Stable Matching-Based Selection in Evolutionary Multiobjective Optimization

Ke Li, *Student Member, IEEE,* Qingfu Zhang, *Senior Member, IEEE,* Sam Kwong, *Fellow, IEEE,* Miqing Li, and Ran Wang, *Student Member, IEEE*

***Abstract*—Multiobjective evolutionary algorithm based on decomposition (MOEA/D) decomposes a multiobjective optimization problem into a set of scalar optimization subproblems and optimizes them in a collaborative manner. Subproblems and solutions are two sets of agents that naturally exist in MOEA/D. The selection of promising solutions for subproblems can be regarded as a matching between subproblems and solutions. Stable matching, proposed in economics, can effectively resolve conflicts of interests among selfish agents in the market. In this paper, we advocate the use of a simple and effective stable matching (STM) model to coordinate the selection process in MOEA/D. In this model, subproblem agents can express their preferences over the solution agents, and vice versa. The stable outcome produced by the STM model matches each subproblem with one single solution, and it tradeoffs convergence and diversity of the evolutionary search. Comprehensive experiments have shown the effectiveness and competitiveness of our MOEA/D algorithm with the STM model. We have also demonstrated that user-preference information can be readily used in our proposed algorithm to find a region that decision makers are interested in.**

***Index Terms*—Decomposition, deferred acceptance procedure, multiobjective evolutionary algorithm based on decomposition (MOEA/D), multiobjective optimization, preference incorporation, stable matching.**

I. Introduction

MULTIOBJECTIVE optimization problems (MOPs), which naturally arise in many disciplines, such as engineering [1], economics [2], and logistics [3], involve more than one objective function to optimize. Since these objectives often conflict with one another, no single solution can optimize all the objectives at the same time. Pareto optimal solutions, which are their best tradeoff candidates, can be of great interest to decision makers. A MOP can have a set of different Pareto optimal solutions. Over the past decades, much effort has been made to develop multiobjective evolutionary algorithms (MOEAs) for approximating the set of Pareto optimal solutions.

Selection is a major issue in designing MOEAs. Both convergence and diversity are important for MOEAs to obtain a good approximation to the set of Pareto optimal solutions. It is desirable that the selection operator can balance them. Based on the different selection strategies, most current MOEAs can be classified into three categories: 1) dominance-based MOEAs (e.g., [4]–[6]); 2) indicator-based MOEAs (e.g., [7]–[9]); and 3) decomposition-based MOEAs (e.g., [10]–[12]). Multiobjective evolutionary algorithm based on decomposition (MOEA/D) is a popular decomposition-based MOEA. It decomposes a MOP into a set of single objective optimization subproblems and optimizes them in a collaborative manner. A number of MOEA/D variants have been suggested and studied (e.g., [13]–[15]). In MOEA/D, the selection of solutions is decided by their aggregation function values, and the population diversity is achieved by the wide spread of subproblems. If we think of the subproblems and solutions as two different agent sets, the selection in MOEA/D can be regarded as a two-sided matching problem. Therefore, matching theory and techniques [16] can be applied for designing the selection operators of MOEA/D in a systematic and rational way. This paper presents a first attempt along this direction.

Matching, first proposed and studied in a Nobel Prize winning paper [17], has found many applications in various fields (e.g., [18]–[20]). The stable marriage problem (SMP), introduced in [17], is about how to match two sets of agents, i.e., men and women. Each man ranks the women in the order of his preferences, and vice versa. It is certainly undesirable if a matching contains a man and woman who are not matched together but prefer each other to their assigned spouses. Such a pair make the matching unstable since they have a clear incentive to break up from their current marriages and marry each other instead. Therefore, a stable matching should not have such a pair. Fig. 1 presents a simple marriage market for example, where three men, m_1, m_2, m_3, and three women, w_1, w_2, w_3, have listed their preference orderings over the opposite sex. For example, for m_1, $[w_1, w_2, w_3]$ means that m_1 ranks w_1 first, w_2 second, and w_3 last. Fig. 1

Manuscript received May 22, 2013; revised October 6, 2013; accepted November 19, 2013. Date of publication December 5, 2013; date of current version November 26, 2014. This work was supported by the National Natural Science Foundation of China under Grant 61272289. This paper was recommended by Associate Editor H. Ishibuchi.

K. Li, S. Kwong, and R. Wang are with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong (e-mail: keli.genius@gmail.com; cssamk@cityu.edu.hk; ranwang3-c@my.cityu.edu.hk).

Q. Zhang is with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong (e-mail: qingfu.zhang@cityu.edu.hk). He is on leave from the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: qzhang@essex.ac.uk).

M. Li is with the Department of Information Systems and Computing, Brunel University, Uxbridge, Middlesex UB8 3PH, U.K. (e-mail: miqing.li@brunel.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2013.2293776

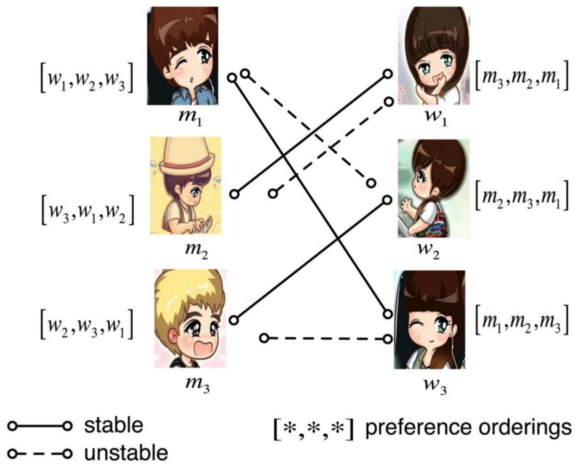


Fig. 1. Simple example of SMP, where the matching $\{(m_1, w_3), (m_2, w_1), (m_3, w_2)\}$ is stable but the matching $\{(m_1, w_2), (m_2, w_1), (m_3, w_3)\}$ is unstable.

shows two matching results. One is connected by solid lines, e.g., (m_1, w_3), the other one is connected by dashed lines, e.g., (m_1, w_2). Obviously, the matching $\{(m_1, w_3), (m_2, w_1), (m_3, w_2)\}$ is stable, whereas the matching $\{(m_1, w_2), (m_2, w_1), (m_3, w_3)\}$ is unstable. This is because m_2 prefers w_3 to w_1, and w_3 prefers m_2 to her spouse m_3.

This paper proposes using a simple and effective stable matching (STM) model to implement the selection operator in MOEA/D. Each subproblem agent in MOEA/D, by using its aggregation function, ranks all solutions in the solution pool. It prefers the solutions with better aggregation function values. Therefore, the preferences of the subproblems encourage the convergence. On the other hand, each solution agent ranks all subproblems according to its distance to the direction vector of these subproblems. The preferences of the solutions can promote the diversity. Then, a matching algorithm can be used to assign a solution to each subproblem. This assignment can balance the preferences of subproblems and solutions, and thus the convergence and diversity of the evolutionary search. We also show that our proposed algorithm can easily accommodate the preferences of a decision maker.

In the remainder of this paper, we first provide some background knowledge in Section II. Then, the technical details of our selection operator, based on the STM model, are described in Section III, and its incorporation with MOEA/D is presented in Section IV. The experimental settings are provided in Section V, and comprehensive experiments, including the user-preference incorporation mechanism, are conducted and analyzed in Section VI. Finally, conclusions of this paper and some future research issues are given in Section VII.

II. Background

This section first gives some basic definitions in multiobjective optimization. Then, we briefly introduce three widely used decomposition approaches for MOPs.

A. Basic Definitions

A MOP can be mathematically defined as follows:

$$\begin{aligned} &\text{minimize} \quad \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\ &\text{subject to} \quad \mathbf{x} \in \Omega \end{aligned} \tag{1}$$

where $\Omega = \Pi_{i=1}^{n}[a_i, b_i] \subseteq \mathbb{R}^n$ is the decision (variable) space, and $\mathbf{x} = (x_1, \cdots, x_n)^T \in \Omega$ is a candidate solution. $\mathbf{F} : \Omega \rightarrow \mathbb{R}^m$ constitutes m real-valued objective functions and $\mathbb{R}^m$ is called the objective space. The attainable objective set is defined as the set $\Theta = \{\mathbf{F}(\mathbf{x})|\mathbf{x} \in \Omega\}$.

Definition 1: $\mathbf{x}^1$ is said to Pareto dominate $\mathbf{x}^2$, denoted by $\mathbf{x}^1 \preceq \mathbf{x}^2$, if $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$ for every $i \in \{1, \cdots, m\}$ and $f_j(\mathbf{x}^1) < f_j(\mathbf{x}^2)$ for at least one index $j \in \{1, \cdots, m\}$.

Definition 2: A solution $\mathbf{x}^* \in \Omega$ is said to be Pareto optimal if there is no other solution $\mathbf{x} \in \Omega$ such that $\mathbf{x} \preceq \mathbf{x}^*$.

Definition 3: The set of all the Pareto-optimal solutions is called the Pareto-optimal set (PS). The set of all the Pareto-optimal vectors, $PF = \{\mathbf{F}(\mathbf{x}) \in \mathbb{R}^m|\mathbf{x} \in PS\}$, is called the Pareto front (PF).

Definition 4: The *ideal objective vector* $\mathbf{z}^*$ is a vector $\mathbf{z}^* = (z_1^*, \cdots, z_m^*)^T$, where $z_i^* = \min_{\mathbf{x}\in\Omega} f_i(\mathbf{x})$, $i \in \{1, \cdots, m\}$.

Definition 5: The *nadir objective vector* $\mathbf{z}^{nad}$ is a vector $\mathbf{z}^{nad} = (z_1^{nad}, \cdots, z_m^{nad})^T$, where $z_i^{nad} = \max_{\mathbf{x}\in PS} f_i(\mathbf{x})$, $i \in \{1, \cdots, m\}$.

Definition 6: Given a finite number of points $x_1, \cdots, x_n$, a convex combination of these points is a point of the form $\sum_{i=1}^{n} w_i x_i$, where $w_i \geq 0$ for all $i \in \{1, \cdots, n\}$ and $\sum_{i=1}^{n} w_i = 1$.

B. Decomposition Approaches

Several approaches can be used to decompose the approximation of the PF into a number of scalar optimization subproblems [10], [21]. In the following, we will briefly introduce the three most commonly used decomposition approaches and their search directions in the objective space.

1) *Weighted Sum (WS) Approach:* This approach considers a convex combination of all the individual objectives. Let $\mathbf{w} = (w_1, \ldots, w_m)^T$ be a weight vector where $w_i \geq 0$ for all $i \in \{1, \ldots, m\}$ and $\sum_{i=1}^{m} w_i = 1$. Then, one optimal solution to the following single objective optimization problem:

$$\begin{aligned} &\text{minimize} \quad g^{ws}(\mathbf{x}|\mathbf{w}) = \textstyle\sum_{i=1}^{m} w_i f_i(\mathbf{x}) \\ &\text{subject to} \quad \mathbf{x} \in \Omega \end{aligned} \tag{2}$$

is a Pareto-optimal solution to (1). By using different weight vectors in (2), one can obtain a set of different Pareto-optimal solutions to approximate the PF when it is convex. This approach, however, may not be able to find all the Pareto-optimal solutions in the case of nonconvex PFs (see [21, p.79]). The search direction of the WS approach is $\mathbf{w} = (w_1, \ldots, w_m)^T$ as shown in Fig. 2(a).

2) *Tchebycheff (TCH) Approach:* In this approach, the single objective optimization problem is[1]

$$\begin{aligned} &\text{minimize} \quad g^{tch}(\mathbf{x}|\mathbf{w}, \mathbf{z}^*) = \max_{1\leq i\leq m} \{|f_i(\mathbf{x}) - z_i^*|/w_i\} \\ &\text{subject to} \quad \mathbf{x} \in \Omega. \end{aligned} \tag{3}$$

For convenience, we allow $w_i = 0$ in setting $\mathbf{w}$, but replace $w_i = 0$ by $w_i = 10^{-6}$ in (3). The direction vector for this subproblem is $\mathbf{w} = (w_1, \ldots, w_m)^T$.

[1]The weight vector setting of the TCH approach in this paper is different from that in [10]. As reported in some recent studies, e.g., [22] and [23], the setting used in this paper can produce more uniformly distributed solutions in the objective space.

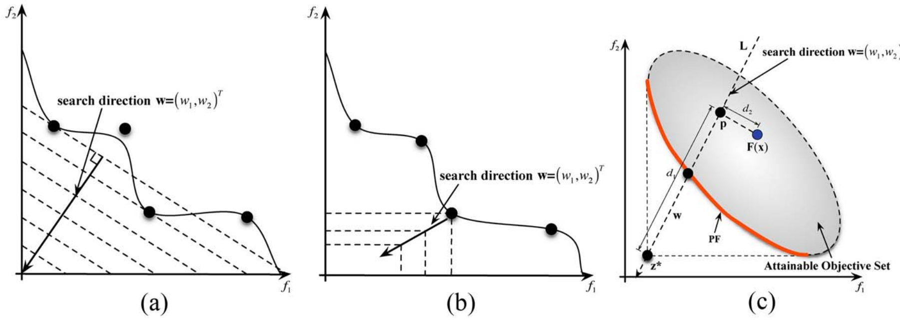


Fig. 2. Illustrations on three decomposition approaches. (a) WS. (b) TCH. (c) PBI.

3) *Penalty-Based Boundary Intersection (PBI) Approach:* This approach is a variant of the normal-boundary intersection method [24], whose equality constraint is handled by a penalty function. More formally, a single objective optimization problem in this approach is defined as

$$\begin{array}{ll}\text{minimize} & g^{pbi}(\mathbf{x}|\mathbf{w}, \mathbf{z}^*) = d_1 + \theta d_2 \\ \text{subject to} & \mathbf{x} \in \Omega\end{array} \tag{4}$$

where $\theta > 0$ is a user-defined penalty parameter, $d_1 = \|(\mathbf{F}(\mathbf{x}) - \mathbf{z}^*)^T \mathbf{w}\| / \|\mathbf{w}\|$ and $d_2 = \|\mathbf{F}(\mathbf{x}) - (\mathbf{z}^* + d_1 \mathbf{w})\|$. As shown in Fig. 2(c), $\mathbf{L}$ is a line passing through $\mathbf{z}^*$ with direction $\mathbf{w}$, and $\mathbf{p}$ is the projection of $\mathbf{F}(\mathbf{x})$ on $\mathbf{L}$. d_1 is the distance between $\mathbf{z}^*$ and $\mathbf{p}$ and d_2 is the perpendicular distance between $\mathbf{F}(\mathbf{x})$ and $\mathbf{L}$. The goal of this approach is to push $\mathbf{F}(\mathbf{x})$ as low as possible so that it can reach the boundary of the attainable objective set. The search direction of this subproblem is $\mathbf{w} = (w_1, \ldots, w_m)^T$.

III. Selection Operator

This section gives the technical details of the selection operator based on the stable matching (STM) model. We also discuss the selection operation in MOEA/D.

A. Selection Based on STM Model

In the MOEA/D framework used in this paper, a MOP is decomposed into N subproblems, and each subproblem has one solution in the current population. The selection process is to choose the appropriate solution from a solution set $S = \{\mathbf{x}^1, \cdots, \mathbf{x}^M\}$ $(M > N)$ for each subproblem. Treating the subproblem set $P = \{p^1, \cdots, p^N\}$ and S as two independent agent sets, we propose using an STM model, which is derived from the classical SMP introduced in Section I, to implement the selection operator.

We assume that each agent has complete and transitive preferences over agents on the other side. For simplicity, we only consider the monotonic preference relations, which means that an agent can rank all those agents on the other side in a sequential order. Let $\mathbf{x}^i \succ_p \mathbf{x}^j$ denote that a given subproblem p prefers solution $\mathbf{x}^i$ to $\mathbf{x}^j$, and $p^i \succ_{\mathbf{x}} p^j$ indicate that a given solution $\mathbf{x}$ prefers subproblem p^i to p^j. A matching, which has N subproblem-solution pairs, is called stable if and only if the following two conditions hold.

1) If solution $\mathbf{x}$ is not paired with any subproblem, no subproblem p prefers $\mathbf{x}$ to its current paired solution;
2) If solution $\mathbf{x}$ is paired with a subproblem but not p, then $\mathbf{x}$ prefers its current partner to p, and p prefers its current partner to $\mathbf{x}$. In other words, pairing $\mathbf{x}$ and p cannot make both of them better off than they are with their current assigned partners.

Clearly, stable matching strikes a balance between the preferences of subproblems and solutions. Therefore, one should consider the convergence and diversity in setting preferences. Different preference settings can lead to different ways of balancing the convergence and diversity in the evolutionary search. For simplicity, this paper defines the preferences in the following way.

1) A subproblem p prefers solutions with low aggregation function values. Therefore, the preference value of p on solution $\mathbf{x}$, denoted as $\Delta_P(p, \mathbf{x})$, is defined as the aggregation function value of $\mathbf{x}$ for p
$$\Delta_P(p, \mathbf{x}) = g(\mathbf{x}|\mathbf{w}, \mathbf{z}^*) \tag{5}$$
where $\mathbf{w}$ is the weight vector of p and $g(*, *)$ is the aggregation function used in MOEA/D.
2) $\mathbf{x}$ prefers subproblems whose direction vectors are close to $\mathbf{x}$. More specifically, the preference value of $\mathbf{x}$ on p, denoted as $\Delta_X(\mathbf{x}, p)$, is defined as [25]
$$\Delta_X(\mathbf{x}, p) = \|\overline{\mathbf{F}}(\mathbf{x}) - \frac{\mathbf{w}^T \overline{\mathbf{F}}(\mathbf{x})}{\mathbf{w}^T \mathbf{w}} \mathbf{w}\| \tag{6}$$
where $\overline{\mathbf{F}}(\mathbf{x})$ is the normalized objective vector of $\mathbf{x}$, and its kth individual objective function is normalized as
$$\overline{f}_k(\mathbf{x}) = \frac{f_k(\mathbf{x}) - z_k^*}{z_k^{nad} - z_k^*} \tag{7}$$
where $k \in \{1, \cdots, m\}$. $\|\cdot\|$ is the ℓ_2 norm. $\Delta_X(\mathbf{x}, p)$ indicates the distance between $\mathbf{x}$ and the direction vector of p. As discussed in Section II-B, the optimal solution of a subproblem should lie on its direction vector. Therefore, a solution should prefer the subproblem that is closer to it.

Using the above preference values, we can easily obtain two preference ordering matrices of subproblems and solutions: Ψ_P and Ψ_X. The elements of the ith row in Ψ_P are the preference orderings of p^i on all the solutions in S, and the elements of the jth row in Ψ_X are the preference orderings of $\mathbf{x}^j$ on all the subproblems in P. Both of them are in the ascending order of preference values. The pseudo-code of computing Ψ_P and Ψ_X is presented in Algorithm 1.

Algorithm 1: COMPTPREF($S, P, \mathbf{z}^*, \mathbf{z}^{nad}$)

Input: solution set S, subproblem set P, the ideal and nadir objective vectors $\mathbf{z}^*$, $\mathbf{z}^{nad}$
Output: preference ordering matrices Ψ_P and Ψ_X
for $i \leftarrow 1$ **to** M **do**
  $\overline{\mathbf{F}}(\mathbf{x}^i) \leftarrow \frac{\mathbf{F}(\mathbf{x}^i) - \mathbf{z}^*}{\mathbf{z}^{nad} - \mathbf{z}^*}$;
end
for $i \leftarrow 1$ **to** M **do**
  for $j \leftarrow 1$ **to** N **do**
    $\Delta_P(p^j, \mathbf{x}^i) \leftarrow g(\mathbf{x}^i | \mathbf{w}^j, \mathbf{z}^*)$;
    $\Delta_X(\mathbf{x}^i, p^j) \leftarrow \| \overline{\mathbf{F}}(\mathbf{x}^i) - \frac{\mathbf{w}^{jT} \overline{\mathbf{F}}(\mathbf{x}^i)}{\mathbf{w}^{jT}\mathbf{w}^j} \mathbf{w}^j \|$;
  end
end
Sort each row of Δ_P and Δ_X in ascending order and keep the sorted indices in Ψ_P and Ψ_X;
return Ψ_P and Ψ_X

Algorithm 2: STM($S, P, \mathbf{z}^*, \mathbf{z}^{nad}$)

Input: solution set S, subproblem set P, the ideal and nadir objective vectors $\mathbf{z}^*$, $\mathbf{z}^{nad}$
Output: solution set $\overline{S}$
1 $\overline{S} \leftarrow \emptyset$;
2 **for** $i \leftarrow 1$ **to** N **do**
3   $F_P[i] \leftarrow 0$;
4 **end**
5 **for** $j \leftarrow 1$ **to** M **do**
6   $F_X[j] \leftarrow 0$;
7 **end**
8 **for** $i \leftarrow 1$ **to** M **do**
9   **for** $j \leftarrow 1$ **to** N **do**
10     $\Phi(i, j) \leftarrow 0$;
11   **end**
12 **end**
13 $[\Psi_P, \Psi_X] \leftarrow$ COMPTPREF($S, P, \mathbf{z}^*, \mathbf{z}^{nad}$);
14 **while** *some subproblems are still free* **do**
15   Randomly choose a subproblem p^i with $F_P[i] = 0$;
16   Find p^i's most preferred solution $\mathbf{x}^j$ with $\Phi(i, j) = 0$;
17   $\Phi(i, j) \leftarrow 1$;
18   **if** $F_X[j] = 0$ **then**
19   p^i and $\mathbf{x}^j$ are set to be paired;
20     $\overline{S} \leftarrow \overline{S} \cup \{\mathbf{x}^j\}$;
21     $F_P[i] \leftarrow 1$, $F_X[j] \leftarrow 1$;
22   **else**
23   **if** $p^i \succ_{\mathbf{x}^j} p^k$ *(the current partner of* $\mathbf{x}^j$*)* **then**
24   p^i and $\mathbf{x}^j$ are set to be paired;
25       $F_P[i] \leftarrow 1$, $F_P[k] \leftarrow 0$;
26   **end**
27   **end**
28 **end**
29 **return** $\overline{S}$

The deferred acceptance procedure suggested in [17] is adopted in Algorithm 2 to find a stable matching between subproblems and solutions, and thus to select a set of solutions $\overline{S}$ from the solution set S. Algorithm 2 first initializes $\overline{S}$ as an empty set in line 1 and sets all subproblems and solutions to be free (i.e., not paired with any other agent on the other side) from line 2 to line 12. In Algorithm 2, $F_P[i] = 0$ means that p^i is free, and 1 means that it is paired. In the same way, $F_X[j]$ indicates whether or not $\mathbf{x}^j$ is free. $\Phi(i, j) = 0$ means that p^i has not proposed to pair with $\mathbf{x}^j$ before, and 1 means that it has done so. Line 13 calls Algorithm 1 to compute Ψ_P and Ψ_X. In the main while-loop, when some subproblems are still free, the algorithm randomly chooses such a subproblem p^i and finds out its highest ranked solution $\mathbf{x}^j$, to which p^i has not proposed yet (line 15 to line 16). If $\mathbf{x}^j$ is also free, then p^i and $\mathbf{x}^j$ will be paired with each other (line 19 to line 21). In the case when $\mathbf{x}^j$ is not free, $\mathbf{x}^j$ will be broken with its current partner p^k and paired with p^i if and only if $p^i \succ_{\mathbf{x}^j} p^k$ (line 24 to line 25). The main while-loop terminates when exactly N subproblem-solution pairs have been formed; in other words, there is no free subproblem. The solutions paired with subproblems form the output $\overline{S}$.

Consider an example with five subproblems and 10 solutions shown in Fig. 3(a). The preference ordering matrices of subproblems and solutions, Ψ_P and Ψ_X, are as follows:

$$\Psi_X = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 5 & 3 & 2 & 1 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 1 & 4 & 5 \\ 3 & 4 & 2 & 5 & 1 \\ 3 & 4 & 2 & 5 & 1 \\ 4 & 5 & 3 & 2 & 1 \\ 5 & 4 & 3 & 2 & 1 \\ 5 & 4 & 3 & 2 & 1 \end{bmatrix} \tag{8}$$

$$\Psi_P = \begin{bmatrix} 1 & 3 & 4 & 2 & 5 & 8 & 7 & 6 & 9 & 10 \\ 1 & 4 & 3 & 2 & 5 & 8 & 7 & 6 & 9 & 10 \\ 2 & 1 & 5 & 8 & 4 & 7 & 3 & 6 & 9 & 10 \\ 2 & 8 & 9 & 10 & 1 & 5 & 7 & 4 & 6 & 3 \\ 9 & 2 & 10 & 8 & 1 & 5 & 7 & 4 & 6 & 3 \end{bmatrix}. \tag{9}$$

Then, the stable matching produced by Algorithm 2 is $\{(p^1, \mathbf{x}^1), (p^2, \mathbf{x}^4), (p^3, \mathbf{x}^5), (p^4, \mathbf{x}^2), (p^5, \mathbf{x}^9)\}$. As a result, $\{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^4, \mathbf{x}^5, \mathbf{x}^9\}$ will be selected to form $\overline{S}$.

Following [17], we can have the following results.

Theorem 1: The matching generated by Algorithm 2 is stable.

Proof: Note that a subproblem makes its proposals to solutions according to the preference orderings in Algorithm 2. No subproblem prefers a free solution to its paired solution. Therefore, the first condition in the stable matching definition is met.

If solution $\mathbf{x}$ is paired with a subproblem p' but not p, and if p prefers $\mathbf{x}$ to its assigned partner, then p has made its proposal to $\mathbf{x}$. There are two possibilities. The first one is that $\mathbf{x}$ prefers its then paired subproblem to p and has rejected p, and the second possibility is that $\mathbf{x}$ has accepted p at some stage but

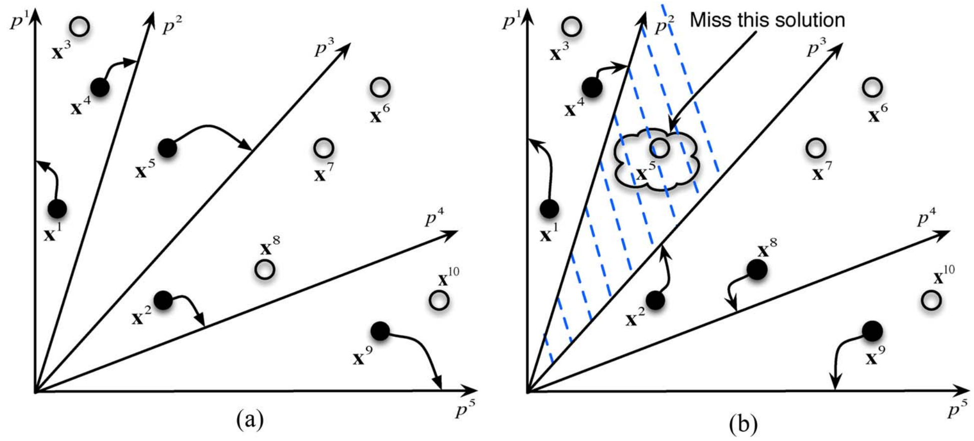


Fig. 3. Illustrative examples of matching relationships. (a) Matching result obtained by STM model. (b) Matching result obtained by MOEA/D.

later left p for another subproblem. Note that a solution always prefers its new partner to its previous ones in Algorithm 2. We can conclude that the final matching generated meets the second condition in the stable matching definition. ■

Theorem 2: If $\mathbf{x}$ is the best solution in S for a suproblem, then $\mathbf{x} \in \bar{S}$.

Proof: If $\mathbf{x}$ is the unique best solution in S for p, then p will make its first proposal to $\mathbf{x}$ in Algorithm 2. If $\mathbf{x}$ is added to $\bar{S}$ in Algorithm 2, $\mathbf{x}$ will not be removed from $\bar{S}$. Therefore, $\mathbf{x} \in \bar{S}$. ■

Theorem 1 implies that Algorithm 2 is able to balance the preferences of both subproblem agents and solution agents. Subproblem agents' preferences encourage the search convergence and solution agents' preferences promote the population diversity. Therefore, Algorithm 2 provides a systematic and rational approach for trading off the convergence and diversity in MOEAs. On the other hand, Theorem 2 guarantees that each subproblem can be optimized as much as possible in each generation.

B. Discussions

In the original MOEA/D proposed in [10], the matching relationship is randomly assigned at first and gradually refined during the evolutionary search. Each solution is associated with a different subproblem, and neighborhood relationships among all the subproblems are defined beforehand. A new solution $\mathbf{x}$ is generated by using the information extracted from solutions of the neighboring subproblems of a subproblem p. $\mathbf{x}$ will replace the solutions of one or more neighboring subproblems if $\mathbf{x}$ has a better aggregation function value for these subproblems. In a sense, only the subproblems have their preferences on solutions, while solutions have not explicitly expressed their preferences on subproblems. Thus, the established matching relationship may not be stable.

For example, consider the same case in Fig. 3(a), the selection of the original MOEA/D may produce a matching $\{(p^1, \mathbf{x}^1), (p^2, \mathbf{x}^4), (p^3, \mathbf{x}^2), (p^4, \mathbf{x}^8), (p^5, \mathbf{x}^9)\}$, as shown in Fig. 3(b). However, $p^4 \succ_{\mathbf{x}^2} p^3$, $\mathbf{x}^2 \succ_{p^4} \mathbf{x}^8$, in other words, $\mathbf{x}^2$ and p^4 prefer each other to their assigned partners. Therefore, this matching is not stable. In addition, $\mathbf{x}^5$, which is located in the subregion between the direction vectors of p^2 and p^3 (highlighted as the dotted lines), is important for preserving the population diversity. In contrast to the matching result of Algorithm 2, $\mathbf{x}^5$ is not selected by MOEA/D. Therefore, the solutions selected by Algorithm 2 have a better spread over the objective space.

C. Computational Cost of STM Model

Let us first consider the computational cost of Algorithm 1, which obtains the preference ordering matrices of subproblems and solutions. The normalization of the objective function values of each solution (line 1 to line 3 in Algorithm 1) requires $O(mM)$ computations. Next, the calculation of the mutual preferences between subproblems and solutions (line 4 to line 9 in Algorithm 1) costs $O(mMN)$ computations. Finally, sorting Δ_P and Δ_X (line 10 in Algorithm 1) requires $O(NMlogM)$ and $O(MNlogN)$ comparisons, respectively. Therefore, the overall complexity of Algorithm 1 is $O(NMlogM)$, since $M = 2N$ in our case. Consider the main body of the STM model given in Algorithm 2, the initialization process (line 1 to line 12 in Algorithm 2) requires $O(NM)$ assignments. Then, the evaluations of Ψ_P and Ψ_X cost $O(NMlogM)$ operations as discussed before. At last, similar to the discussion of [17], the worst case complexity of the deferred acceptance procedure (the main-while loop, from line 14 to line 28, in Algorithm 2) is $O(NM)$. In summary, the computational cost of the STM model is $O(NMlogM)$.

IV. Integration of STM Model With MOEA/D

In this section, we present an instantiation of MOEA/D, which uses the selection operator proposed in Section III. This instantiation, referred to as the MOEA/D-STM, is derived from MOEA/D-DRA [14], a MOEA/D variant with a dynamic resource allocation scheme. MOEA/D-DRA was the winner in the CEC2009 MOEA competition [26]. It is worthwhile noting that the only difference between MOEA/D-STM and MOEA/D-DRA is in selection. The pseudo-code of MOEA/D-STM is given in Algorithm 3. Some important components of MOEA/D-STM are further illustrated in the following.

A. Initialization

We assume that we have no prior knowledge about the position of the PS. The initial population $P_1 = \{\mathbf{x}^1, \cdots, \mathbf{x}^N\}$

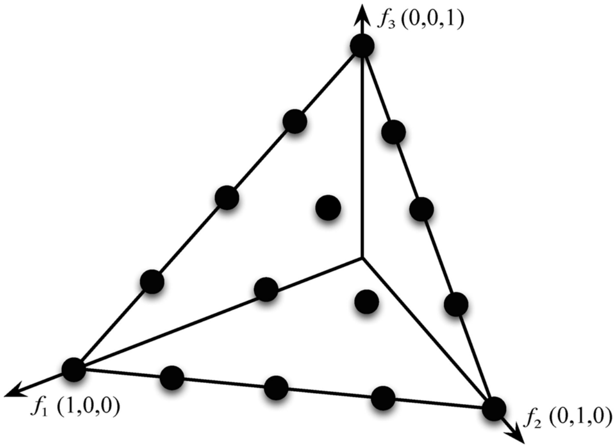


Fig. 4. Fifteen weight vectors are sampled from the simplex in the 3-D space with $H = 4$.

is randomly sampled from the decision space. Since the ideal and nadir objective vectors are unknown *a priori*, we use their approximations that are respectively set as the minimum and maximum F-function values on each objective in the current population, i.e., $z_i^* = \min\{f_i(\mathbf{x})|\mathbf{x} \in P_t\}$ and $z_i^{nad} = \max\{f_i(\mathbf{x})|\mathbf{x} \in P_t\}$, for all $i \in \{1, \cdots, m\}$, where t is the generation counter.

In generating a set of weight vectors, each element of a weight vector $\mathbf{w}$ takes its value from $\{\frac{0}{H}, \frac{1}{H} \cdots, \frac{H}{H}\}$, where H is a user-defined integer. The total number of such weight vectors is $N = C_{H+m-1}^{m-1}$. Fig. 4 illustrates the weight vectors generated by setting $m = 3$ and $H = 4$. In this case, $N = 15$ evenly distributed weight vectors are generated. In MOEA/D, each weight vector is given a neighborhood, which includes its T $(1 \leq T \leq N)$ nearest weight vectors.

B. Reproduction

Reproduction operator is to generate an offspring population $Q_t = \{\bar{\mathbf{x}}^1, \cdots, \bar{\mathbf{x}}^N\}$ of N members from population P_t. Any genetic operator or mathematical programming technique can serve this purpose. In this paper, we use the differential evolution (DE) operator [27] and polynomial mutation [28] as in [13]. More specifically, let $\mathbf{x}^{r_1}$, $\mathbf{x}^{r_2}$, and $\mathbf{x}^{r_3}$ be three parent solutions, an offspring solution $\bar{\mathbf{x}}^i = (\bar{x}_1^i, \cdots, \bar{x}_n^i)$ is generated as follows:

$$u_j^i = \begin{cases} x_j^{r_1} + F \times (x_j^{r_2} - x_j^{r_3}) & \text{if } rand < CR \text{ or } j = j_{rand} \\ x_j^i & \text{otherwise} \end{cases} \tag{10}$$

where $j \in \{1, \cdots, n\}$, $rand \in [0, 1]$, CR and F are two control parameters and j_{rand} is a random integer uniformly chosen from 1 to n. Then, the polynomial mutation is applied on each $\mathbf{u}^i$ to generate $\bar{\mathbf{x}}^i$

$$\bar{x}_j^i = \begin{cases} u_j^i + \sigma_j \times (b_j - a_j) & \text{if } rand < p_m \\ u_j^i & \text{otherwise} \end{cases} \tag{11}$$

with

$$\sigma_j = \begin{cases} (2 \times rand)^{\frac{1}{\eta+1}} - 1 & \text{if } rand < 0.5 \\ 1 - (2 - 2 \times rand)^{\frac{1}{\eta+1}} & \text{otherwise} \end{cases} \tag{12}$$

where the distribution index η and the mutation rate p_m are two control parameters. a_j and b_j are the lower and upper bounds of the jth decision variable. For simplicity, the violated decision variable is set to its nearer boundary value.

C. Utility of Subproblem

The utility of subproblem p^i, denoted as π^i, where $i \in \{1, \cdots, N\}$, measures how much improvement has been achieved by its current solution $\mathbf{x}^{new}$ in reducing the aggregation function value of p^i. Formally, it is defined as [14]

$$\pi^i = \begin{cases} 1 & \text{if } \Delta^i > 0.001 \\ (0.95 + 0.05 \times \frac{\Delta^i}{0.001}) \times \pi^i & \text{otherwise} \end{cases} \tag{13}$$

where Δ^i represents the relative decrease of the aggregation function value of p^i, and it is evaluated as

$$\Delta^i = \frac{g(\mathbf{x}^{old}|\mathbf{w}^i, \mathbf{z}^*) - g(\mathbf{x}^{new}|\mathbf{w}^i, \mathbf{z}^*)}{g(\mathbf{x}^{old}|\mathbf{w}^i, \mathbf{z}^*)} \tag{14}$$

where $\mathbf{x}^{old}$ is the solution of p^i in the previous generation.

V. Experimental Setting

A. Test Instances

Ten unconstrained MOP test instances from the CEC2009 MOEA competition [26] (UF1 to UF10) are used in our experimental studies. The number of decision variables of the UF instances is set to be 30.

B. Performance Metrics

No unary performance metric can give a comprehensive assessment on the performance of a MOEA [29], [30]. In our experimental studies, we employ the following two widely used [31], [32] performance metrics.

1) Inverted Generational Distance (IGD) metric [33]: Let P^* be a set of points uniformly sampled along the PF, and S be the set of solutions obtained by some given MOEA. The IGD value of S is calculated as

$$IGD(S, P^*) = \frac{\sum_{\mathbf{x} \in P^*} dist(\mathbf{x}, S)}{|P^*|} \tag{15}$$

where $dist(\mathbf{x}, S)$ is the Euclidean distance between the solution $\mathbf{x}$ and its nearest point in S, and $|P^*|$ is the cardinality of P^*. The PF of the underlying MOP is assumed to be known a priori when using the IGD metric. In our empirical studies, 1000 uniformly distributed points are sampled along the PF for the bi-objective test instances, and 10 000 for three-objective cases, respectively.

2) Hypervolume (HV) metric [34]: Let $\mathbf{z}^r = (z_1^r, \ldots, z_m^r)^T$ be a reference point in the objective space that is dominated by all Pareto-optimal objective vectors. HV metric measures the size of the objective space dominated by the solutions in S and bounded by $\mathbf{z}^r$

$$HV(S) = \text{VOL}\Big(\bigcup_{\mathbf{x} \in S} [f_1(\mathbf{x}), z_1^r] \times \ldots [f_m(\mathbf{x}), z_m^r]\Big) \tag{16}$$

Algorithm 3: MOEA/D-STM

```
Initialize the population P ← {x^1, ··· , x^N}, a set of
    weight vectors W ← {w^1, ··· , w^N}, the ideal and nadir
    objective vectors z*, z^nad;
Set neval ← 0, iteration ← 0;
for i ← 1 to N do
    B(i) ← {i_1, ··· , i_T} where w^{i_1}, ··· , w^{i_T} are the T
        closest weight vectors to w^i;
    π^i ← 1;
end
while Stopping criterion is not satisfied do
    Let all indices of the subproblems whose objectives
        are MOP individual objectives f_i form the initial I.
        By using 10-tournament selection based on π^i, select
        other ⌊N/5⌋ − m indices and add them to I.
    Q ← ∅;
    for each i ∈ I do
        if uniform(0, 1) < δ then
            E ← B(i);
        else
            E ← P;
        end
        Randomly select three solutions x^{r_1}, x^{r_2}, and x^{r_3}
            from E;
        Generate a candidate x̄ by using the method
            described in Section IV-B and Q ← Q ⋃ {x̄};
        Evaluate the F-function value of x̄;
        Update the current ideal objective vector z*;
        Update the current nadir objective vector z^nad;
        neval++;
    end
    R ← P ∪ Q;
    P ← STM(R, W, z*, z^nad);
    iteration++;
    if mod(iteration, 30) = 0 then
        Update the utility of each subproblem;
    end
end
return P;
```

where $\textsc{Vol}(\cdot)$ is the Lebesgue measure. In our empirical studies, $\mathbf{z}^r = (2.0, 2.0)^T$ for bi-objective test instances and $\mathbf{z}^r = (2.0, 2.0, 2.0)^T$ for three-objective cases, respectively.

To a certain extent, both IGD and HV metrics can measure the convergence and diversity of S. The lower the IGD value is (the larger the HV value is), the better the quality of S for approximating the whole PF. In the comparison tables of the following sections, the best mean metric values are highlighted in bold face with gray background. In order to have statistically sound conclusions, Wilcoxon's rank sum test at a 5% significance level is conducted to compare the significance of difference between the metric values of two algorithms.

C. MOEAs in Comparison

Five MOEAs are used in this paper to compare with our proposed algorithm.

1) NSGA-II [4]: it is the most popular dominance-based algorithm. It is characterized by its fast nondominated sorting procedure for emphasizing the convergence and its crowding distance for maintaining population diversity. As in [13], we use the reproduction method described in Section IV-B to generate new solutions.
2) MSOPS-II [35]: it is a decomposition-based MOEA. As an extension of MSOPS [12], it is featured by an automatic target vector generation scheme and an improved fitness assignment method.
3) HypE [9]: it is a well known indicator-based MOEA, which uses the HV metric as the guideline of its selection process. In order to reduce the computational cost in its HV calculation, HypE employs Monte Carlo simulation to approximate the HV value.
4) MOEA/D-DE [13]: it is a variant of MOEA/D [10]. It uses the reproduction method described in Section IV-B to generate new solutions. In order to maintain the population diversity, a new solution is allowed to replace only a small number of old solutions.
5) MOEA/D-DRA [14]: it is another variant of MOEA/D, which won the CEC2009 MOEA competition. In this algorithm, different subproblems will receive different computational resources based on their utility values.

In MOEA/D-STM, MOEA/D-DE, and MOEA/D-DRA, the TCH approach explained in (3) is adopted as the decomposition method.

D. General Parameter Settings

The parameters of NSGA-II, MSOPS-II, HypE, MOEA/D-DE, and MOEA/D-DRA are set according to their corresponding references [4], [9], [13], [14], [35], respectively. All these MOEAs are implemented in JAVA,[2] except MSOPS-II in MATLAB,[3] and HypE in ANSI C.[4] The detailed parameter settings of our proposed MOEA/D-STM are summarized as follows.

1) Settings for reproduction operators: The mutation probability $p_m = 1/n$ and its distribution index is set to be 20, i.e., $\mu_m = 20$ [36]. For the DE operator, we set $CR = 1.0$ and $F = 0.5$ as recommended in [13].
2) Population size: $N = 600$ for bi-objective test instances, 1000 for the three-objective ones.
3) Number of runs and stopping condition: Each algorithm is run 30 times independently on each test instance. The algorithm stops after 300 000 function evaluations.
4) Neighborhood size: $T = 20$.
5) Probability used to select in the neighborhood: $\delta = 0.9$.

[2]The source codes are from the jMetal 4.2 at http://jmetal.sourceforge. net.
[3]The source code is from http://code.evanhuges.org.
[4]The source code is from http://www.tik.ee.ethz.ch/sop/download/supplementary/hype/.

TABLE I
IGD RESULTS OF MOEA/D-STM AND FIVE OTHER MOEAS ON UF TEST INSTANCES

Problem	MOEA/D-STM	MOEA/D-DE	MOEA/D-DRA	NSGA-II	MSOPS-II	HypE
UF1	**1.064E-3(6.86E-5)**	1.332E-3(9.59E-5)†	1.516E-3(8.28E-4)†	3.478E-2(3.69E-3)†	7.348E-2(8.56E-3)†	1.045E-1(1.29E-2)†
UF2	**2.692E-3(1.17E-3)**	5.612E-3(2.06E-3)†	5.417E-3(2.98E-3)†	4.435E-2(2.54E-3)†	4.515E-2(6.46E-3)†	7.153E-2(5.46E-2)†
UF3	**6.754E-3(7.79E-3)**	9.985E-3(1.23E-2)†	8.547E-3(1.25E-2)†	6.806E-2(1.26E-2)†	3.167E-1(1.45E-2)†	2.178E-1(6.21E-2)†
UF4	**5.194E-2(3.24E-3)**	5.621E-2(3.37E-3)†	5.495E-2(4.23E-3)†	5.857E-2(1.93E-3)†	6.828E-2(3.00E-3)†	6.151E-2(1.21E-2)†
UF5	**2.471E-1(3.17E-2)**	3.052E-1(4.47E-2)†	2.911E-1(7.12E-2)†	1.547E+0(1.10E-1)†	3.337E-1(9.38E-2)†	2.949E-1(4.43E-2)†
UF6	**7.031E-2(2.72E-2)**	1.026E-1(1.01E-1)†	9.601E-2(4.30E-2)†	3.987E-1(2.03E-2)†	1.560E-1(5.72E-2)†	2.104E-1(8.01E-2)†
UF7	**1.114E-3(1.11E-4)**	1.492E-3(3.41E-4)†	1.123E-3(1.16E-4)	1.490E-2(9.39E-4)†	4.228E-2(6.23E-3)†	2.182E-1(1.14E-1)†
UF8	**2.250E-2(1.46E-3)**	5.672E-2(9.31E-3)†	3.577E-2(2.53E-3)†	1.466E-1(8.73E-3)†	1.787E-1(1.75E-3)†	4.121E-1(3.13E-2)†
UF9	**2.100E-2(8.45E-4)**	4.515E-2(3.06E-2)†	3.037E-2(2.17E-2)†	9.912E-2(3.81E-2)†	2.536E-1(1.08E-2)†	1.612E-1(6.31E-2)†
UF10	8.054E-1(1.76E-1)	5.372E-1(5.97E-2)‡	4.555E-1(4.75E-2)‡	2.585E+0(1.84E-1)†	**3.643E-1(8.48E-2)**‡	8.902E-1(2.01E-1)†

Wilcoxon's rank sum test at a 0.05 significance level is performed between MOEA/D-STM and each of the other competing algorithms. † and ‡ denotes that the performance of the corresponding algorithm is significantly worse than or better than that of MOEA/D-STM, respectively. The best mean is highlighted in boldface with gray background. UF1 to UF7 have two objectives and UF8 to UF10 have three objectives.

TABLE II
HV RESULTS OF MOEA/D-STM AND FIVE OTHER MOEAS ON UF TEST INSTANCES

Problem	MOEA/D-STM	MOEA/D-DE	MOEA/D-DRA	NSGA-II	MSOPS-II	HypE
UF1	**3.6631(4.74E-4)**	3.6609(9.22E-4)	3.6531(7.15E-3)†	3.5993(6.44E-3)†	3.3682(1.10E-2)†	3.3515(9.76E-2)†
UF2	**3.6575(8.44E-3)**	3.6419(6.57E-3)†	3.6465(1.31E-2)†	3.5652(4.02E-3)†	3.4662(5.15E-2)†	3.3898(4.02E-2)†
UF3	**3.6537(1.31E-2)**	3.6308(7.01E-2)†	3.6411(5.34E-2)†	3.5434(1.86E-2)†	2.5602(1.71E-2)†	2.7941(1.91E-1)†
UF4	**3.1815(1.40E-2)**	3.1674(1.68E-2)†	3.1709(1.42E-2)†	3.1637(7.04E-3)†	3.1623(1.02E-2)†	3.1298(1.03E-1)†
UF5	**2.9426(8.94E-2)**	2.6504(1.27E-1)†	2.6990(1.97E-1)†	7.397E-1(1.81E-1)†	2.3612(2.01E-2)†	2.6528(1.93E-1)†
UF6	**3.2072(5.36E-2)**	3.1008(2.69E-1)†	3.1080(1.33E-1)†	2.4190(7.46E-2)†	2.9277(2.13E-1)†	2.7910(2.23E-1)†
UF7	**3.4968(5.97E-4)**	3.4916(5.16E-3)	3.4962(1.03E-3)	3.4688(1.86E-3)†	3.3933(6.52E-2)†	2.8346(3.19E-1)†
UF8	**7.4241(2.91E-3)**	7.3360(1.69E-2)†	7.3575(4.89E-3)†	7.0026(4.30E-2)†	6.4238(3.50E-3)†	6.4460(7.51E-2)†
UF9	**7.7541(3.64E-3)**	7.5810(1.35E-1)†	7.6565(1.02E-1)†	7.4400(3.48E-2)†	6.6770(1.23E-2)†	6.9863(3.69E-1)†
UF10	2.5199(6.15E-1)	3.3291(2.39E-1)‡	3.6674(2.59E-1)‡	1.240E-3(6.69E-3)†	**4.1722(1.23E-2)**‡	2.4211(6.27E-1)†

Wilcoxon's rank sum test at a 0.05 significance level is performed between MOEA/D-STM and each of the other competing algorithms. † and ‡ denote whether the performance of the corresponding algorithm is significantly worse than or better than that of MOEA/D-STM, respectively. The best mean is highlighted in boldface with gray background. UF1 to UF7 have two objectives and UF8 to UF10 have three objectives.

VI. EMPIRICAL RESULTS AND DISCUSSION

A. Performance Comparisons With Other MOEAs

Comparison results of MOEA/D-STM with the other five MOEAs in terms of IGD and HV metric values are presented in Tables I and II. From these results, it is clear that MOEA/D-STM has obtained the best mean metric values on all the test instances except UF10. UF10 has many local PFs. No algorithm can approximate its PF very well. MOEA/D-DE and MOEA/D-DRA have demonstrated much better performance than MSOPS-II on all test instances except UF10. NSGA-II is better than MSOPS-II and HypE on most test instances. The indicator-based algorithm, HypE, performs worse than MOEA/D-STM on all test instances.

Figs. 5–9 plot the evolution of the median IGD metric value versus the number of function evaluations in each algorithm on each test instance. A typical phenomenon observed from these figures is that MOEA/D-STM, which is often not the best at early stage, can win other algorithms at late stage. This could be due to the fact that MOEA/D-STM does try to promote and balance both the convergence and population diversity via its selection throughout the entire search process, whereas other MOEAs often promote the convergence at their early search stage and the population diversity at their late search stage. For most test instances, other algorithms can easily stagnate due to the loss of population diversity after a number of generations, while MOEA/D-STM can still make progress with its well maintained population diversity.

B. STM Model Versus Other Matching Variants

In order to further investigate the underlying rationality of the STM model, we compare it with the following two different matching schemes.

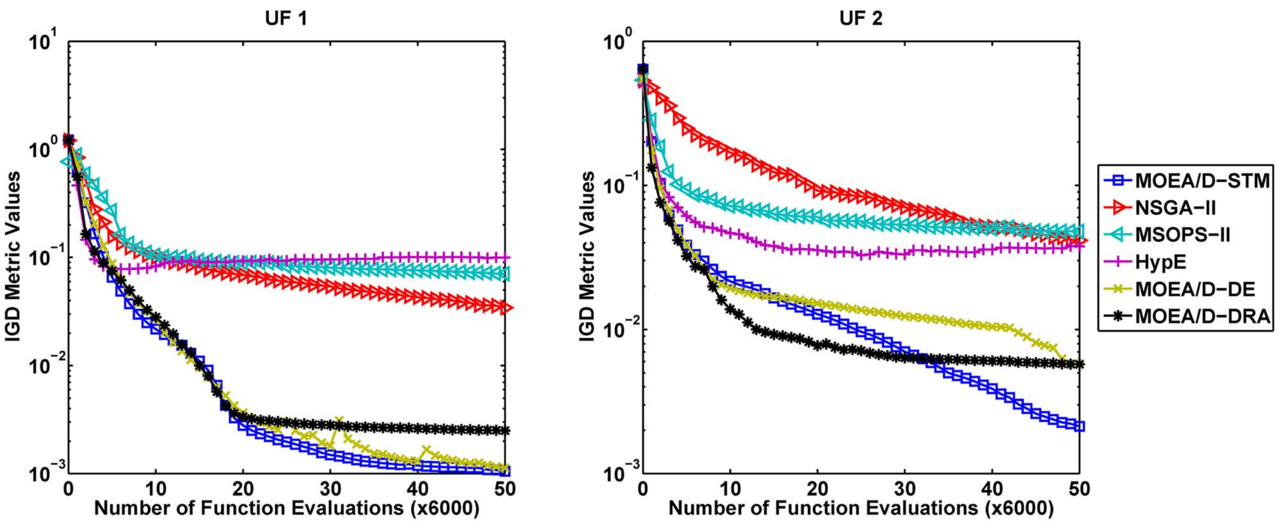


Fig. 5. Evolution of the median IGD metric values versus the number of function evaluations.

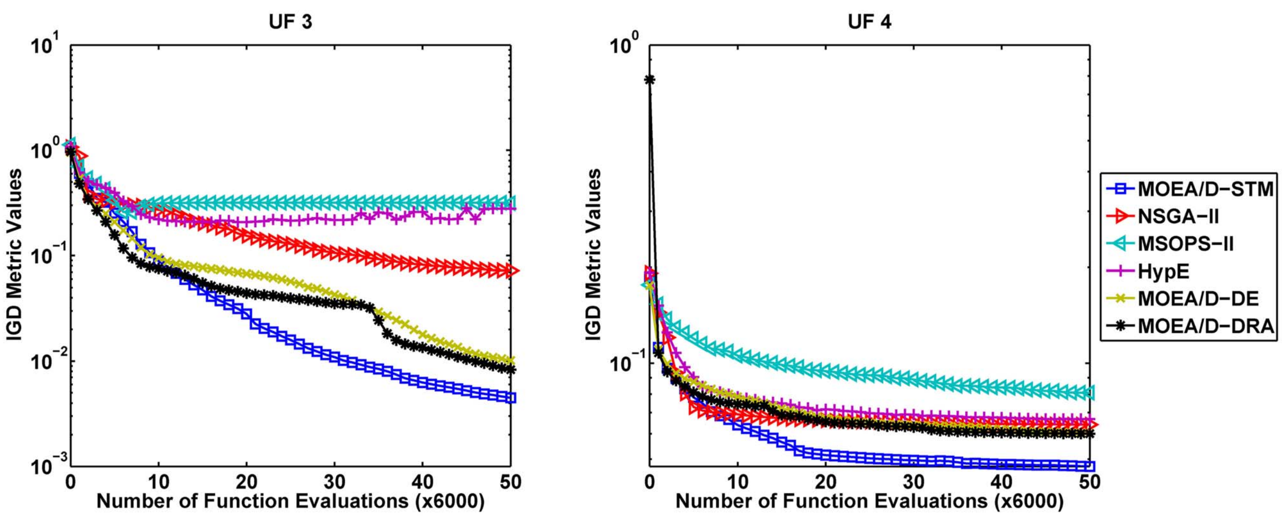


Fig. 6. Evolution of the median IGD metric values versus the number of function evaluations.

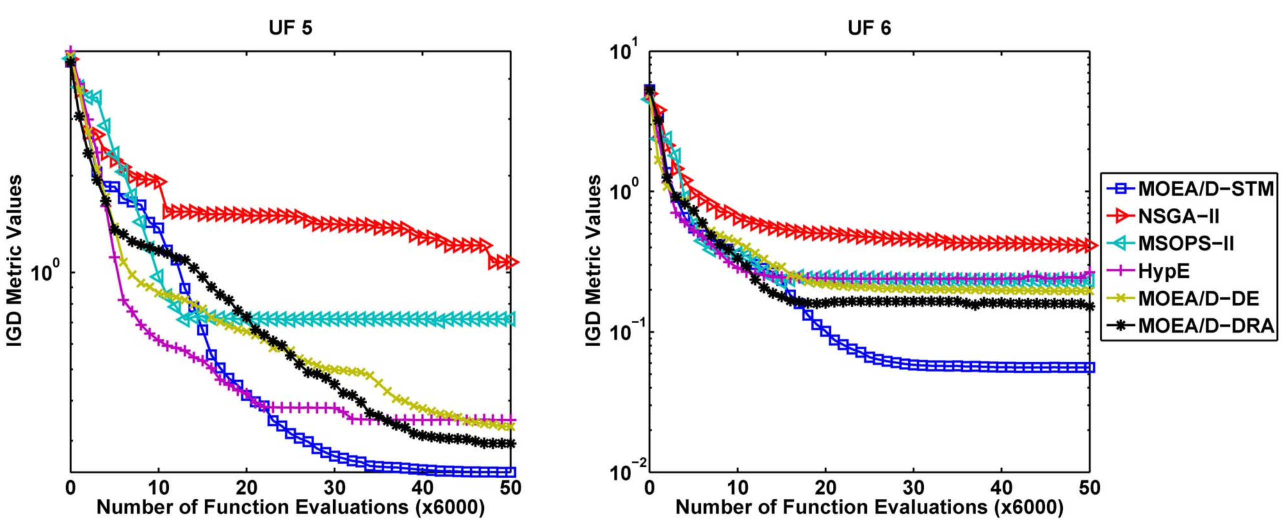


Fig. 7. Evolution of the median IGD metric values versus the number of function evaluations.

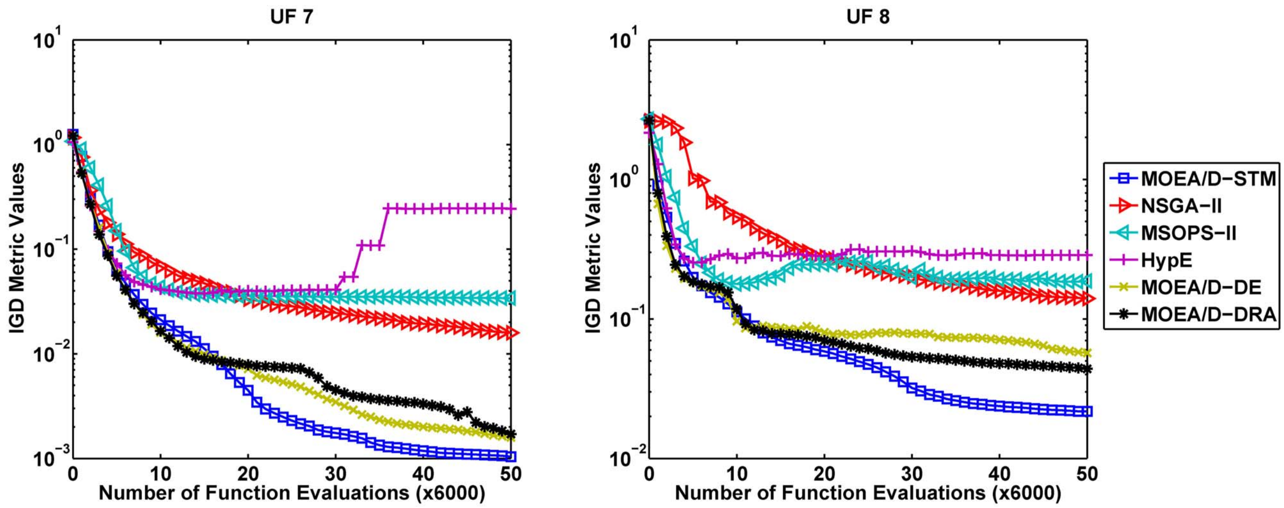


Fig. 8. Evolution of the median IGD metric values versus the number of function evaluations.

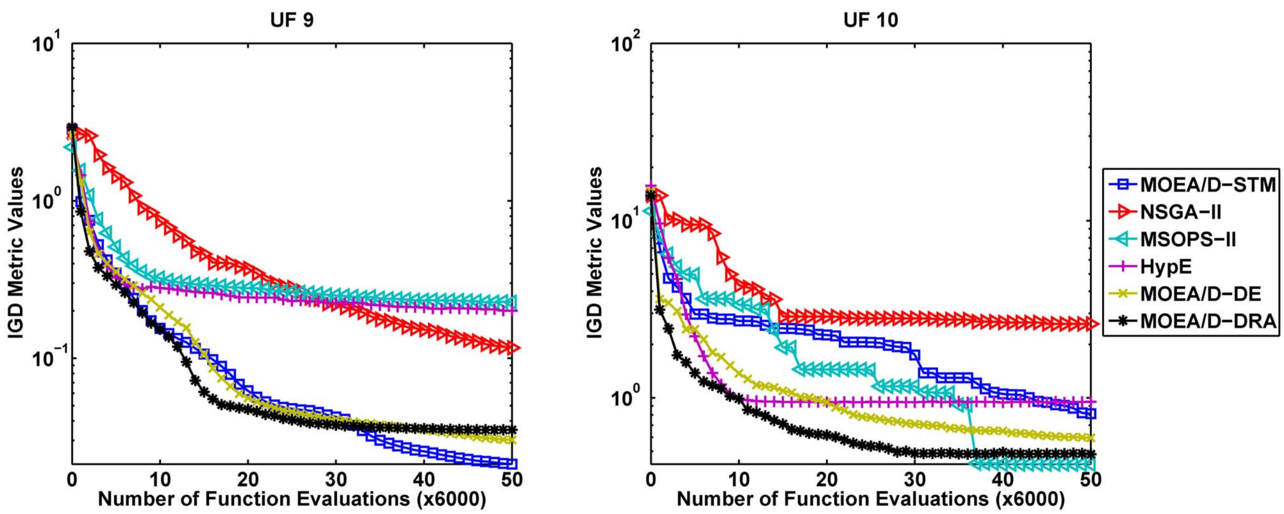


Fig. 9. Evolution of the median IGD metric values versus the number of function evaluations.

TABLE III
PERFORMANCE COMPARISONS OF MOEA/D-STM AND ITS TWO VARIANTS

	IGD			HV		
Test Instance	MOEA/D-STM	*Variant*-I	*Variant*-II	MOEA/D-STM	*Variant*-I	*Variant*-II
UF1	**1.064E-3(6.86E-5)**	1.456E-3(7.61E-5)†	3.157E-3(1.31E-3)†	**3.6631(4.74E-4)**	3.6602(7.27E-4)	3.6583(2.80E-3)†
UF2	2.692E-3(1.17E-3)	**1.885E-3(9.05E-4)**‡	3.432E-2(1.02E-2)†	3.6575(8.44E-3)	**3.6613(2.95E-3)**‡	3.6068(1.97E-2)†
UF3	**6.754E-3(7.79E-3)**	1.255E-2(7.13E-3)†	2.277E-2(1.97E-2)†	**3.6537(1.31E-2)**	3.6355(3.90E-2)†	3.6243(3.36E-2)†
UF4	**5.194E-2(3.24E-3)**	5.388E-2(3.90E-3)†	5.246E-2(3.33E-3)	**3.1815(1.40E-2)**	3.1712(1.46E-2)†	3.1793(1.42E-2)
UF5	**2.471E-1(3.17E-2)**	3.913E-1(1.34E-1)†	2.585E-1(3.42E-2)†	**2.9426(8.94E-2)**	2.2029(3.64E-1)†	2.9106(9.09E-1)†
UF6	**7.031E-2(2.72E-2)**	5.106E-1(2.25E-1)†	1.109E-1(7.16E-2)†	**3.2072(5.36E-2)**	2.5069(5.46E-1)†	3.1142(1.70E-1)†
UF7	**1.114E-3(1.11E-4)**	1.969E-3(9.15E-2)†	9.045E-3(1.28E-1)†	**3.4968(5.97E-4)**	3.4290(2.27E-1)†	3.2953(2.92E-1)†
UF8	**2.250E-2(1.46E-3)**	3.710E-2(2.30E-2)†	2.965E-2(7.45E-3)†	**7.4241(2.91E-3)**	7.4043(3.65E-2)	7.4033(1.63E-2)
UF9	**2.100E-2(8.45E-4)**	1.222E-1(4.63E-2)†	9.831E-2(1.02E-1)†	**7.7541(3.64E-3)**	7.3129(2.20E-1)†	7.6301(1.49E-1)†
UF10	8.054E-1(1.76E-1)	**3.887E-1(1.13E-1)**‡	8.043E-1(1.90E-1)	2.5199(6.15E-1)	**4.0733(1.15E+0)**‡	2.7414(6.71E-1)‡

Wilcoxon's rank sum test at a 0.05 significance level is performed between MOEA/D-STM Variant-I and Variant-II. † and ‡ denote whether the performance of the corresponding algorithm is significantly worse than or better than that of MOEA/D-STM, respectively. The best mean is highlighted in boldface with gray background. UF1 to UF7 have two objectives and UF8 to UF10 have three objectives.

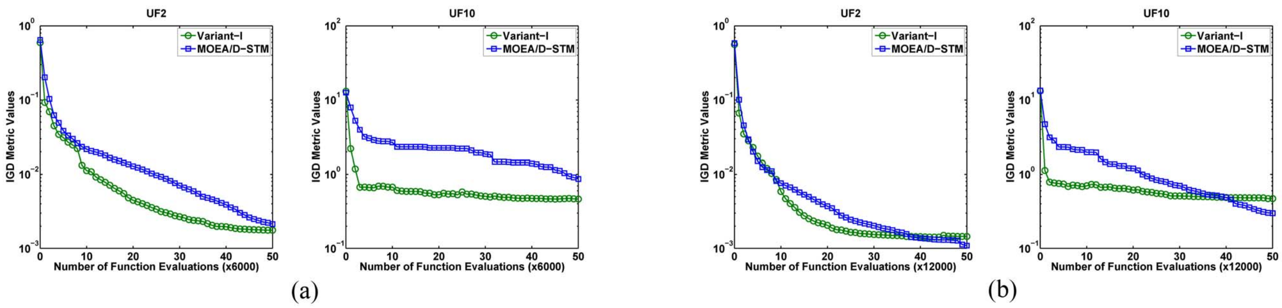


Fig. 10. Evolution of the median IGD metric values versus the number of function evaluations. (a) Comparisons with 300 000 function evaluations. (b) Comparisons with 600 000 function evaluations.

1) *Variant-I*: Instead of finding the stable matching between subproblems and solutions, in this variant, each subproblem chooses the best solution (in terms of its aggregation function value) in S as its solution. In this case, different subproblems can be assigned with the same solution.
2) *Variant-II*: The only difference of this variant from Algorithm 2 is that a solution agent, instead of a subproblem agent, makes a proposal to its most preferred agent of the other side that has not rejected it before. The pseudo-code of this variant is given in Algorithm 4, where $\chi(i, j) = 0$ indicates that $\mathbf{x}^i$ has not proposed to pair with p^j before, and 1 means that it has done so. The main while-loop in Algorithm 4 (line 14 to line 28) terminates when the number of free solutions is equal to the number of subproblems.

Algorithm 4: VARIANT-II(S, P, $\mathbf{z}^*$, $\mathbf{z}^{nad}$)

Input: solution set S, subproblem set P, the ideal and nadir objective vectors $\mathbf{z}^*$, $\mathbf{z}^{nad}$
Output: solution set $\overline{S}$

```
1  S̄ ← ∅;
2  for i ← 1 to N do
3  |  F_P[i] ← 0;
4  end
5  for j ← 1 to M do
6  |  F_X[j] ← 0;
7  end
8  for i ← 1 to N do
9  |  for j ← 1 to M do
10 |  |  χ(i, j) ← 0;
11 |  end
12 end
13 [Ψ_P, Ψ_X] ← COMPTPREF(S, P, z*, z^nad);
14 while some solutions are still free do
15 |  Randomly choose a solution x^i with F_X[i] = 0;
16 |  Find x^i's most preferred subproblem p^j with
   |  χ(i, j) = 0;
17 |  χ(i, j) ← 1;
18 |  if F_P[j] = 0 then
19 |  |  x^i and p^j are set to be paired;
20 |  |  S̄ ← S̄ ∪ {x^i};
21 |  |  F_X[i] ← 1, F_P[j] ← 1;
22 |  else
23 |  |  if x^i ≻_{p^j} x^k (the current partner of p^j) then
24 |  |  |  x^i and p^j are set to be paired;
25 |  |  |  F_X[i] ← 1, F_X[k] ← 0;
26 |  |  end
27 |  end
28 end
29 return S̄
```

We have developed two MOEA/D variants by replacing the STM model in MOEA/D-STM with the above two variants. With the same parameter settings as in Section V-D, these two variants have been experimentally compared with MOEA/D-STM on the UF test instances. The experimental results, in terms of IGD and HV metric values, are presented in Table III. To be specific, Variant-I adopts a very greedy strategy and matches each subproblem with its best solution in S. From the experimental results, we can observe that the performance of Variant-I is poorer than MOEA/D-STM on eight out of ten instances. This could be because Variant-I can match the same solution with several different subproblems and lead to the loss of population diversity. Particularly on UF10, although Variant-I outperforms MOEA/D-STM, the solutions obtained by both algorithms are still far away from the real PF as indicated by their IGD and HV values. It is evident from Table III that Variant-II cannot do better than MOEA/D-STM on any test instance. The major reason could be that subproblem agents make proposals to solution agents in the STM model and therefore, can select the best solution for each subproblem as shown in Theorem 2, while Variant-II does it in an oppositive way and is unable to keep all the elite solutions.

C. More Investigations on STM model

The experimental studies in Sections VI-A and VI-B have shown that the MOEA/D-STM is poorer than some other algorithms only on UF2 and UF10 with 300 000 function evaluations. Fig. 10(a) compares the evolution of the median IGD metric value versus the number of function evaluations of MOEA/D-STM and Variant-I on these two test instances. Clearly, the IGD value of MOEA/D-STM is still in a trend to decrease at the late search stage whereas this is not the case for Variant-I. A question naturally arises: can MOEA/D-STM perform better than other algorithms with a larger number of function evaluations on UF2 and UF10? To answer this question, we have tested MOEA/D-STM, Variant-I, and the five MOEAs introduced in Section V-C on UF2 and UF10

TABLE IV

IGD RESULTS ON UF2 AND UF10 WITH 600 000 FUNCTION EVALUATIONS

Problem	MOEA/D-STM	*Variant*-I	MOEA/D-DE	MOEA/D-DRA	NSGA-II	MSOPS-II	HypE
UF2	**1.437E-3(7.39E-4)**	1.621E-3(5.96E-4)†	4.010E-3(5.49E-4)†	3.953E-3(1.33E-3)†	2.010E-2(1.68E-3)†	1.545E-2(5.78E-3)†	3.514E-2(6.91E-4)†
UF10	**2.987E-1(2.18E-1)**	3.401E-1(1.02E-1)†	4.647E-1(5.35E-2)†	4.273E-1(8.36E-2)†	2.267E+0(1.20E-1)†	3.141E-1(1.53E-1)†	6.515E-1(3.18E-2)†

Wilcoxon's rank sum test at a 0.05 significance level is performed between MOEA/D-STM and each of the other competing algorithms. † and ‡ denote whether the performance of the corresponding algorithm is significantly worse than or better than that of MOEA/D-STM, respectively. The best mean is highlighted in boldface with gray background. UF2 has two objectives and UF10 has three objectives.

TABLE V

HV RESULTS ON UF2 AND UF10 WITH 600 000 FUNCTION EVALUATIONS

Problem	MOEA/D-STM	*Variant*-I	MOEA/D-DE	MOEA/D-DRA	NSGA-II	MSOPS-II	HypE
UF2	**3.6622(6.38E-3)**	3.6614(4.44E-3)	3.6540(9.85E-3)†	3.6536(8.59E-3)†	3.6337(2.52E-3)†	3.6010(5.58E-3)†	3.5990(9.18E-3)†
UF10	**4.8810(7.26E-1)**	4.8603(9.72E-1)†	4.2013(3.23E-1)†	4.5363(7.03E-1)†	4.582E-3(1.71E-2)†	4.6991(5.03E-1)†	3.2020(2.89E-2)†

Wilcoxon's rank sum test at a 0.05 significance level is performed between MOEA/D-STM and each of the other competing algorithms. † and ‡ denote whether the performance of the corresponding algorithm is significantly worse than or better than that of MOEA/D-STM, respectively. The best mean is highlighted in boldface with gray background. UF2 has two objectives and UF10 has three objectives.

TABLE VI

PERFORMANCE COMPARISONS OF MOEA/D-STM WITH MOEA/D-DE AND MOEA/D-DRA (nr = 1)

	IGD			HV		
Test Instance	MOEA/D-STM	MOEA/D-DE	MOEA/D-DRA	MOEA/D-STM	MOEA/D-DE	MOEA/D-DRA
UF1	**1.064E-3(6.86E-5)**	1.296E-3(9.87E-5)†	1.352E-3(9.31E-5)†	**3.6631(4.74E-4)**	3.6601(1.16E-3)	3.6593(1.17E-3)†
UF2	**2.692E-3(1.17E-3)**	6.593E-3(3.08E-3)†	3.822E-3(1.65E-3)†	**3.6575(8.44E-3)**	3.6368(1.94E-2)†	3.6503(8.15E-3)†
UF3	**6.754E-3(7.79E-3)**	8.318E-3(9.12E-3)†	7.539E-3(1.32E-2)†	**3.6537(1.31E-2)**	3.6423(1.46E-2)†	3.6455(5.34E-2)†
UF4	**5.194E-2(3.24E-3)**	5.711E-2(3.17E-3)†	5.394E-2(3.62E-3)†	**3.1815(1.40E-2)**	3.1677(1.37E-2)†	3.1722(1.39E-2)†
UF5	**2.471E-1(3.17E-2)**	3.568E-1(6.25E-2)†	3.129E-1(5.64E-2)†	**2.9426(8.94E-2)**	2.5610(1.86E-1)†	2.7027(1.59E-1)†
UF6	7.031E-2(2.72E-2)	**6.123E-2(6.66E-3)**‡	6.969E-2(2.89E-2)	3.2072(5.36E-2)	**3.2135(2.89E-2)**‡	3.1966(6.34E-2)
UF7	**1.114E-3(1.11E-4)**	1.533E-3(2.68E-4)†	1.204E-3(8.50E-5)	**3.4968(5.97E-4)**	3.4901(6.01E-3)	3.4962(8.04E-4)
UF8	**2.250E-2(1.46E-3)**	5.528E-2(1.03E-2)†	3.342E-2(7.64E-3)†	**7.4241(2.91E-3)**	7.3423(2.11E-2)†	7.3995(1.80E-2)†
UF9	**2.100E-2(8.45E-4)**	4.657E-2(2.97E-2)†	4.237E-2(3.89E-2)†	**7.7541(3.64E-3)**	7.5938(1.41E-1)†	7.6528(1.77E-1)†
UF10	8.054E-1(1.76E-1)	8.836E-1(1.31E-1)†	**6.526E-1(7.23E-2)**‡	2.5199(6.15E-1)	2.2289(3.84E-1)†	**3.0174(2.90E-1)**‡

Wilcoxon's rank sum test at a 0.05 significance level is performed between MOEA/D-STM and MOEA/D-DE and MOEA/D-DRA. † and ‡ denote whether the performance of the corresponding algorithm is significantly worse than or better than that of MOEA/D-STM, respectively. The best mean is highlighted in boldface with gray background. UF1 to UF7 have two objectives and UF8 to UF10 have three objectives.

with 600 000 function evaluations. Tables IV and V present the experimental results. Clearly, MOEA/D-STM beats all other algorithms. Fig. 10(b) plots the evolution of the median IGD metric value versus the number of function evaluations of MOEA/D-STM and Variant-I on UF2 and UF10. These figures indicate that MOEA/D-STM outperforms Variant-I after around 480 000 function evaluations.

MOEA/D-DE and MOEA/D-DRA use a steady-state manner to update their populations and allow one new solution to replace more than one old solution, which could decrease their population diversity. In contrast, MOEA/D-STM pairs each subproblem with one single solution and thus different subproblems have different solutions. Another issue is whether we can improve the performance of MOEA/D-DE and MOEA/D-DRA by only allowing at most one old solution to be replaced by each offspring (i.e., set the parameter $nr = 1$ in MOEA/D-DE and MOEA/D-DRA). We have tested MOEA/D-DE and MOEA/D-DRA with $nr = 1$ (the other parameters are set the same as in Section V-D) and compared both of them with MOEA/D-STM. From the experimental results shown in Table VI, one can observe that the performance of MOEA/D-DE and MOEA/D-DRA can be improved by setting $nr = 1$ on UF1, UF3, UF6, and UF8. Nevertheless, MOEA/D-STM is still the best on most test instances. Therefore, it can be concluded that the good performance achieved by MOEA/D-STM does come from its stability in matching, which well maintains the population diversity.

TABLE VII
SETTINGS OF REFERENCE POINTS

Problem	Reference Points
ZDT1 and ZDT4	$(0.3, 0.4)^T$, $(0.65, 0.3)^T$
ZDT2	$(0.2, 0.8)^T$, $(0.9, 0.4)^T$
ZDT3	$(0.15, 0.4)^T$, $(0.4, 0.0)^T$
ZDT6	$(0.9, 0.3)^T$, $(0.5, 0.7)^T$
DTLZ1	$(0.05, 0.05, 0.2)^T$, $(0.3, 0.3, 0.2)^T$
DTLZ2 to DTLZ 4	$(0.2, 0.5, 0.6)^T$, $(0.7, 0.8, 0.5)^T$
DTLZ5 to DTLZ 6	$(0.1, 0.3, 0.5)^T$, $(0.6, 0.7, 0.5)^T$
DTLZ7	$(0.165, 0.71, 4.678)^T$, $(0.75, 0.15, 6.0)^T$, $(0.1, 0.1, 4.0)^T$

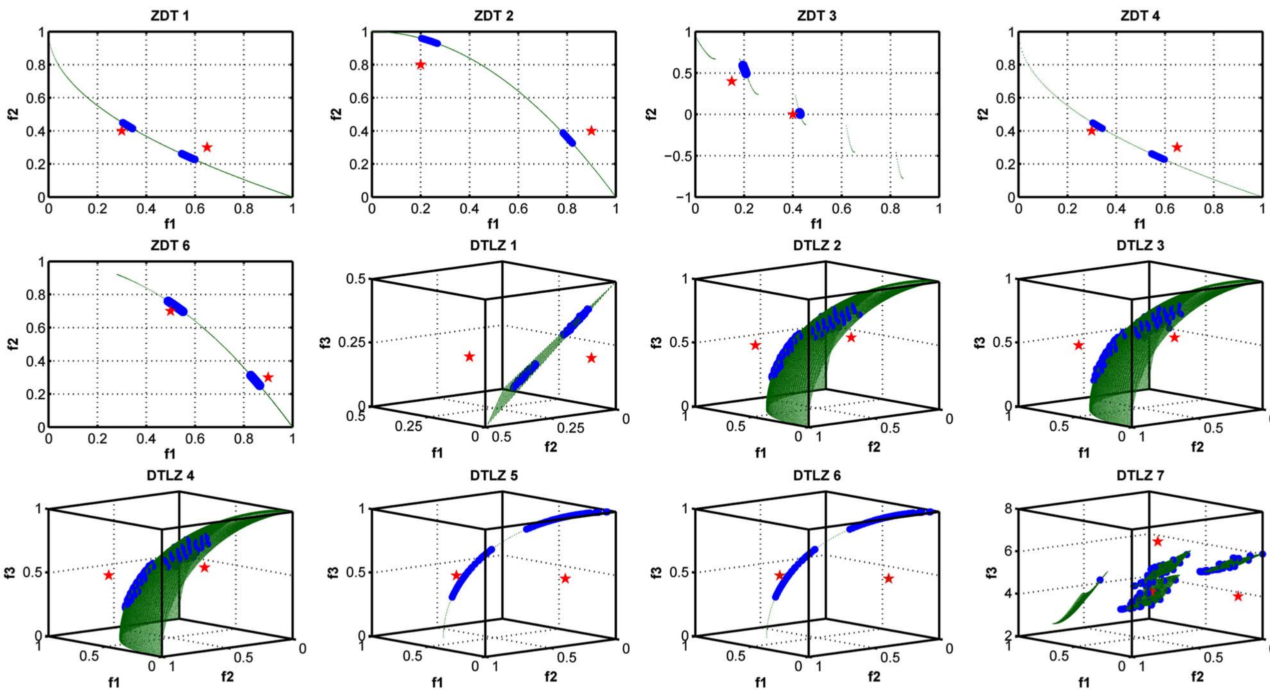


Fig. 11. Nondominated solutions in the preferred regions found by r-MOEA/D-STM.

D. Preference Incorporation by STM Model

A decision maker (DM) may have some preferences on solutions. It is widely accepted that using DM preference information in MOEAs can potentially reduce the computational overheads and drive the search toward a particular area that is desirable to the DM. A DM may provide a reference point in the objective space as her preference information [37]. In the following, we make a very preliminary study to investigate whether the reference point information can make MOEA/D-STM approximate the regions that the DMs are interested in.

Let $\mathbf{r}$ be a reference point in the objective space provided by the DM. Then, we use the method described in Section IV-A to generate K weight vectors. Among them, N weight vectors closest to $\mathbf{r}$ are chosen to define N subproblems for MOEA/D-STM. We call this algorithm r-MOEA/D-STM. Experimental studies have been conducted on a set of benchmark problems including the bi-objective ZDT [38] and the three-objective DTLZ [39] test instances with different characteristics such as multimodality (e.g., ZDT4, DTLZ1, and DTLZ3), discontinuous PF (e.g., ZDT3 and DTLZ7), degeneration (e.g., DTLZ5 and DLTZ6), and bias mapping (e.g., ZDT6, DTLZ4 and DTLZ6). $K = 1\,000$ and $N = 100$ are set in our experiments. The total number of function evaluations is 20 000 for the bi-objective test instances, and 30 000 for the three-objective ones except DTLZ3, where 100 000 function evaluations are used due to its difficulties. The other parameters of r-MOEA/D-STM are the same as in Section V-D except $CR = F = 0.5$.

The reference points used in our experiments, which include both feasible and infeasible ones, are provided in Table VII. From the experimental results shown in Fig. 11, it can be concluded that r-MOEA/D-STM is able to obtain a good distribution of solutions near the provided reference points for all test instances. It indicates that the reference point information can be very useful for guiding MOEA/D-STM.

VII. Conclusion

MOEA/D decomposes a MOP into a number of single objective optimization subproblems and optimizes them in a collaborative manner. In MOEA/D, each subproblem is paired with a solution in the current population. Stable matching, proposed by economists, has solid theoretical foundations and has been used in various fields. This paper treats subproblems and solutions in MOEA/D as two different sets of agents. Therefore, selection in MOEA/D can be naturally modeled as a matching problem. In our approach, a subproblem prefers solutions that can lower its aggregation function value, and a solution prefers subproblems whose direction vectors are close to it. Therefore, the subproblem preference encourages convergence whereas the solution preference promotes population diversity. We have proposed using the STM model to balance these two preferences and, thus, the convergence and diversity of the evolutionary search. Extensive experimental studies have been conducted to compare our proposed MOEA/D-STM with other MOEAs, and to investigate the ability and behavior of MOEA/D-STM.

This paper presents a first attempt at using matching theory and techniques to enhance evolutionary algorithms. In the following, we list several possible research issues along this line.

1) We believe that some other issues in evolutionary computation can also be modeled and tackled as matching problems. For example, mating selection can also be regarded as a matching problem. Specifically, unlike unisexual populations used in classical EAs, solutions can be assigned with different genders. Therefore, mating selection becomes a sexual selection [40] that matches each male solution with its ideal spouse.
2) In the STM model used in this paper, the resulting stable matching always favors the agents who make proposals. It is desirable in many cases to find a "fair" stable matching that does not favor either side. Some efforts have been made in economics on this direction, such as egalitarian stable matching [41], which minimizes the total rank sum of the outcome in the marriage model. It should be very interesting to study the use of these models in evolutionary computation.

References

[1] A. A. Montaño, C. A. C. Coello, and E. Mezura-Montes, "Multiobjective evolutionary algorithms in aeronautical and aerospace engineering," *IEEE Trans. Evol. Comput.*, vol. 16, no. 5, pp. 662–694, Aug. 2012.

[2] M. G. C. Tapia and C. A. C. Coello, "Applications of multi-objective evolutionary algorithms in economics and finance: A survey," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 532–539.

[3] H. R. Cheshmehgaz, M. Desa, and A. Wibowo, "A flexible three-level logistic network design considering cost and time criteria with a multi-objective evolutionary algorithm," *J. Intell. Manuf.*, vol. 24, no. 2, pp. 1–17, 2011.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[5] K. Li, S. Kwong, J. Cao, M. Li, J. Zheng, and R. Shen, "Achieving balance between proximity and diversity in multi-objective evolutionary algorithm," *Inf. Sci.*, vol. 182, no. 1, pp. 220–242, 2012.

[6] K. Li, S. Kwong, R. Wang, K.-S. Tang, and K.-F. Man, "Learning paradigm based on jumping genes: A general framework for enhancing exploration in evolutionary multiobjective optimization," *Inf. Sci.*, vol. 226, pp. 1–22, Mar. 2013.

[7] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proc. PPSN*, 2004, pp. 832–842.

[8] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1653–1669, 2007.

[9] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.

[10] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[11] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Trans. Syst., Man, Cybern. C, Applicat. Rev.*, vol. 28, no. 3, pp. 392–403, Aug. 1998.

[12] E. J. Hughes, "Multiple single objective Pareto sampling," in *Proc. IEEE Congr. Evol. Comput.*, Dec. 2003, pp. 2678–2684.

[13] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.

[14] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *Proc. IEEE Congr. Evol. Comput.*, May 2009, pp. 203–208.

[15] K. Li, Á. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 114–130, Feb. 2014.

[16] D. Gusfield and R. W. Irving, *The Stable Marriage Problem: Structure and Algorithms*, Cambridge, MA, USA: MIT Press, 1989.

[17] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *Amer. Math. Mon.*, vol. 69, no. 1, pp. 9–15, 1962.

[18] A. E. Roth, "Deferred acceptance algorithms: History, theory, practice and open questions," *Int. J. Game Theory*, vol. 36, pp. 537–569, Jul. 2008.

[19] H. Xu and B. Li, "Anchor: A versatile and efficient framework for resource management in the cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1066–1076, Jun. 2013.

[20] C. T. S. Sng and D. Manlove, "Popular matchings in the weighted capacitated house allocation problem," *J. Discrete Algorithms*, vol. 8, no. 2, pp. 102–116, Jun. 2010.

[21] K. Miettinen, *Nonlinear Multiobjective Optimization*, vol. 12. Norwell, MA, USA: Kluwer, 1999.

[22] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-dominated Sorting Approach, Part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, 2013, accepted for publication.

[23] S. Jiang, Z. Cai, J. Zhang, and Y.-S. Ong, "Multiobjective optimization by decomposition with Pareto-adaptive weight vectors," in *Proc. ICNC*, 2011, pp. 1260–1264.

[24] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM J. Optimiz.*, vol. 8, no. 3, pp. 631–657, 1998.

[25] G. Strang, *Introduction to Linear Algebra*, 3rd ed. Cambridge, MA, USA: Wellesley-Cambridge Press, 2003, pp. 222–225.

[26] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," *Univ. Essex and Nanyang Technol. Univ.*, Essex, U.K./Singapore, Tech. Rep. CES-487, 2008.

[27] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[28] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Comput. Sci. Inf.*, vol. 26, no. 4, pp. 30–45, 1996.

[29] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

[30] O. Schütze, X. Esquivel, A. Lara, and C. A. C. Coello, "Using the averaged Hausdorff distance as a performance measure in evolutionary multi-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 4, pp. 504–522, Aug. 2012.

[31] K. Li, Á. Fialho, and S. Kwong, "Multi-objective differential evolution with adaptive control of parameters and operators," in *Proc. LION*, Jan. 2011, pp. 473–487.

[32] K. Li, S. Kwong, R. Wang, J. Cao, and I. J. Rudas, "Multi-objective differential evolution with self-navigation," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2012, pp. 508–513.

[33] P. Bosman and D. Thierens, “The balance between proximity and diversity in multiobjective evolutionary algorithms,” *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 174–188, Apr. 2003.
[34] E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach,” *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.
[35] E. J. Hughes, “MSOPS-II: A general-purpose many-objective optimiser,” in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 3944–3951.
[36] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: Wiley, 2001.
[37] A. Mohammadi, M. N. Omidvar, and X. Li, “Reference point based multi-objective optimization through decomposition,” in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2012, pp. 1150–1157.
[38] E. Zitzler, K. Deb, and L. Thiele, “Comparison of multiobjective evolutionary algorithms: Empirical results,” *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.
[39] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable test problems for evolutionary multiobjective optimization,” in *Evolutionary Multiobjective Optimization (Advanced Information and Knowledge Processing)*, A. Abraham, L. Jain, and R. Goldberg, Eds. London, U.K.: Springer, 2005, pp. 105–145.
[40] K. S. Goh, A. Lim, and B. Rodrigues, “Sexual selection for genetic algorithms,” *Artif. Intell. Rev.*, vol. 19, no. 2, pp. 123–152, Apr. 2003.
[41] D. F. Manlove, R. W. Irving, K. Iwama, S. Miyazaki, and Y. Morita, “Hard variants of stable marriage,” *Theoretical Comput. Sci.*, vol. 276, pp. 261–279, 2002.

Ke Li (S’10) was born in Hunan, China, in 1985. He received the B.Sc. and M.Sc. degrees in computer science and technology from the Xiangtan University, Hunan, China, in 2007 and 2010, respectively, and the Ph.D. in computer science from City University of Hong Kong, Kowloon, Hong Kong, in 2014.

His research interests include evolutionary multi-objective optimization, surrogate-assisted evolutionary algorithms, and statistical machine learning techniques.

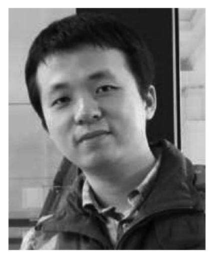
Qingfu Zhang (M’01-SM’06) received the B.Sc. in mathematics from Shanxi University, Shanxi, China, in 1984, the M.Sc. in applied mathematics, and the Ph.D. in information engineering from Xidian University, Xidian, China, in 1991 and 1994, respectively.

He is currently a Professor with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong, and a Professor on leave from the School of Computer Science and Electronic Engineering, University of Essex, Essex, U.K. He holds two patents and is the author of many research publications. His research interests include evolutionary computation, optimization, neural networks, data analysis, and their applications.

Dr. Zhang is an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the IEEE TRANSACTIONS ON CYBERNETICS. He is also an Editorial Board Member of three other international journals. MOEA/D, a multiobjective optimization algorithm developed in his group, won the Unconstrained Multiobjective Optimization Algorithm Competition at the Congress of Evolutionary Computation 2009, and was awarded the 2010 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award.

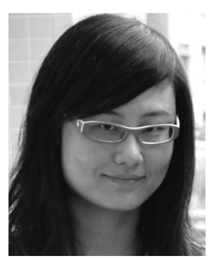
Sam Kwong (M’93–SM’04–F’14) received the B.S. and M.S. degrees in electrical engineering from State University of New York at Buffalo, in 1983, University of Waterloo, Waterloo, ON, Canada, in 1985, and the Ph.D. degree from the University of Hagen, Hagen, Germany, in 1996. From 1985 to 1987, he was a Diagnostic Engineer with Control Data Canada. He joined Bell Northern Research Canada as a Member of Scientific Staff. In 1990, he became a Lecturer in the Department of Electronic Engineering, City University of Hong Kong, where he is currently a Professor and Head in the Department of Computer Science. His main research interests include evolutionary computation and video coding.

Dr. Kwong is an Associate Editor of the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and the *Information Sciences Journal*.



Miqing Li received the B.Sc. degree in computer science from the School of Computer and Communication, Hunan University, Hunan, China, in 2004, and the M.Sc. degree in computer science from the College of Information Engineering, Xiangtan University, Hunan, China, in 2008, respectively. He is currently working toward the Ph.D. degree with the School of Information Systems, Computing, and Mathematics, Brunel University, Essex, U.K.

He has published over 20 research papers since 2008. His research interests include evolutionary computation, multiobjective optimization, and dynamic optimization.



Ran Wang (S’09) received the B.S. degree from the College of Information Science and Technology, Beijing Forestry University, Beijing, China, in 2009, and the Ph.D. degree from the City University of Hong Kong, Kowloon, Hong Kong, in 2013.

She is currently a Research Associate with the Department of Computer Science, City University of Hong Kong. Her research interests include support vector machines, extreme learning machines, decision tree induction, active learning, multiclass classification, and the related applications of machine learning.