Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 48 (2015) 22 – 28

Procedia Computer Science

International Conference on Intelligent Computing, Communication & Convergence

(ICCC-2015)

Conference Organized by Interscience Institute of Management and Technology,

Bhubaneswar, Odisha, India

Differential Search Algorithm for Multiobjective Problems

Vijay Kumar[a], Jitender Kumar Chhabra[b], Dinesh Kumar[c,] a*

"a Computer Science & Engineering Department, Manipal University, Jaipur, Rajasthan, India"
[b] *Computer Engineering Department, National Institute of Technology, Kurukshetra, Haryana, India*
[c] Computer Science & Engineering Department, GJUS&T, Hisar, Haryana, India

Abstract

In this paper, a novel Differential Search Algorithm (DSA) approach is proposed to solve multiobjective optimization problems, called Multiobjective Differential Search Algorithm (MODSA). MODSA utilizes the concept of Pareto dominance to determine the direction of a super-organism and it maintains non-dominated solutions in the external repository. This approach also uses the external repository of super-organisms that is used to guide other super-organisms. It guides the artificial organisms to search towards non-crowding and external regions of Pareto front. The performance of proposed approach is evaluated against the other well-known multiobjective optimization algorithms over a set of multiobjective benchmark test functions. Experimental results reveal that the MODSA outperforms the other competitive algorithms for benchmark test functions.

Peer-review under responsibility of scientific committee of International Conference on Computer, Communication and Convergence (ICCC 2015)

*Keywords:*Multiobjective optimization; Metaheuristic; Pareto Optimal; Differential search algorithm

* Corresponding author. Tel.: +91-9466255106.
E-mail address: vijaykumarchahar@gmail.com

Peer-review under responsibility of scientific committee of International Conference on Computer, Communication and Convergence (ICCC 2015)
doi:10.1016/j.procs.2015.04.105

1. Introduction

Multi-objective optimization is the process of simultaneously optimizing two or more conflicting objectives. Real-life problems contain more than one conflicting objective function. Hence it requires multi-objective optimization approach. The optimal solution is clearly defined in a single objective function optimization. However, multiobjective optimization problems do not restrict to obtain a single objective function [5]. The multiobjective optimization problems (MOPs) contain a set of solutions called non-dominated solutions. Each solution in the non-dominated set is called as a Pareto-Optimal. The Pareto Optimal solutions are mapped in the objective space that are known as Pareto front [6]. The main objective of multiobjective optimization is to obtain the Pareto front of a given a MOP. Generally, the multiobjective optimization problems are computational intensive as the search space for MOPs is very large. The use of metaheuristic algorithms for multiobjective optimization has significantly grown in the last few years [8]. Some of these metaheuristic algorithms are Simulated Annealing (SA), Genetic Algorithm (GA), Tabu Search (TA), Ant Colony Optimization (ACO), Differential Evolution (DE), Particle Swarm Optimization (PSO), and so on. However, any single metaheuristic may not be the best fit for all problems; rather it may be problem specific. Although many metaheuristic algorithms for solving multiobjective problems have been proposed as reported in literature, the results are unsatisfactory. Hence, an improvement to metaheuristic algorithms for solving multiobjective problems is still required.

Recently, Pinar Civicioglu [2] developed a new metaheuristic search algorithm called Differential Search Algorithm (DSA) for uni-objective optimization. DSA simulates a superorganism migrating between the two stopovers sites. DSA has unique mutation and crossover operators. DSA has only two control parameters that are used for controlling the movement of superorgnisms. DSA has been applied for a variety of applications. Till now, it had not been extended to solve multiple objectives. DSA appears more suitable for multiobjective problems as high speed of convergence and less overhead of parameters setting. In this paper, a novel approach named multiobjective differential search algorithm (MODSA), which allows the DSA to deal with multiobjective optimization problems. MODSA is based on non-dominated sorting strategy. The concept of Pareto dominance is incorporated in MODSA to determine which solution is better. The constraint handling mechanism is added in the MODSA to increase the ability of exploration of DSA. MODSA has been compared with other recently proposed multiobjective metaheuristic algorithms and validated on benchmark test functions.

The reminder of the paper is organized as follows. Section 2 gives a brief description of the previous relevant work. Section 3 gives the basic concept of DSA. The proposed MODSA is presented in Section 4, followed by results and discussion are shown in Section 5. The statistical analysis of the multiobjective optimization algorithms is performed in Section 6. Finally, the conclusions are drawn in Section 7.

2. Differential Search Algorithm

Differential Search Algorithm (DSA) is a novel metaheuristic algorithm proposed by Civicioglu [2]. It mimics the *Brownian-like random-walk movement.* The main motivation of DSA is the migration behaviour of living beings, which move away from a habitat having low capacity of food resources. The migration process entails movement of large number of individuals comprising a superorganism. A superoragnism moves towards habitat having more food capacity. Once it finds new fruitful habitat named as stopover site, it settles in the new habitat for the time being and continues its migration towards more fruitful habitats. DSA starts by generating individuals of respective optimization problem corresponding to an artificial-superorganism. Hereafter artificial-superorganism tries to migrate from its current position to the global minimum value.

In DSA each individual of a superorganism is represented as X_i, $i = 1, 2, ..., N$ and as many members as the dimension of the problem i.e., x_{ij}, $j = 1, 2, ..., D$. Here N and D represent the number of individuals and the size of problem respectively. The DSA consists of the following steps [2, 8]:

Step 1. Initialize the artificial-organism: Each member of a individual (or artificial-organism) is initialized to a random position. This is achieved as follows:

$$x_{ij} = rand \times \left(up_j - low_j \right) + low_j \tag{1}$$

where low_j and up_j represents the lower and upper bounds of j^{th} dimension of respective problem.

Step 2. Initialize the control parameters: The values of p_1 and p_2 are used to determine the frequency of perturbation of members in a position corresponding to an individual. The $Scale$ is used to determine the amount of perturbation in size of position of the members in an individuals. It is generated through a gamma random number generator. The values of p_1, p_2 and $Scale$ are given below:

$$p_1 = 0.3 \times rand_1 \quad and \quad p_2 = 0.3 \times rand_2 \tag{2}$$

$$Scale = randg\left[2 \times rand_3\right] \times \ rand_4 - rand_5 \tag{3}$$

where $randg$ is the gamma random number generator. $rand_1$, $rand_2$, $rand_3$, $rand_4$, and $rand_5$ are the uniform random number generators.

Step 3. Compute the stopover site: During the migration process, *Brownian-like random-walk model* is used to determine the intermediate stopover sites. To explore the stopover sites, randomly selected individuals move towards targets represented by *donor*. The position of stopover site is computed based on randomly selected individual, which is given below:

$$StopoverSite = Superorganism + Scale \times \ donor - Superorganism \tag{4}$$

$$donor = \left[X_{random\_shiffling(i)} \right] \tag{5}$$

Step 4. Compute the participating members: The members participating in the search process are determined through stochastic scheme. The values of p_1 and p_2 are used in stochastic scheme, which is described in [2].

Step 5. Check the limit of stopover site: The stochastic process is completely random. There is a probability that an element of the stopover site is beyond the limit of the habitat. In such a situations, the elements of stopover site are to be maintained within the boundary of the specified search space.

Step 6. Check the termination criterion: If a stopover site is more fruitful than the sources associated with an individual of artificial-organisms, the corresponding individual moves to stopover site. The search for global minimum continues and the individual stops near the intermediate stopover for the time being and then continues its search from the current position. Steps 2 to 6 are repeated until the value of global minimum is reached.

3. Multiobjective Differential Search Algorithm

The main contribution of this paper is the development of a novel multiobjective optimization technique based on DSA. The proposed approach is inspired from multiobjective particle swarm optimization (MOPSO) [4]. In this paper, Pareto dominance is incorporated into differential search algorithm (DSA) to solve problems having number of objective functions. This novel approach uses the external repository which helps to guide the artificial-organism during migration. The proposed approach of the multiobjective optimization using differential search algorithm (MODSA) is given in Section 3.1.

3.1. Proposed Algorithm

Step 1. Initialize the artificial-Superorganism **(** SUP **):**

a) For $i = 1$ to number_ of_ artificial-organism

b) Initialize $SUP\ i$

Step 2. Evaluate each of the artificial-organisms in SUP**.**

Step 3. Store the positions of the artificial-organisms that represent nondominated solutions in the external repository **(** REP**).**

Step 4. Generates hypercubes of the search space which have been explored and locate the artificial-organisms using these hypercubes as a coordinate system where position of each artificial-organism is defined according to its objective functions.

Step 5. Repeat the following until maximum number of iterations has been reached

a) Compute p_1, p_2 and $Scale$ using equations (2) and (3) respectively.

b) Compute random process designated as map for each artificial-organism of the superorganism using stochastic scheme (mentioned in Step 4 of *DSA*).

c) Compute the position for each artificial-organism using the following expression:

$$POS\ i = Scale \times map \times REP\ h - SUP\ i \quad (6)$$

where $SUP\ i$ represents the current value of the artificial-organism i and $REP\ h$ represents a value that is taken from repository. The index h is computed is following manner.
The roulette-wheel selection procedure will select the hypercube. From this selected hypercube, we further select an artificial-organism randomly.

d) Compute the new position of the artificial-organism by adding the position generated from previous step.

$$SUP\ i = SUP\ i + POS\ i \quad (7)$$

e) Maintain the artificial-organism within search space.

f) Evaluate each artificial-organism in SUP.

g) Update the contents of REP: This update mentions the nondominated solutions in repository and eliminates the dominated solutions from the REP. When the REP is full, the artificial-organisms located in less populated areas are preserved.

h) When the position of the artificial-organism in its memory is not better than the current position, the artificial-organism's position is updated with current position. Pareto dominance is applied to decide whether the position from memory should be retained or not.

Step 6. Return the optimal solution.

4. Computational Results

To evaluate the performance of MODSA, seven standard benchmark test functions named as Schaffer (SCH), Fonseca (FON), ZDT1 to ZDT4, ZDT6, and DTLZ2 have been used for experimentation. We have compared the performance of MODSA with the two well-established multiobjective optimization algorithms such as NSGA-II [1] and MOPSO [4]. The results are evaluated and compared using acceptable three performance metrics such as Hypervolume (HV) [9], Spread [9], and Epsilon [3].

4.1. Parameter Setting for the Involved Algorithms

The parameters settings of the methods (NSGA-II and MOPSO) used are tabulated in Table 1. In addition to parameters mentioned in Table 1, the maximum number of generations for NSGA-II, MOPSO and MODSA is fixed as 500. The results have been compared in terms of 'mean' and 'standard deviation' over ten independent runs, each run consisting of 500 generations.

Table 1.Parameter setting for multiobjective optimization algorithms

	Population Size	Mutation Rate	Crossover Rate	Repository Size	Grid Size	C_1	C_2	P_1	P_2
NSGA-II [1]	100	0.8	0.3	-	-	-	-	-	-
MOPSO [4]	100	-	-	100	10	1.50	1.50	-	-
MODSA (Proposed)	100	-	-	100	10	-	-	$0.3\times rand$	$0.3\times rand$

4.2. Results and Discussions

The comparison of the results for seven multi-objective test functions over three algorithms is given in Tables 2-4. For SCH test problem, MODSA outperforms other algorithms in terms of Epsilon, Spread and hypervolume. The results indicate that the MODSA produced a set of solutions that are approximately distributed in the objective space. The MODSA obtained the best results for the FON test problem. It provides better diversity and spread among other competitive algorithms. For ZDT1 and ZDT2 test problems, MODSA provided better results among all competitive algorithms in terms of Epsilon, Spread and Hypervolume. The MOPSO provided a set of solutions that were well converged for ZDT3 and ZDT4 test problems. MODSA provided better spread and hypervolume values than other algorithms for these test problems. The MODSA successfully converged to optimal solution and produced solutions having uniform distribution for ZDT6 and DTLZ2 test problems. Hence the MODSA surpasses other algorithms for solving SCH, FON, ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, and DTLZ2 test problems.

Table 2.Mean and Standard deviation of Epsilon Metric

	SCH	FON	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	DTLZ2
NSGA-II	3.951E+00 (4.63E-02)	9.795E+00 (1.26E-03)	9.754E-01 (1.65E-02)	9.869E-01 (1.34E-02)	8.459E-01 (3.65E-03)	9.849E-01 (2.43E-02)	9.996E-01 (1.26E-03)	**6.811E-01 (1.74E-02)**
MOPSO	3.835E+00 (6.84E-02)	9.474E-01 (1.24E-02)	9.634E-01 (2.41E-02)	9.838E-01 (7.83E-03)	**4.965E-01 (9.82E-02)**	**9.679E-01 (2.51E-02)**	9.885E-01 (2.09E-02)	8.384E-01 (8.19E-02)
MODSA (Proposed)	**3.633E+00 (1.45E-01)**	**9.148E-01 (2.84E-02)**	**9.613E-01 (2.53E-02)**	**9.829E-01 (6.18E-03)**	8.173E-01 (4.44E-02)	9.842E-01 (7.87E-03)	**8.562E-01 (9.99E-02)**	1.096E+00 (6.56E-02)

Table 3.Mean and Standard deviation of Spread Metric

	SCH	FON	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	DTLZ2
NSGA-II	5.000E-01 (0.00E+00)	5.542E-01 (1.32E-02)	5.826E-01 (5.07E-02)	6.137E-01 (6.31E-02)	5.665E-01 (2.46E-02)	5.875E-01 (3.19E-02)	5.179E-01 (1.41E-02)	9.193E-01 (5.31E-02)
MOPSO	5.000E-01 (0.00E+00)	5.508E-01 (1.09E-02)	6.378E-01 (2.71E-02)	6.875E-01 (1.49E-02)	5.974E-01 (9.88E-02)	5.052E-01 (1.59E-02)	5.063E-01 (7.53E-03)	5.258E-01 (2.22E-02)
MODSA (Proposed)	**5.000E-01 (0.00E+00)**	**5.435E-01 (1.38E-02)**	**5.793E-01 (2.21E-02)**	**6.116E-01 (1.52E-02)**	**5.294E-01 (3.13E-02)**	**5.038E-01 (3.35E-02)**	**5.034E-01 (2.55E-03)**	**4.846E-01 (2.13E-02)**

Table 4.Mean and Standard deviation of Hypervolume Metric

	SCH	FON	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	DTLZ2
NSGA-II	5.644E-03 (1.22E-02)	1.219E-01 (2.61E-01)	6.449E-01 (4.13E-01)	4.850E-01 (3.91E-01)	1.741E-01 (4.34E-02)	6.672E-01 (4.37E-01)	**4.828E-01 (1.33E-01)**	2.863E-01 (1.45E-02)
MOPSO	7.829E-01 (1.66E-01)	3.109E-01 (2.21E-01)	5.899E-01 (3.46E-01)	4.048E-01 (2.52E-01)	7.096E-01 (3.30E-01)	5.576E-01 (4.04E-01)	2.704E-01 (1.92E-01)	1.485E-01 (1.46E-01)
MODSA (Proposed)	**8.073E-01 (1.26E-01)**	**5.036E-01 (1.69E-01)**	**7.495E-01 (2.81E-01)**	**6.684E-01 (2.55E-01)**	**8.663E-01 (1.56E-01)**	**7.322E-01 (1.65E-01)**	7.078E-02 (1.03E-01)	**4.592E-01 (2.92E-01)**

4.3. Statistical Analysis

Wilcoxon test was used to find out whether there is a significant difference in the performance of one algorithm compared to other algorithms for each benchmark test function with respect to above-mentioned performance metrics. Tables 5-7 recapitulate the results of Wilcoxon test in Epsilon, Spread and HV performance metrics respectively. In each row, seven benchmark test functions are denoted by a symbol. We have used three symbols: "-" means that there is no statistical significance between algorithms, "⊥" means that the algorithm in the column has produced better results than the algorithm in the row, and "┼" means that the algorithm in the column is statistically better than the algorithm in the row. A look at the Tables 5-7 reveals that the MODSA is statistical better than other algorithms in most of the benchmark test functions.

Table 5.Epsilon Metric: statistical tests for benchmark test problems.

	MOPSO								MODSA							
NSGA-II	┼	┼	⊥	⊥	┼	┼	┼	┼	┼	┼	⊥	⊥	┼	┼	┼	┼
MOPSO									┼	┼	⊥	⊥	┼	┼	┼	┼

Table 6. Spread Metric: statistical tests for benchmark test problems.

	MOPSO								MODSA							
NSGA-II	-	⊥	┼	┼	⊥	┼	┼	┼	-	⊥	⊥	⊥	┼	┼	┼	┼
MOPSO									-	⊥	┼	┼	┼	⊥	┼	┼

Table 7. Hypervolume Metric: statistical tests for benchmark test problems.

	MOPSO								MODSA							
NSGA-II	┼	⊥	⊥	⊥	┼	⊥	┼	┼	┼	┼	⊥	⊥	┼	⊥	┼	┼
MOPSO									⊥	┼	⊥	┼	┼	⊥	┼	┼

5. Conclusions

In this paper, a novel approach for multiobjective differential search algorithm has been proposed. The proposed approach utilized the external repository to store the nondominated artificial-organisms found during search. The

performance of MODSA was compared with recently developed multi-objective algorithms for eight benchmark test functions. The results have been evaluated on the basis of performance metrics characterizing the convergence and diversity. The results reveal that the proposed approach is superior over NSGA-II and MOPSO in terms of convergence and diversity. The statistical test demonstrate the statistical significance of proposed approach. Future research work includes the application of MODSA on automatic cluster evolution and brain image segmentation.

References

1. Deb K, Pratap A, Agarwal S, Mayarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 2002; **6**:182-197.
2. Civicioglu P. Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm, Comput. Geosci. 2012; **46**: 229-247.
3. Durillo JJ, Nebro AJ. jMetal: a Java framework for multi-objective optimization, Adv. Eng. Softw. 2011; **42**: 760-771.
4. Coello Coello CA, Pulido GT, Lechuga MS, Handling multiple objectives with particle swarm optimization, IEEE Trans. Evol. Comput. 2004; **8**: 256-279.
5. Zou W, Zhu Y, Chen H, Zhang B. Solving multiobjective optimization problems using artificial bee colony algorithm. Discrete Dynamics in Nature and Society. 2011;1-37.
6. Zhou A, Qu B-Y, Li H, Zhao S-H, Suganthan PN, Zhang Q. Multiobjective evolutionary algorithms: a survey of the state of art. Swarm and Evolutionary Computation. 2011; **1**:32-39.
7. Figueira JR, Talbi E-G. Emergent nature inspired algorithms for multi-objective optimization, Comput. Oper. Res. 2013; **40(6)**: 1521-1523.
8. Goswami D, Chakraborty S. Differential search algorithm-based parametric optimization of electrochemical micromachining process, Int. J. Industrial Engg. Comput. 2014; **5(1)**: 41-54.
9. Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: empirical results. Evol. Comput. 2000; **8**:173-195.