciences 2017; 24: 703-710.

[11] Coello C.A.C. and Lechuga M.S., MOPSO: a proposal for multiple objective particle swarm optimization, *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, Honolulu, HI, USA, 1051-1056.

[12] Deb K., Pratap A., Agarwal S., and Meyarivan T., A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 2002; 6: 182-197.

[13] Zitzler E., Deb K., and Thiele L., Comparison of Multiobjective Evolutionary Algorithms: Empirical Results, Evolutionary Computation 2000; 8: 173-195.

[14] Zitzler E., Thiele L., Laumanns M., Fonseca C.M., and Fonseca V.G., Performance assessment of multiobjective optimizers: an analysis and review, IEEE Transactions on Evolutionary Computation 2003; 7: 117-132.

[15] Tian Y., Cheng R., Zhang X., and Jin Y., PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization [Educational Forum], IEEE Computational Intelligence Magazine 2017; 12: 73-87.