Journal of AI and Data Mining
Vol 5, No 2, 2017, 183-195

A Hybrid MOEA/D-TS for Solving Multi-Objective Problems

Sh. Lotfi and F. Karimi*

Department of Computer Science, University of Tabriz, Tabriz, Iran.

Received 06 June 2016; Revised 27 November 2016; Accepted 18 January 2017
**Corresponding author: f.karimi@tabrizu.ac.ir (F. Karimi).*

Abstract
In many real-world applications, various optimization problems with conflicting objectives are very common. In this work, we employ Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D), a newly developed method beside Tabu Search (TS) accompaniment to achieve a new manner for solving multi-objective optimization problems (MOPs) with two or three conflicting objectives. This improved hybrid algorithm, namely MOEA/D-TS, uses the parallel computing capacity of MOEA/D along with the neighborhood search authority of TS for discovering Pareto optimal solutions. Our goal is to exploit the advantages of evolutionary algorithms and TS to achieve an integrated method to cover the totality of the Pareto front by uniformly distributed solutions. In order to evaluate the capabilities of the proposed method, its performance based on various metrics is compared with SPEA, COMOEATS, and SPEA2TS on the well-known Zitzler-Deb-Thiele's ZDT test suite and DTLZ test functions with separable objective functions. According to the experimental results obtained, the proposed method could significantly outperform the previous algorithms and produce fully satisfactory results.

Keywords: *Multi-objective Problems, Evolutionary Algorithms, Hybrid Method, MOEA/D, Tabu Search.*

1. Introduction
Multi-objective optimization problems (MOPs) with the aim of optimizing a collection of various objectives, systematically and simultaneously, are among important challenges in the today's world. Unlike single-objective optimization, finding an optimal trade-off among conflicting objectives in a multi-objective problem is often more complex and challenging [1]. Also it is necessary to determine a community of points, which are compatible with a pre-determined definition for an optimum. For trading off between solutions, a vast piece of information about the desired problem is required to opt the best solutions and omit the unwanted ones based on the problem constraints. Typically, a number of potentially Pareto optimal solutions are good candidates as optimal trade-off for these kinds of problems [2].
Many researchers believe that Evolutionary Algorithms (EAs), which make use of the strategy of population evolutionary to optimize the problems, are able to perform better than other blind search strategies confronting MOPs [3-5]. Within the last decade, various techniques have been proposed, which demonstrate the power of Multi-Objective Evolutionary Algorithms (MOEAs) for solving MOPs [7-16]. These kinds of methods can produce a set of Pareto-optimal solutions in a single run using a population of candidate solutions [17]. As an important population-based EA, Genetic Algorithm (GA) is well-suited to solve multi-objective optimization problems. Multi-Objective Genetic Algorithm (MOGA) [18], Niched Pareto Genetic Algorithm (NPGA) [19], and Non-dominated Sorting Genetic Algorithm (NSGA) [5] are among the first efforts to take advantage of GA having specialized fitness functions and various methods to promote solution diversity [8].
One fundamental shortcoming of these methods is the neglect of elitism strategy, which was recognized and supported experimentally in the multi-objective searches a few years later [20, 21]. Strength Pareto Evolutionary Algorithm (SPEA) [21] was one of the first techniques that outperformed the (non-elitist) alternative approaches [21,22]. An improved version of

SPEA, namely SPEA2 [23], is a powerful algorithm with the ability to overcome its predecessor shortcomings and achieve acceptable results. This updated method was the basis of our previous hybrid algorithm, namely Strength Pareto Evolutionary Algorithm2 Tabu Search (SPEA2TS) [24], which uses the exploration capacity of SPEA2 along with the power of TS in neighborhood research to find Pareto optimal solutions in different multi-objective problems.
A majority of the current MOEAs do not employ the decomposition concept. The manner these algorithms adopt is considering the whole MOP, and do not affiliate each separate solution with any particular scalar optimization problem [25]. This idea is adopted by a limited number of MOEAs to a certain amount [26-28], and Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) is the more recent one [25]. MOEA/D transforms the task of approximating the Pareto front (PF) into a number of single-objective optimization sub-problems using the traditional aggregation methods, and then optimizes these sub-problems simultaneously [6]. Considering the best solution found so far (i.e. from the start of algorithm's run) at each generation, the population is composed of each sub-problem. According to the distances between their aggregation coefficient vectors, these sub-problems find the neighborhood relations among them. The only information used for optimization of each sub-problem by MOEA/D comes from its neighbors.
In this work, we improved our earlier work (SPEA2TS) [24] by taking the advantage of MOEA/D as the optimization tool beside the capabilities of Tabu Search for dealing with various multi-objective optimization problems. Our goal was to exploit the advantages of EA and TS to achieve an integrated method to cover the totality of the Pareto front by uniformly distributed solutions.
The structure of this paper is as what follows. Section 2 introduces the main concepts of the multi-objective optimization. Section 3 provides a comprehensive literature review on the different methods used for solving MOPs. The Multi-Objective Evolutionary Algorithm based on Decomposition is described with more details in Section 4, while as a general overview of our proposed method, MOEA/D-TS is available in Section 5. Section 6 provides the experimental settings that are used in Section 7 to elaborate the experimental results for selected benchmark problems. Finally, a brief summary and conclusion are provided in Section 8.

2. Multi-objective optimization

We could define a multi-objective optimization problem as follows [6]:

$$\begin{aligned} &\text{Max } F(x)=\left(f_1(x),f_2(x),\dots,f_m(x)\right) \\ &\text{Subject to} \\ &\quad g_i(x)\le 0, \qquad i=1,2,\dots,q \\ &\quad h_i(x)=0, \qquad i=1,2,\dots,p \end{aligned} \tag{1}$$

where, $x=(x_1,\dots,x_n)\in X\subset R^n$ is called the decision variable, and X is the n-dimensional decision space. $f_i(x)(i=1,\dots,m)$ is the i-th objective to be minimized, $g_j(x)(j=1,2,\dots,q)$ defines the j-th inequality constraint, and $h_j(j=1,2,\dots,p)$ defines the j-th equality constraint. Furthermore, all the constraints determine the set of feasible solutions, which is denoted by Ω. To be specific, we tried to find a feasible solution $x\in\Omega$ minimizing each objective function $f_i(x)(i=1,\dots,m)$ in F.
Suppose $x,v\in\Omega$. We say ***x** dominates* $\boldsymbol{v}(x\succ v)$ if and only if $f_i(x)\ge f_i(v)$ for every $i\in\{1,2,\dots,m\}$, and $f_j(x)>f_j(v)$ for at least one index $j\in\{1,2,\dots,m\}$. A solution vector x is said to be Pareto optimal with respect to Ω if $\nexists z\in\Omega: z\succ x$. The set of Pareto optimal solutions (PS) is defined as $PS=\{x\in\Omega \mid \nexists z\in\Omega: z\succ x\}$. Finally, the Pareto optimal front (PF) is defined as all $f(x)$, where $x\in PS$. It should be mentioned that usually multi-objective optimization problems (MOPs) refer to those with two or three objectives, while those with more than three objectives are known as many-objective optimization problems (MaOPs) [29].

3. Related work

Recently, the development of EAs to solve multi-objective optimization problems has had considerable progresses [12-16, 30-31]. One significant goal in the field of MOEAs is to find a set of representative Pareto optimal solutions in a single run. Try to produce a set of Pareto optimal solutions to represent the whole PF as diverse as possible. For a desired MOP, a Pareto optimal solution is defined as a set of optimal solution for all scalar optimization problems with the aim of optimizing their aggregation function [30]. Hence, the PF approximation can be divided into a number of scalar objective optimization sub-problems and is the basis of many previous mathematical programming methods [32].

The first multi-objective GA that uses Pareto-based ranking and niching techniques explicitly together is MOGA [18]. This algorithm encourages the search toward the true Pareto front, while maintaining diversity in the population. Hence, it could be a considerable evidence to demonstrate how Pareto-based ranking and fitness sharing can be integrated in a multi-objective GA. The concept of elitism has not yet been considered in this method. In another non-elitist strategy, NSGA, the population is classified into non-dominated fronts, and then a dummy fitness value is assigned to each front (F_1, F_2,..) using a fitness sharing function so that the worst fitness value assigned to F_i is better than the best fitness value assigned to F_{i+1}.

In the MOEA literatures, many algorithms use population categorization based on the non-dominance strategy to assign a fitness value based on the non-dominance rank of the members [6]. For example, Non-dominated Sorting Genetic Algorithm II (NSGAII) [10], proposed by Deb et al. in 2002, uses the crowding distance method and the elitism strategy to obtain a uniform spread of solutions along the best-known Pareto front without using a fitness sharing parameter [8].

Zitzler et al. [33] have proposed the strength Pareto evolutionary algorithm (SPEA) [22], which assigns better fitness values to non-dominated solutions using a ranking procedure at the under-represented regions of the objective space [8]. SPEA is among the first techniques that clearly outperformed the (non-elitist) alternative approaches. It employs a fixed size external list E to store non-dominated solutions that have been investigated during the search hitherward, and a strength value is defined for each solution $y \in E$. Finally, according to these strength values, the ranking of the solution is calculated. SPEA2 [23], which is also based on the elitism strategy, differentiates between solutions with the same rank using a density estimation measure, where the density of a solution is a simple inverse of the distance of its k-th nearest neighbor in objective function space [8].

In contrast to the mentioned algorithms, which mainly rely on Pareto dominance to guide their search, MOEA/D [25] makes use of the traditional aggregation methods to transform the task of approximating the Pareto front (PF) into a number of single-objective optimization sub-problems. During the years, many metaheuristic algorithms applied the idea of decomposition for MOPs [34] [35]. In the two-phase local search (TPLS), for instance, at first, an initial solution is generated by optimizing only one single-objective, and then a search is started from this solution exploiting for non-dominated solutions based on aggregations of the objectives. The multi-objective genetic local search (MOGLS) tries to optimize all aggregations produced by the weighted sum approach or Tchebycheff approach simultaneously [36]. Various multi-objective problems with different characteristics like many objectives, discrete decision variables, and complicated Pareto set could achieve admissible results using MOEA/D [37, 38].

Moreover, some hybrid algorithms have employed the MOEA/D strategy as their basic element. For example, MOEA/D with differential evolution and particle swarm optimization has been proposed by Mashwani [39]. Ke et al. [17] have proposed a MOEAD-ACO, in which each ant (i.e. agent) is responsible for solving one sub-problem and records the best solution found so far for its sub-problem during the search. An ant combines information from its group's pheromone matrix, its own heuristic information matrix, and its current solution to construct a new solution. Li and Landa-Silva [40] have combined MOEA/D and Simulated Annealing (SA) to solve MOPs. In their proposed method, EMOSA, the weight vector of each sub-problem is adaptively modified at the lowest temperature in order to diversify the search towards the unexplored parts of the Pareto optimal front. Moreover, MOEA/D has been used to solve various kinds of problems (e.g. [37, 38]).

This paper proposes a combination of MOEA/D and Tabu Search (TS) [4] to achieve a new manner for solving multi-objective optimization problems.

This improved hybrid algorithm, namely MOEA/D-TS, uses the parallel computing capacity of MOEA/D for a comprehensive exploration of the search space along with the exploitation power of TS for discovering Pareto optimal solutions. The following sections provide more details about the proposed method.

4. Multi-objective evolutionary algorithm based on decomposition

Decomposition of MOP into N scalar optimization sub-problems and solving them altogether is a general manner of MOEA/D. By exchanging information at each generation, these sub-problems collaborate with each other [25]. There are some primary features of MOEA/D: (1) In the current population, there is the best solution found so far per each scalar optimization problem. (2) There are many sub-problems in the neighboring of each scalar optimization problem so that each two neighbor sub-problems have analogous optimal solutions. (3) In MOEA/D, information from neighboring of each sub-problem is used for its optimization. (4) Since each solution is associated with a scalar optimization problem, using scalar optimization methods in MOEA/D is very common [1].

Although decomposition of a high-dimensional MOP into a set of simpler and low-dimensional sub-problems is interesting, without a prior knowledge about the objective function, it is not clear how to decompose it [33]. Moreover, it is difficult to use such a decomposition method to solve all the multi-objective optimization problems (MOPs) because their objective functions are commonly conflicting with one another. That is to say, changing decision variables will generate incomparable solutions. Basically, a separability function means that the decision variables involved in the problem can be optimized independent from any other variable, while a non-separability function means that there exist interactions between at least two decision variables. Formal definition of separable and non-separable functions can be found in [33].

There are several approaches available to convert the problem of Pareto front approximation to some scalar optimization problems [25]. The weight sum and Tchebycheff approach are the most popular ones [41,42]. In this research work, we employed the Tchebycheff approach as the basic method, although the results of applying weight sum approach was also evaluated.

4.1. Tchebycheff and weighted sum approaches

Suppose that $\lambda = (\lambda_1, ..., \lambda_m)^T$ shows a collection of weight vectors and Z^*, $Z^* = (z_1^*, ..., z_m^*)^T$ is the ideal vector, where $Z_i^* = \max\{f_i(x) \mid x \in \Omega\}$ for i = 1, . . . , m. Using the Tchebycheff approach, decomposition of the main problem into N scalar sub-problems could be done in a way that the objective function of the *j*-th sub-problem is:

Minimize

$$g^{te}(X \mid \lambda^j, z^*) = \max_{1 \le i \le m} [\lambda_i^j \mid f_i(X) - Z_i^* \mid] \quad (2)$$

Subject to $x \in \Omega$

where, $\lambda^j = (\lambda_1^j, ..., \lambda_m^j)^T$ [25].

In the weighted sum approach, if $\sum_{i=1}^{m} \lambda_i = 1$ for weight vector **λ**, then the optimal solution to the following scalar optimization problem is a Pareto optimal point to (1):

$$Maximize\ g^{ws}(X \mid \lambda^j) = \sum_{i=1}^{m} \lambda_i^j f_i(X) \quad (3)$$

Subject to $x \in \Omega$

If PF is concave (convex in the case of minimization), this approach could work well. However, not every Pareto optimal vector can be obtained by this approach in the case of non-concave PFs. Also it should be noted that minimization of z by MOEA/D is not essential when the weight sum approach is used [25].

MOEA/D, which uses the Tchebycheff approach, keeps some information at each generation *t* including:

(1) N individual $X^1, \ldots, X^N \in \Omega$ (population), where the current solution to the *i*-th subproblem is X^i;
(2) $FV^1, \ldots, FV^N$, where $FV^i = F(X^i)$;
(3) $z = (z_1, ..., z_m)^T$ is the vector of the best value found so far for objective f_i; and
(4) An External Population (EP), which is used to store non-dominated solutions found during the search.

The desired algorithm receives MOP as input and output EP. In this process, other inputs include the number of considered sub-problems, *N*, the number of weight vectors in the neighborhood of each weight vector, *T* , a uniform distribution of N weight vectors $\lambda^1, \ldots, \lambda^N$, and the maximum number of generations, gen_{max}. In accordance with [24,43], the proposed method utilizes a binary tournament strategy as the selection operator. Two important procedures in evolutionary algorithms, recombination, and mutation operators apply on different individuals in order to replace the old population by the resulting off-spring. Also it is necessary to keep the non-dominated solutions found during the search; for this purpose, MOEA/D employs an archive namely the external population (EP). The overall pseudo-code of MOEA/D is shown in Algorithm 1 [25].

During the initialization step, for each index I, $B(i) = \{i_1, ..., i_T\}$ is computed. The Euclidean distance is used in order to compute the proximity of any two weight vectors, and also always i ∈ B(i). As j ∈ B(i), the *j*-th sub-problem is considered as a neighbor of the *i*-th sub-problem [1]. The *T* neighbors around the *i*-th sub-problem are considered in the *i*-th pass of the loop in Step 2 [40]. The available solutions to the neighbors of the *i*-th sub-problem are represented by x_k and x_j in part 1 of Step 2; hence, the resulting off-spring probably is a good candidates to be considered as an appropriate solution for the *i*-th sub-problem. When *y* violates any constraint, and/or optimizes the *i*-th $\mathbf{g^{te}}$, a heuristic is employed to repair *y* in Step 2.3. Thus the obtained solution y' is feasible with a lower function value for the neighbors of the *i*-th sub-problem. Step 2.4 considers the whole neighbors of the *i*-th sub-problem, and if y' accomplishes better than x_j due to the *j*-th sub-problem, it replaces x_j with y'. Since finding the actual ideal vector z^* is often very time-consuming, *z* is used, and Step 1 initializes and Step 2.5 updates it. At the end off Step 2.6, the external population EP utilizes the newly-generated solution y' for its update.

In order to compare the effects of the decomposition methods on the results obtained, we also considered the weight sum approach in MOEA/D. In the whole document, T-MOEA/D stands for MOEA/D using the Tchebycheff approach as a decomposition method (i.e. using $\mathbf{g^{te}}$ function (2)) [25], whereas W-MOEA/D represents MOEA/D that decomposes MOP using the weight sum approach (i.e. using $\mathbf{g^{ws}}$ function (3)).

Algorithm 1. The MOEA/D general framework

Step 1 Initialization

- Set $EP = \varnothing$ and gen = 0.
- Generate an initial population $P_0\{X^1,...,X^N\}$ and initialize $z = (z_1,...,z_m)^T$ using the lowest value for f_i found in the initial population as Z_i. Set $FV^i = F(X^i)$.
- Consider any two weight vectors, then calculate between them, and then work out the *T* closest weight vectors to each weight vector. For each I = 1,…, N, set $B(i) = \{i_1,...,i_T\}$, where $\lambda_1^i,...,\lambda_T^i$ are the T closet weight vectors to λ^i.

Step 2 Update: For I = 1, …, N do

1. **Reproduction:** In a random manner, pick out two indices **k** and **l** from B(i), and then utilizing appropriate genetic operators generate a new solution y from X^k and X^l.
2. **Mutation:** Apply Mutation operator on y to produce Y'.
3. **Update of** z**:** For each *j* = 1, …, m, if $f_j(Y') < z_j$, then set $z_j = f_j(Y')$.
4. **Update of Neighboring Solutions:** For each index $j \in B(i)$, if $g^{te}(Y' \mid \lambda^j, z) \le g^{te}(X^j \mid \lambda^j, z)$, then set $X^j = Y'$ and $FV^j = F(Y')$.
5. **EP Update:**
   - Remove the whole vectors dominated by $F(Y')$ from EP.
   - If no vectors in EP dominates $F(Y')$, add $F(Y')$ to EP.
6. **Replacement:** Use binary tournament replacement strategy

Step 3 Stopping Criteria: If gen = gen_{max}, stop and output EP. Otherwise, gen = gen + 1, go to **Step 2**.

5. Hybrid multi-objective evolutionary algorithm/D-Tabu search

The main idea behind this work was to introduce a combination of recently developed multi-objective optimization algorithms, MOEA/D and Tabu Search, for an extensive and precise probe on different multi-objective problems. The result of this hybrid method is Pareto optimal solutions with uniform distribution that cover the Pareto front as much as possible [24]. Tabu search (TS), proposed by Glover [4], is a kind of metaheuristic algorithm that aims at finding good quality solutions in an admissible time using a local search method. During the process of solution improvement, at first, the problem space was searched by TS for a potential solution x, and then other similar solutions in its neighboring N(x) were checked.

Trapping in the local optima were avoided in TS using a tabu list that remembers the history of the previous searches. Then the candidate solution with a better fitness value in the N(x) was selected as a destination for algorithm movement. The only forbidden moves are those leading to the solutions on the tabu list. The pseudo-code of Tabu Search is shown in Algorithm 2 [42].

Algorithm 2. TS general framework

Step 1: In a search space S, consider an initial solution

Set $i^* = i$ and k = 0

Step 2: k = k + 1

Make a subset of solutions in N(i,k) in a way that:

- The tabu movements are not chosen
- The aspiration criterion a(i,m) is applied
- At iteration k, N(i,k) is the neighborhood of the current solution i.

Step 3: Among N(i,k), find the best solution i', then apply i = better i'

Step 4: If $f(i) \le f(i^*)$, then apply $i^* = i$

Step 5: Update the list T and aspiration criterion.

Step 6: If a stop condition is reached, then stop. Otherwise, return to Step 2.

For each individual, MOEA/D directly defines a single-objective optimization sub-problem, and then the computational effort is distributed among these sub-problems. This process is among the major reasons why MOEA/D outperforms NSGA-II-DE on a set of continuous test instances with complicated PS shapes [30]. The proposed method in this manuscript is based on Zhang and Hui [25], and our previous work [24] was based on cooperation between SPEA2 and TS. This method, namely MOEA/D-TS, employs a comprehensive search in two levels, one global and one local, among problem spaces. The areas with high potential solutions are found during the first level search, and at the second level, a local search tries to explore the best solutions with good distribution. In what follows, the main steps of the proposed method are described:

- Applying a global search to discover multiple optimal solutions at the first step is the MOEA/D's responsibility. A Pareto front of non-dominated solutions is produced within each iteration by MOEA/D, and then it generates and sets them as the starting points for the next steps.
- In the next step, a local search should be done among the solutions obtained from MOEA/D. The Improved Diversificator Tabu Search (IDTS) [24] is a good candidate to perform a local search in order to detect new solutions [24, 43]. The covering of the Pareto front with well-distributed solutions is a significant aim in this step.

The local search using IDTS for multi-objective problems includes two steps:

1. The first step detects a less explored zone of the search space, and performs a local search in order to discover new solutions. It finds two most distant and consecutive points (SL1 or SL2) on the Pareto Front. Then it calculates the middle point Cm (the middle vector cost of SL1 and SL2) to mark the best solution belonging to the hatched dominant zone Cm.
2. During the second step, this procedure continues IDTS between SL1 and Cm (finding a new point Cm1) and between Cm and SL2 (finding a new point Cm2) to explore the best solutions in the specified dominant regions [24].

Figure 1 shows the process of IDTS for a local search in a bi-objective problem space. This method, in comparison with the simple DTS [43], reduces unexplored areas within the problem space and distributes the resulting solution on the Pareto front uniformly [24].

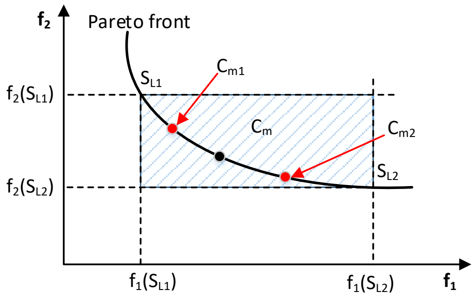

Figure 1. Search space for IDTS.

As mentioned earlier, in order to update the old population with promising solutions discovered by IDTS, the algorithm employs the binary tournament strategy. This population is used as an initial solution in the next generation. Figure 2 shows the pseudo-code of the proposed algorithm.

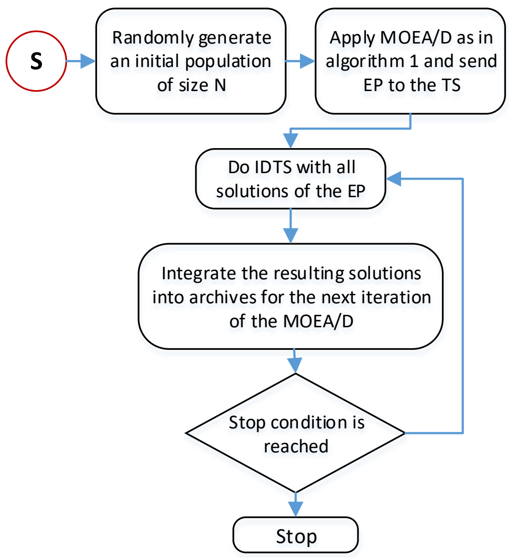

Figure 2. Flow chart of proposed algorithm (MOEA/D-TS).

6. Experimental study

In order to evaluate the capability of the proposed method and compare it with the other works in this field, namely SPEA, COMOEATS [43], and our previous algorithm SPEA2TS [24], the similar parameters as in [43] were considered and all three methods were implemented separately. The population size N was set to 100 and T in

MOEA/D-TS was considered 10% of N for all of the test instances. Table 1 illustrates the desired values for all parameters [43].

Table 1. Experimental parameters.

Parameter	Value
Initial Pop-Size (N)	100
Generation#	400
Crossover Probability (Pc)	0.9
Mutation Probability (Pm)	0.01
Tabu list size	50
Number of TS iterations	200
Tabu Life	50

The performance of the algorithm was studied on widely used bi-objective Zitzler-Deb-Thiele's test suite, namely (ZDT1 to ZDT4 and ZDT6) [44]. The test problems in the ZDT package introduce five basic functions including a distribution function f_1, a distance function g, and a shape function f_2, in which f_1 tests the ability of an MOEA to maintain diversity along the PF, function g is used for testing the ability of an MOEA to converge to PF, and function f_2 is used to define the shape of PF. These various test problems have different characteristics. Specifically, ZDT3 has a disconnected PF, which is partly convex and partly concave; ZDT4 contains a large number of local PFs and ZDT6 has a non-uniform fitness landscape. All these test instances are minimization of the objectives, and except ZDT5, which is binary-coded, the others are real-coded.

Unlike test problems in the ZDT suite, which are all bi-objective, in the DTLZ package, the test problems are scalable to have any number of objectives [33]. Each one of the nine problems in the DTLZ test suite has many unique characteristics. For instance, DTLZ1 and DTLZ3 contain a large number of local PFs in their fitness landscape, and the Pareto optimal solutions of DTLZ4 have highly non-uniform distributions. According to the similar research works [6, 25], here, we evaluated the performance of the proposed method on DTLZ1 and DTLZ2 with three objective functions. Table 2 shows the properties of these test problems.

Table 2. Experimental parameters.

Test function	Search space	Objectives	Pareto front type
ZDT1	$[0,1]^n$	$f_1(x)=x_1$ $f_2(x)=g(x)[1-\sqrt{f_1(x)/g(x)}]$ $g(x)=1+9(\sum_{i=2}^{n}(x_i-0.2)^2)/(n-1)$	convex
ZDT2	$[0, 1]^n$	$f_1(x)=x_1$ $f_2(x)=g(x)[1-(f_1(x)/g(x))^2]$ $g(x)=1+9(\sum_{i=2}^{n}(x_i-0.2)^2)/(n-1)$	Non-convex
ZDT3	$[0, 1]^n$	$f_1(x)=x_1$ $f_2(x)=g(x)[1-\sqrt{f_1(x)/g(x)}-\frac{x_1}{g(x)}\sin(10\pi x_1)]$ $g(x)=1+9(\sum_{i=2}^{n}(x_i-0.2)^2)/(n-1)$	disconnected
DTLZ1	[0, 1]n	$f_1(x)=(1+g(x))x_1x_2$ $f_2(x)=(1+g(x))x_1(1-x_2)$ $f_3(x)=(1+g(x))(1-x_1)$ $g(x)=100(n-2)+100(\sum_{i=3}^{n}\{(x_i-0.5)^2-\cos[20\pi(x_i-0.5)]\}$	Non-convex
DTLZ2	$[0, 1]^n \times [-1, 1]^{n-2}$	$f_1(x)=(1+g(x))\cos(\frac{x_1\pi}{2})\cos(\frac{x_2\pi}{2})$ $f_2(x)=(1+g(x))\cos(\frac{x_1\pi}{2})\sin(\frac{x_2\pi}{2})$ $f_3(x)=(1+g(x))\sin(\frac{x_1\pi}{2})$ $g(x)=\sum_{i=3}^{n}x_i^2$	Non-convex

There are some well-known metrics that are used to have comparison among the developed approaches [45]. These four metrics include:

Spacing: In an objective space, this metric expresses the uniformity of the solution distribution. The spacing metric calculates the distance between solutions and gives an interesting indication on the convergence of the considered method [46].

Contribution: This metric evaluates the proportion of Pareto solution brought by each one of the two (or three) foreheads F1 and F2 (and F3) [47].

Entropy: Solution entropy should be calculated to evaluate the distribution of solutions on the Pareto front. The closer the values to 1, the better the solution distribution.

***Metric S*:** This metric (that is also known as hyper-volume) measures the quality for solution sets in Pareto optimization. The Pareto front and a desired reference point are considered, and this metric calculates the hyper-volume of the multi-dimensional region between them [46].

7. Results and discussion

In this section, some simulation results and comparisons that prove the potential of MOEA/D-TS are presented. Table 3 represents a comparison between the results obtained using different algorithms (SPEA, COMOEATS, and SPEA2TS) at the level of four mentioned metrics on ZDT1 benchmark. The attained results of applying MOEA/D-TS on this convex POF show significant improvements in all the four metric values. The different values related to the spacing metric prove the capability of MOEA/D-TS to generate more uniform Pareto optimal solutions than the three other methods. In this way, more discovered zones can be covered with a good uniform distribution. Moreover, the outcomes of table 3 depict that using the Tchebycheff approach as a decomposition method in MOEA/D-TS (i.e. T-MOEA/D) in most cases (except metric S) leads to better results in comparison with exploiting the weighted sum approach for decomposition (i.e. W-MOEA/D).

The statistics of the values obtained by each algorithm in ZDT2 are represented in table 4. Here, we are faced with a non-convex POF. It is obvious from the results that MOEA/D-TS outperforms other methods due to the three metrics except entropy. Although the new method did not have enough power to overcome SPEA2TS, it achieved better results in comparison with the other algorithms.

Tables 5-7 depict the various results obtained by each algorithm in ZDT3 (with a discontinuous PF), ZDT4, and ZDT6 test functions, respectively. For these problems, our proposed algorithm shows its ability to achieve interesting results at the level of all four metrics.

Tables 8 and 9 compare the results obtained by different algorithms in the three-objective problems DTLZ1 and DTLZ2. It is quite clear from these results that MOEA/D-TS performs much better than the other algorithms at the level of four criteria, and using the Tchebycheff decomposition approach compared with the weighted sum approach mainly achieves more satisfactory outcomes in these three-objective instances.

According to the attained results presented in table 3-9, the MOEA/D-TS is able to handle various multi-objective problems having two and three objective and convex, non-convex, and discontinuous POFs. In addition, it is obvious that T-MOEA/D achieves better results than W-MOEA/D at most of the metrics except *metric S* at ZDT1, ZDT2, and DTLZ1, and also metric *Spacing* at ZDT3. These results may be due to the one weakness of the Tchebycheff approach, in which the aggregation function is not smooth for continuous MOPs (i.e. ZDT1, ZDT2, and DTLZ1) [25]. In this case, calculation of the hyper-volume of the multi-dimensional region between Pareto front and desired reference point (i.e. metric S) is complicated.

In order to visually compare the performance of the four algorithms, the solutions obtained by them in these test problems are shown in figures 4 and 5. These figures show the distributions of the solutions on Pareto fronts in 30 independent runs. The comparisons mainly focus on two aspects: 1) the coverage of the solutions obtained to the true PF; and 2) the diversity of the solutions obtained. Obviously, both SPEA and COMOEATS cannot locate the global PF in any instance, and the results attained by SPEA2TS are not completely satisfactory. In contrast, MOEA/D-TS can approximate the PFs of these instances quite well. These solutions obtained by MOEA/D-TS have covered most of less discovered zones, with a uniform distribution that confirm our claim about the effect of IDTS to cover most of the unexplored zones of the Pareto front. These results indicate that the diversity and coverage of solutions obtained by the algorithm MOEA/D-TS are better than those obtained by SPEA, COMOEATS, and even SPEA2TS on these test problems.

Table 3. Metrics values for ZDT1.

Algorithm / Metric	SPEA	COMOEATS	SPEA2TS	MOEA/D-TS W-MOEA/D	MOEA/D-TS T-MOEA/D
Spacing	0.0203861	0.0256606	0.0234362	0.0206327	0.018847
Contribution	0.492958	0.507042	0.556231	0.56035	0.591044
Entropy	0.360803	0.367399	0.505162	0.58183	0.61354
Metric S	0.5524335	0.55787	0.56085	0.55572	0.55924

Table 4. Metrics values for ZDT2.

Algorithm / Metric	SPEA	COMOEATS	SPEA2TS	MOEA/D-TS W-MOEA/D	MOEA/D-TS T-MOEA/D
Spacing	0.0203861	0.0276606	0.018173	0.014208	0.011386
Contribution	0.492958	0.507092	0.566471	0.58363	0.62043
Entropy	0.360803	0.371775	0.517232	0.42522	0.48803
Metric S	0.5524335	0.55689	0.542853	0.40917	0.47261

Table 5. Metrics values for ZDT3.

Algorithm / Metric	SPEA	COMOEATS	SPEA2TS	MOEA/D-TS W-MOEA/D	MOEA/D-TS T-MOEA/D
Spacing	0.0206785	0.0116797	0.0074512	0.002258	0.009341
Contribution	0.496454	0.507042	0.627452	0.84512	0.95386
Entropy	0.365199	0.373243	0.387123	0.402102	0.449638
Metric S	0.750998	0.741164	0.725361	0.677366	0.651146

Table 6. Metrics values for ZDT4.

Algorithm / Metric	SPEA	COMOEATS	SPEA2TS	MOEA/D-TS W-MOEA/D	MOEA/D-TS T-MOEA/D
Spacing	0.0224316	0.0263217	0.023267	0.018803	0.012651
Contribution	0.462758	0.507092	0.523716	0.58363	0.60342
Entropy	0.362131	0.40721	0.503571	0.52058	0.54507
Metric S	0.529386	0.52309	0.5201453	0.50152	0.48713

Table 7. Metrics values for ZDT6.

Algorithm / Metric	SPEA	COMOEATS	SPEA2TS	MOEA/D-TS W-MOEA/D	MOEA/D-TS T-MOEA/D
Spacing	0.0220126	0.0270117	0.023267	0.018803	0.012651
Contribution	0.483217	0.490278	0.523716	0.58363	0.60342
Entropy	0.362891	0.41834	0.503571	0.52058	0.54507
Metric S	0.741834	0.725037	0.720341	0.693617	0.67571

Table 8. Metrics values for DTLZ1.

Algorithm / Metric	SPEA	COMOEATS	SPEA2TS	MOEA/D-TS W-MOEA/D	MOEA/D-TS T-MOEA/D
Spacing	0.27318	0.24533	0.13867	0.10391	0.08651
Contribution	0.52174	0.57723	0.60265	0.63241	0.65829
Entropy	0.394321	0.43145	0.577141	0.70148	0.72328
Metric S	0.83307	0.82198	0.80721	0.71057	0.75251

Table 9. Metrics values for DTLZ2.

Algorithm / Metric	SPEA	COMOEATS	SPEA2TS	MOEA/D-TS W-MOEA/D	MOEA/D-TS T-MOEA/D
Spacing	0.15257	0.12324	0.10015	0.08391	0.05651
Contribution	0.54812	0.57034	0.59107	0.67763	0.69135
Entropy	0.362031	0.39557	0.421972	0.53261	0.58204
Metric S	0.78307	0.731078	0.70681	0.67152	0.63142

8. Summary and conclusion

In this paper, we proposed a hybrid method derived from Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) and Tabu Search (TS) for solving various multi-objective optimization problems. This algorithm, namely MOEA/D-TS, at its first level uses the capabilities of MOEA/D for exploration of the problem space by decomposing MOP into single-objective optimization sub-problems. An Improved Diversificator Tabu Search (IDTS) is utilized to perform local search among the problem space at the second level. The main goal of IDTS is achievement to a Pareto front with minimum unknown parts and well-distributed solutions. The experimental results considering seven benchmarks with different numbers of objective functions and various POF demonstrate that MOEA/D-TS has more functionality than SPEA, COMOEATS, and SPEA2TS to discover solution sets with a better quality. Also the results obtained indicate that using the Tchebycheff approach as a decomposition method in these kinds of problems will lead to better values than using the weighted sum decomposition approach. The main reason is that the weighted sum approach is compatible with concave (convex in the case of minimization) PFs, and not every Pareto optimal vector can be obtained by this approach in the case of non-concave PFs.

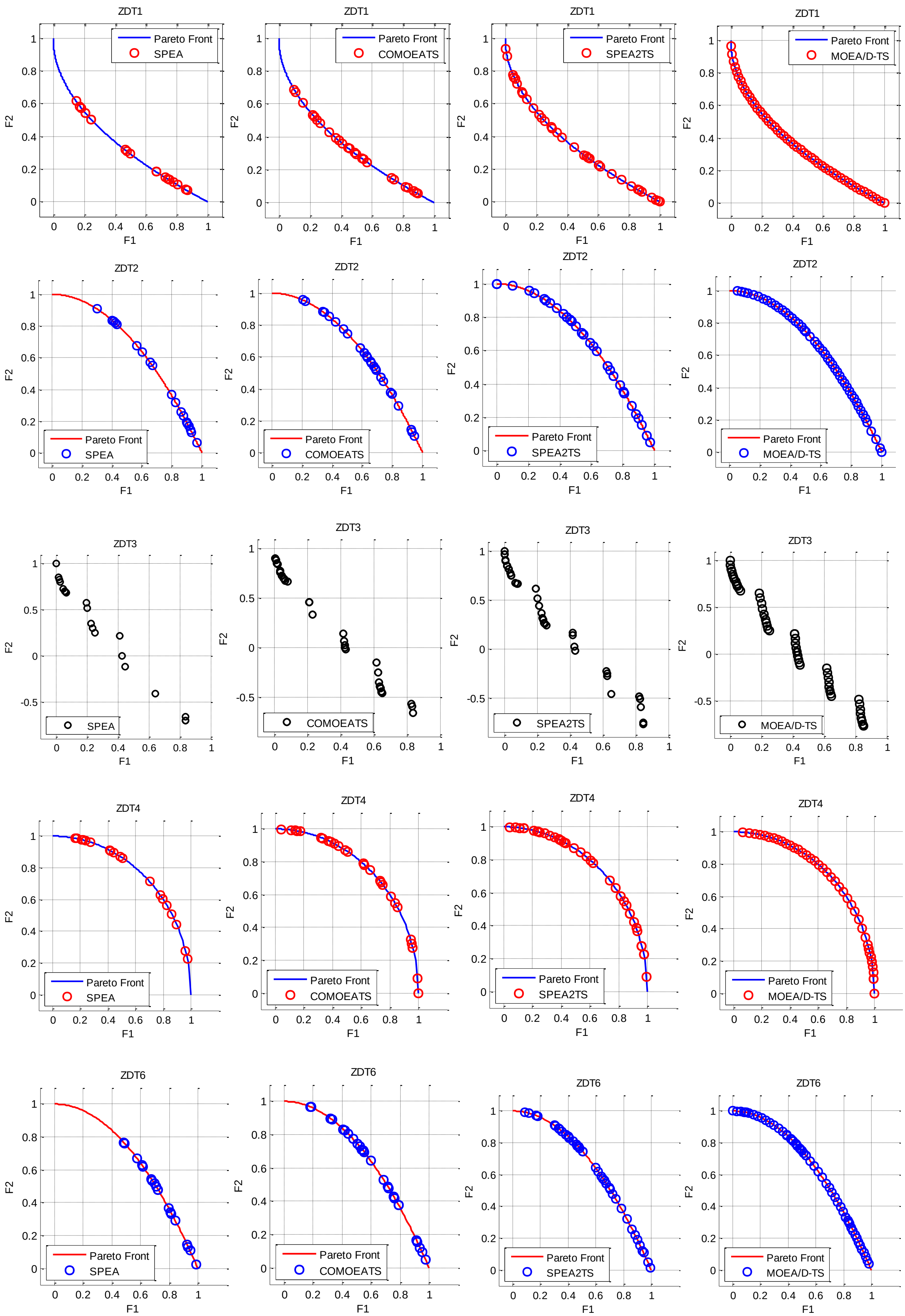

Figure 3. Solutions obtained by SPEA, COMOEATS, SPEA2TS, and MOEA/D-TS on ZDT test functions.

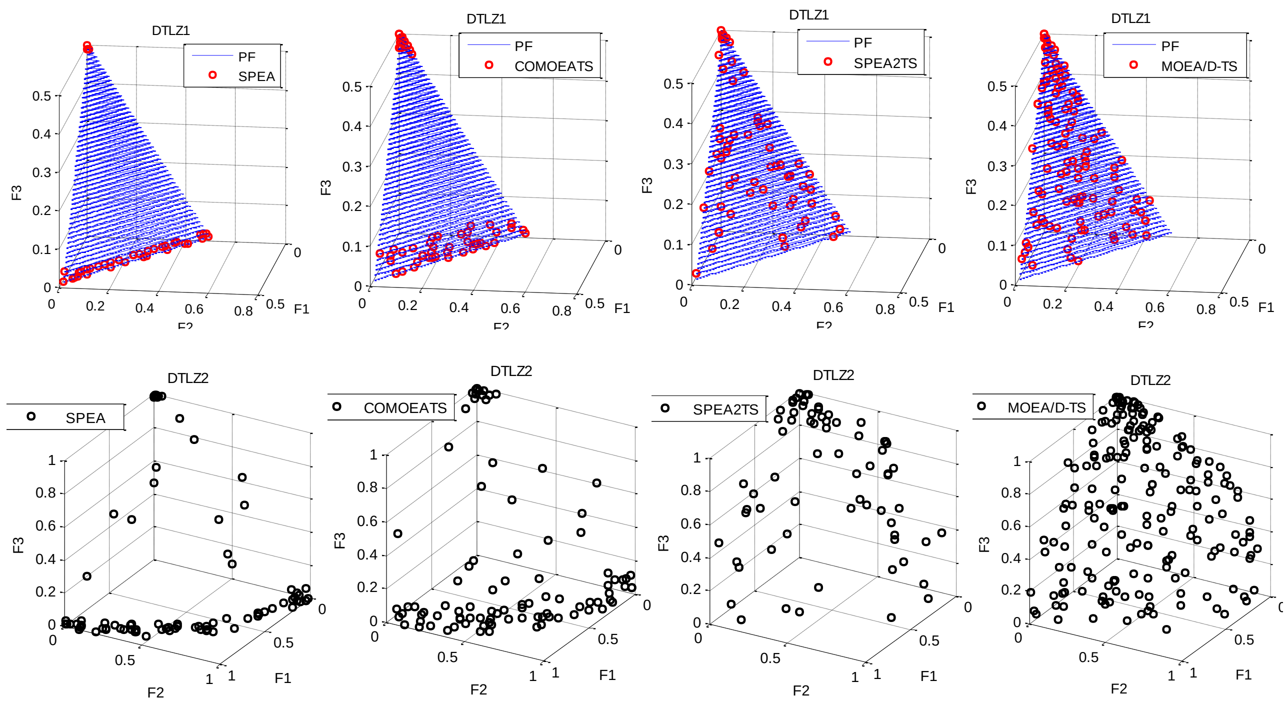

Figure 4. Solutions obtained by SPEA, COMOEATS, SPEA2TS, and MOEA/D-TS on DTLZ test function.

References

[1] Marler, R. T. & Arora, J. S. (2004). Survay of multi-objective optimization methods for engineering. Structural and Multidisciplinary Optimization, vol. 26, no. 6, pp. 369-395.

[2] Suman, B. (2004). Study of simulated annealing based algorithms for multiobjective optimization of a constrained problem. Computers and Chemical Engineering, vol. 28, pp. 1849-1871.

[3] Fonseca, C. M. & Fleming, P. J. (1995). An overview of evolutionary algorithms in multiobjective optimization, Evolutionary Computation., vol. 3, no. 1, pp. 1-16.

[4] Valenzuela-Rend´on, M. & Uresti-Charre, E. (1997). A nongenerational genetic algorithm for multiobjective optimization, 7th Int. Conf. GeneticAlgorithms, T. B¨ack, Ed. San Francisco, CA: Morgan Kaufmann, pp. 658-665.

[5] Srinivas N. & Deb K., (1994). Multiobjective optimization using nondominated sorting in genetic algorithms, Evolutionary Computation vol. 2, no. 3, pp. 221-248.

[6] Dai, C. & Wang, Y. (2014). A New Multiobjective Evolutionary Algorithm Based on Decomposition of the Objective Space for Multiobjective Optimization, Journal of Applied Mathematics, vol. 2014, Article ID 906147, 9 pages, 2014. doi:10.1155/2014/906147.

[7] Gong, M., et al. (2008). Multiobjective immune algorithm with nondominated neighbor-based selection, Evolutionary Computation, vol. 16, no. 2, pp. 225-255.

[8] Konak, A., Coit, D. W. & Smith, A. E. (2006) Multi-objective optimization using genetic algorithms: a tutorial, Reliability Engineering and System Safety, vol. 91, no. 9, pp. 992-1007.

[9] Chang, P.-C. & Chen, S.-H. (2009). The development of a sub-population genetic algorithm II (SPGA II) for multi-objective combinatorial problems, Appl. Soft Comput., vol. 9, no. 1, pp. 173-181.

[10] Deb, K., et al. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactionson Evolutionary Computation, vol. 6, no. 2, pp. 182-197.

[11] Liu, R., et al. (2013). A preference multi-objective optimization based on adaptive rank clone and differential evolution, Natural Computing, vol. 12, no. 1, pp. 109-132.

[12] Matjaz, D., Roman, T. & Bogdan, F. (2013). Asynchronous masterslave parallelization of differential evolution for multi-objective optimization, Evolutionary Computation, vol. 21, no. 2, pp. 261-291.

[13] Goh, C. K., et al. (2010). A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design, European Journal of Operational Research, vol. 202, no. 1, pp. 42-54.

[14] Gómez, R., Coello Coello, C. A. & Alba Torres E. (2016). A Multi-Objective Evolutionary Algorithm based on Parallel Coordinates, In Proceedings of the Genetic and Evolutionary Computation Conference 2016 (GECCO '16), Tobias Friedrich (Ed.). ACM, New York, NY, USA, pp. 565-572.

[15] Guo, W., et al. (2016). Hyper multi-objective evolutionary algorithm for multi-objective optimization problems, soft computing, pp. 1-9, doi:10.1007/s00500-016-2163-5.

[16] Guo, W., et al. (2016). A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables, IEEE Transactions on Evolutionary Computation, vol. 20, no. 2, pp. 275-298.

[17] Paquete, L. & Stutzle, T. (2013). MOEA/D-ACO: A Multiobjective Evolutionary Algorithm Using Decomposition and Ant Colony, IEEE Transactions On Cybernetics, vol. 43, no. 6, pp. 1845-1859.

[18] Fonseca, C. M. & Fleming, P. J. (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, In S. Forrest (Ed.), Proceedings of the Fifth International Conference on Genetic Algorithms, San Mateo, California, pp. 416-423.

[19] Horn, J., Nafpliotis, N. & Goldberg, D. E. (1994). A niched pareto genetic algorithm for multiobjective optimization, First IEEE Conferenceon Evolutionary Computation, IEEE World Congress on Computational Computation, vol. 1, Piscataway, NJ, pp. 82-87.

[20] Naujoks, B., Beume, N. & Emmerich, M. (2005). Multi-objective Optimisation using S-metric Selection: Application to three-dimensional Solution Spaces, 2005 IEEE Congress on Evolutionary Computation, pp. 1282-1289.

[21] Zitzler, E., Deb, K. & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results, Evolutionary Computation, vol. 8, no. 173, pp. 173-195.

[22] Zitzler E. & Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms: A comparative case study, In A. E. Eiben, T. Back, M. Schoenauer, H. P. Schwefel (Eds.), Parallel problem solving from Nature, vol. V, p. 292-301, Berlin, Germany: Springer.

[23] Zitzler E., Laumanns, M. & Thiele, L. (2002). SPEA2: Improving the Strength Pareto Evolutionary Algorithm, from Evolutionary Methods for Design, Optimisation and Control, CIMNE, Barcelona, Spain.

[24] Karimi, F. & Lotfi, Sh. (2014). Solving multi-objective problems using SPEA2 and Tabu search, Intelligent Systems (ICIS), Iranian Conference on, IEEE.

[25] Zhang, Q. & Hui, L. (2007). A Multi-objective Evolutionary Algorithm Based on Decomposition, Evolutionary Computation, IEEE Transactions on, vol. 11, no. 6, pp. 712-731.

[26] Ishibuchi, H. & Murata, T. (1998). Multi-objective genetic local search algorithm and its application to flowshop scheduling, IEEE Transactions on Systems, Man and Cybernetics, vol. 28, no. 3, pp. 392-403.

[27] Leung, Y. W. & Wang, Y. (2000). Multiobjective programming using uniform design and genetic algorithm. IEEE Transactions on Systems, Man, and Cybernetics, Part C, vol. 30, no. 3, pp. 293-304.

[28] Jin, Y., Okabe, T. & Sendhoff, B. (2001). Adapting weighted aggregation for multiobjective evolutionary strategies, in EMO '01: Evolutionary Multicriterion OPtimization, Springer LNCS, pp. 96-110.

[29] Cheng, R., et al. (2016). Test Prblems for Large-Scale Multiobjective and Many-Objective Optimization, IEEE Transactions on Cybernetics, Epub, vol. PP, Issue 99, pp. 1-14.

[30] Zhang, Q. & Li, H. (2009). Comparison Between NSGA-II and MOEA/D on a Set of Multiobjective Optimization Problems with Complicated Pareto Sets, Evolutionary Computation, IEEE Transactions on, vol. 13, no. 2, pp. 284-302.

[31] Motameni, H. (2016). PSO for multi-objective problems: Criteria for leader selection and uniformity distribution, Journal of AI and Data Mining, Vol. 4, No. 1, pp. 67-76.

[32] Miettinen, K. (1999). Nonlinear Multiobjective Optimization. Kluwer Academic Publishers.

[33] Ma, X., et al. (2016). Multiobjective Evolutionary Algorithm Based on Decision Variable Analyses for MultiobjectiveOptimization Problems With Large-Scale Variables, IEEE Transactions On Evolutionary computation, vol. 20, no. 2, pp. 275-298.

[34] Ulungu, E., et al. (1999). Mosa method: A tool for solving multiobjective combinatorial optimization problem, Journal of Multi-Criteria Decision Analysis, vol. 8, no. 4, pp. 221-236.

[35] Paquete, L. & Stutzle, T. (2003). A two-phase local search for the biobjective traveling salesman problem. in EMO, pp. 479-493.

[36] Jaszkiewicz, A. (2002). On the performance of multiple-objective genetic local search on the 0:1 knapsack problem-a comparative experiment, IEEE Trans. Evolutionary Computation, vol. 6, no. 4, pp. 402-412.

[37] Shim, V. A., Tan, K. C. & Tan, K. K. (2012). A hybrid estimation of distribution algorithm for solving the multi-objective multiple traveling salesman problem, in Proceedings of the IEEE Congress on Evolutionary Computation (CEC'12), pp. 1-8.

[38] Chan, Y.-H., Chiang, T.-C. & Fu, L.-C. (2010). A two-phase evolutionary algorithm for multiobjective mining of classification rules, in Proceedings of the IEEE Congress on Evolutionary Computation (CEC'10).

[39] Mashwani, W. K. (2011). MOEA/D with DE and PSO: MOEA/DDE+PSO, in Proceedings of the 31st SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, pp. 217-221.

[40] Özdemir, S., Attea, B. A. & Khalil, Ö. A. (2012). Multi-Objective Evolutionary Algorithm Based on Decomposition for Energy Efficient Coverage in Wireless Sensor Networks, springer, pp. 1-6.

[41] Miettinen, K. (1999). Nonlinear Multiobjective Optimization. Kluwer Academic Publishers.

[42] Jaszkiewicz, A. (2002). On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - a comparative experiment, IEEE Trans. Evolutionary Computation, vol. 6, no. 4, pp. 402-412.

[43] Hajlaoui, R., Gzara, M. & Abdelaziz, D. (2011). Hybrid Model for Solving Multi-Objective Problems Using Evolutionary Algorithm and Tabu Search, World of Computer Science and Information Technology Journal, vol. 1, no. 1, pp. 5-9.

[44] Okabe, T., et al. (2004). On test functions for evolutionary multi-objective optimization, in Parallel Problem Solving from Nature. Springer, pp. 792-80.

[45] ZeinEldin, R. A. (2014). A Hybrid SS-SA Approach for Solving Multi-Objective Optimization Problems, European Journal of Scientific Research, vol. 121, no. 3, pp. 310-320.

[46] Naujoks, B., Beume, N. & Emmerich, M. (2005). Multi-objective Optimisation using S-metric Selection: Application to three-dimensional Solution Spaces, IEEE.

[47] Haussler, D. & Opper, M. (1997). Mutual information, metric entropy and cumulative relative entropy risk, Ann. Statist., vol. 25, no. 6, pp. 2451-2492.

نشریه هوش مصنوعی و داده کاوی

ترکیب الگوریتم تکاملی چند-هدفه بر مبنای تجزیه و جستجوی ممنوعه برای حل مسائل چند-هدفه

شهریار لطفی و فاطمه کریمی*

گروه علوم کامپیوتر، دانشگاه تبریز، تبریز، ایران.

ارسال ۲۰۱۶/۰۶/۰۶؛ بازنگری ۲۰۱۶/۱۱/۲۷؛ پذیرش ۲۰۱۷/۰۱/۱۸

چکیده:

مسائل گوناگون بهینه‌سازی با اهداف متضاد، در بسیاری از کاربردهای دنیای واقعی کاملاً متعارف می‌باشند. در این پژوهش ما از یک روش جدید به نام الگوریتم تکاملی چند-هدفه بر مبنای تجزیه (MOEA/D) در کنار قابلیت‌های جستجوی ممنوعه (TS)، جهت دستیابی به شیوه‌ای نوین برای حل مسائل بهینه‌سازی چند-هدفه با دو یا چند هدف متضاد بهره‌گیری نموده‌ایم. این الگوریتم ترکیبی بهبودیافته، به نام MOEA/D-TS، از قابلیت پردازش موازی MOEA/D در کنار قدرت TS در جستجوی محلی، برای کشف راه‌حل‌های بهینه‌ی پرتو استفاده می‌نماید. هدف ما، بهره‌برداری کردن از مزایای الگوریتم‌های تکاملی و جستجوی ممنوعه جهت دستیابی به یک روش یکپارچه برای پوشش کامل جبهه‌ی پرتو با راه‌حل‌های توزیع‌شده به شکل نرمال می‌باشد. به منظور ارزیابی قابلیت‌های روش پیشنهادی، کارآیی آن بر اساس معیارهای مختلف بر روی مجموعه‌های آزمون معروف ZDT و DTLZ با توابع هدف تفکیک‌پذیر، با SPEA، COMOEATS و SPEA2TS مقایسه شده‌است. بر اساس نتایج به دست آمده، روش پیشنهادی قادر است به طور قابل توجهی بر الگوریتم‌های پیشین غلبه کرده و نتایج کاملاً رضایت‌بخشی را ارائه نماید.

کلمات کلیدی: مسائل چند-هدفه، الگوریتم‌های تکاملی، روش ترکیبی، الگوریتم تکاملی چند-هدفه بر مبنای تجزیه، جستجوی ممنوعه.