Two-Level Stable Matching-Based Selection in MOEA/D

Mengyuan Wu*, Sam Kwong*,§ Qingfu Zhang*, Ke Li†, Ran Wang*,¶ and Bo Liu‡
*Department of Computer Science, City University of Hong Kong, Hong Kong SAR
†Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48864, USA
‡Department of Computing, Glyndwr University, Wrexham, LL11 2AW, U.K.
§City University of Hong Kong Shenzhen Research Institute, Shenzhen 518057, China
¶Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China

***Abstract*—Stable matching-based selection models the selection process in MOEA/D as a stable marriage problem. By finding a stable matching between the subproblems and solutions, the solutions are assigned to subproblems to balance the convergence and the diversity. In this paper, a two-level stable matching-based selection is proposed to further guarantee the diversity of the population. More specifically, the first level of stable matching only matches a solution to one of its most preferred subproblems and the second level of stable matching is responsible for matching the solutions to the remaining subproblems. Experimental studies demonstrate that the proposed selection scheme is effective and competitive comparing to other state-of-the-art selection schemes for MOEA/D.**

I. Introduction

A multiobjective optimization problem (MOP) can be formulated as follows:

$$\begin{array}{ll} \text{minimize} & F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_m(\mathbf{x}))^{\mathrm{T}} \\ \text{subject to} & \mathbf{x} \in \Omega \end{array} \tag{1}$$

where $\mathbf{x} = (x_1, x_2, ..., x_n)^{\mathrm{T}}$ is the decision variable, $\Omega \in \mathcal{R}^n$ is the search space and $F : \mathcal{R}^n \rightarrow \mathcal{R}^m$ is the objective vector containing conflicting objectives to be optimized. Since an MOP has multiple objectives to be optimized, no solution can optimizes all the objectives at the same time. Pareto optimality is used to define the optimal solutions of an MOP.

- Solution $\mathbf{x}^1$ is said to **dominate** solution $\mathbf{x}^2$ if and only if $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$ for all $i \in \{1, 2, ..., m\}$ and $f_j(\mathbf{x}^1) < f_j(\mathbf{x}^2)$ for at least one $j \in \{1, 2, ..., m\}$.
- A solution $\mathbf{x}^*$ is said to be **Pareto optimal** if and only if there is no other solution in the search space that dominates it. The set of all Pareto optimal solutions is called the **Pareto optimal set** (PS).
- The **Pareto front** (PF) is the set of corresponding objectives vectors of all solutions in the PS.

Multiobjective evolutionary algorithms (MOEAs), which are capable of finding a set of solutions to estimate the PF of an MOP, have attracted much attention from researchers on computational intelligence. Diversity and convergence are two important factors affecting the performance of an MOEA. During the evolutionary process, diversity is needed to escape local optimum and maintain a set of diverse solutions, while convergence is required to approach the PF. The selection process is a key step to control the diversity and convergence of an MOEA [1], [2]. According to the selection scheme, MOEAs can be categorized into three groups. Pareto-based MOEAs use the domination relation to select offspring solutions, which contributes to the convergence. Diversity are maintained according to density estimation such as crowding distance used in NSGA-II [3] and clustering analysis in SPEA2 [4], respectively. Indicator-based MOEAs apply performance indicators [5], e.g., hypervolume [6] which can measure the convergence and diversity at the same time, to guide the selection. Decomposition-based MOEAs decompose the MOP into a number of single objective optimization problems and optimize them simultaneously. As a representative of this sort, MOEA based on decomposition (MOEA/D) [7] uses the elite-based updating scheme and a set of evenly distributed weight vectors to balance the convergence and the diversity.

In the original MOEA/D, it employs a steady-state evolution model, where the update procedure takes place right after the generation of an offspring. As discussed in [8], this selection mechanism has some side effects on maintaining diversity, especially when tackling problems with complicated properties. In [8], Li. et al modeled the selection process in MOEA/D as a matching process, and employ the classic stable marriage model [9] to design the selection operator therein. More specifically, it defines the subproblems and solutions as two sets of agents with mutual preferences on convergence and diversity respectively. Thereby, the stable matching between subproblems and solutions strikes the balance between convergence and diversity of the selection process. However, this method gives more emphasis on the convergence side. Later, Li *et al* [10] further improve the preference definition between subproblems and solutions by including the niche count as a component and develop a straightforward but more effective selection scheme based on the interrelationship between subproblems and solutions. In this paper, we develop a decomposition-based MOEA with two-level stable matching based-selection (MOEA/D-STM2L) to further guarantee the diversity. More specifically, in the first-level stable matching,

978-1-4799-8697-2/15 $31.00 © 2015 IEEE
DOI 10.1109/SMC.2015.302

we restrict the number of subproblems to which a solution is allowed to be matched. Therefore, a solution can only be assigned to one of its most preferred subproblems. In the second-level stable matching, the remaining unmatched subproblems are assigned with appropriate solutions. The performance of our proposed algorithm is validated on 17 unconstrained benchmark problems, comparing with other three state-of-the-art MOEA/D variants.

The remainder of this paper is organized as follows. Section II introduces the background knowledge. Section III describes two-level stable matching-based selection. The experimental settings are listed in Section IV and corresponding results are analyzed in Section V. Section VI concludes the paper.

II. Background

A. MOEA/D framework

MOEA/D decomposes the MOP into a set of N subproblems using a set of evenly distributed weight vectors in the objective space. The optimal solution of each subproblem is a Pareto optimal solution of the original MOP. There are several decomposition approaches that can be used in MOEA/D, such as the weighted sum (WS), Tchebycheff (TCH) and boundary intersection (BI) approaches [11]. As discussed in [12], under some mild condition, TCH approach has similar effects as Pareto dominance in selection. In this case, this paper chooses TCH approach for decomposition. The aggregation function of the TCH approach is defined as

$$\begin{aligned} &\text{minimize} \quad g(\mathbf{x}|\lambda, \mathbf{z}^*) = \max_{1\le i\le m} \{|f_i(\mathbf{x}) - \mathbf{z}_i^*|/\lambda_i\} \\ &\text{subject to} \quad \mathbf{x} \in \Omega \end{aligned} \tag{2}$$

where $\lambda = (\lambda_1, \lambda_2, ..., \lambda_m)^{\mathrm{T}}$ is a weight vector with $\lambda_i \geq 0$ for all $i \in \{1, 2, ..., m\}$ and $\sum_{i=1}^{m} \lambda_i = 1$. $\mathbf{z}^* = (z_1^*, z_2^*, ..., z_m^*)^{\mathrm{T}}$, where $z_i^* = \min_{\mathbf{x}\in\Omega} f_i(\mathbf{x})$ for each $i \in \{1, 2, ..., m\}$ is the ideal objective vector. In particular, λ_i is set to be 10^{-6} when $\lambda_i = 0$.

An important feature of MOEA/D is the use of neighboring subproblems as parents to generate offspring solutions. A set of neighboring subproblems are defined for each subproblem based on the distance between their weight vectors. Since neighboring subproblems have similar objective vectors, a promising child solution is likely to be generated using information from its neighboring subproblems. Therefore, the subproblems can be optimized in a collaborative manner.

MOEA/D-DRA [13] is an MOEA/D variant and adopts the dynamic resource allocation. A utility function π^j is defined for each subproblem p^j $(j \in \{1, 2, ..., N\})$ to estimate the possible improvement of the aggregation function value of that subproblem. The formula of π^j is defined as

$$\pi^j = \begin{cases} 1 & \text{if}\Delta^{\mathrm{j}} > 0.001 \\ (0.95 + 0.05 \times \frac{\Delta^j}{0.001}) \times \pi^j & \text{otherwise} \end{cases} \tag{3}$$

where Δ^j is the relative decrease in the aggregation function value of subproblem p^j, which can be calculated as

$$\Delta^j = \frac{g(\mathbf{x}^{old}|\lambda^j, \mathbf{z}^*) - g(\mathbf{x}^{new}|\lambda^j, \mathbf{z}^*)}{g(\mathbf{x}^{old}|\lambda^j, \mathbf{z}^*)} \tag{4}$$

where $\mathbf{x}^{new}$ and $\mathbf{x}^{old}$ are current and previous solutions of p^j.

The MOEA/D framework with dynamic resource allocation used in this paper is presented in Algorithm 1. More details of the base algorithm can be found in [13].

Algorithm 1: MOEA/D-DRA framework

Generate N evenly spread weight vectors $\{\lambda^1, \lambda^2, ..., \lambda^N\}$;
Initialize population $P \leftarrow \{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^N\}$ randomly in the search space, evaluate their objective vectors and randomly assign them to N subproblems;
for *each subproblem* p^j *with weight vector* λ^j **do**
    Find its T closest weight vectors excluding λ^j to form set $B(j)$;
    $\pi^j \leftarrow 1$;
end
Estimate $\mathbf{z}^* = (z_1^*, z_2^*, ..., z_m^*)^{\mathrm{T}}$ by $z_i^* \leftarrow \min_{\mathbf{x}\in P} f_i(\mathbf{x})$;
$iteration \leftarrow 0$;
while *Stopping criterion is not satisfied* **do**
    $Q \leftarrow \emptyset$;
    Select $N/5$ subproblems based on π^j by using 10-tournament selection to form set I;
    for *each* $k \in I$ **do**
        Uniformly randomly generate a number $rand \in (1, 0)$;
        if $rand < \delta$ **then**
            Randomly select two solutions $\mathbf{x}^{r1}$ and $\mathbf{x}^{r2}$ from $B(k)$;
        else
            Randomly select two solutions $\mathbf{x}^{r1}$ and $\mathbf{x}^{r2}$ from the whole population P;
        end
        Use DE operator to generate a new solution $\mathbf{x}^{k,new} = \mathbf{x}^k + F(\mathbf{x}^{r1} - \mathbf{x}^{r2})$;
        Do polynomial mutation to $\mathbf{x}^{k,new}$ and fix it if it is out of the search space;
        $Q \leftarrow Q \bigcup x^{k,new}$;
    end
    Update $\mathbf{z}^*$;
    $P \leftarrow P \bigcup Q$;
    $iteration \leftarrow iteration + 1$;
    if *mod(iteration,30)=0* **then**
        Update π^j for all subproblems;
    end
end

B. Stable matching-based selction

How to select promising offspring solutions is one of the most important factors in MOEA/D. In original MOEA/D, the selection process is simply based on the aggregation function values. Once a new solution is generated, it is compared with the current solution of the neighboring subproblems. If the new solution is more promising, it will update at least one of them. As this selection scheme focuses too much on selecting the elite solutions, it may lose the diversity for some MOPs with complicated properties.

Stable marriage problem (SMP) is to find one-to-one stable matching between a set of men and a set of women, where each man has an ordered preference list of all women and vice versa. In terms of matching pairs, it is obviously not desirable if a man and a woman are matched to each other but both of them prefer other partners. Such matching pair is called an unstable pair. A stable matching solution is defined as a matching solution that does not have any unstable pair [9].

Different from the original MOEA/D, the selection operator proposed in [8] uses a generational evolution model, where the selection of the next parents takes place after the generation of a population of offspring. MOEA/D-STM [8] treats the subproblems and solutions as two sets of agents, and defines their mutual preferences by considering convergence and diversity respectively. Thereafter, the selection process is modeled as a matching process between subproblems and solutions. The stable matching between subproblems and solutions achieves an equilibrium between their mutual preferences, thus striking the balance between convergence and diversity of the selection process. More specifically, the preferences between subproblems and solutions are formulated as follows:

- In order to guarantee the convergence of the algorithm, a subproblem prefers solutions with lower aggregation function values. The preference value of subproblem p on solutions $\mathbf{x}$ is defined as the aggregation function value of $\mathbf{x}$ on p:
$$\Delta P(p, \mathbf{x}) = g(\mathbf{x}|\lambda, \mathbf{z}^*) \tag{5}$$
where λ is the weight vector of p.
- On the other hand, a solution prefers subproblems that are closer to its objective vector. In such a way, each subproblem is matched to a solution closer to its weight vector in the objective space, resulting in a good diversity of the solutions. The preference value of $\mathbf{x}$ on p is defined as:
$$\Delta P(\mathbf{x}, p) = \overline{F}(\mathbf{x}) - \frac{\lambda^{\mathrm{T}}\overline{F}(\mathbf{x})}{\lambda^{\mathrm{T}}\lambda}\lambda \tag{6}$$
where $\overline{F}(\mathbf{x}) = F(\mathbf{x}) - \mathbf{z}^*$ is the relative objective vector of $\mathbf{x}$. In [8], $\overline{F}(\mathbf{x})$ is defined as the normalized objective vector of $\mathbf{x}$. But the objective vector used in the aggregation function is not normalized. It is more reasonable to keep it coincident.

Given the preferences from both sides, the preference lists are calculated for each subproblem and each solution by sorting the preference values in ascending order. Since the number of solutions is larger than the number of subproblems, some solutions will not be matched to any subproblem. The stable matching solution is described as a matching solution where

- if a solution $\mathbf{x}$ is matched to a subproblem but not p, then $\mathbf{x}$ prefers its current partner to p and p prefers it current partner to $\mathbf{x}$;
- if a solution $\mathbf{x}$ is not matched to any subproblem, no subproblem prefers $\mathbf{x}$ to its current partner.

Algorithm 2 presents the pseudo-code for finding the stable matching between subproblems and solutions [8].

Algorithm 2: Stable matching

Include all the subproblems into unmatched set $\mathbb{P}$;
Include all the solutions into unmatched set $\mathbb{S}$;
while $\mathbb{P} \neq \emptyset$ **do**
  Select a subproblem from $\mathbb{P}$, denoted as p, and propose a matching request to the first solution $\mathbf{x}$ on p's preference list, to which p has not yet proposed;
  if $\mathbf{x} \in \mathbb{S}$ **then**
    Match $\mathbf{x}$ to p;
    Set $\mathbb{P} \leftarrow \mathbb{P} \setminus p$ and $\mathbb{S} \leftarrow \mathbb{S} \setminus \mathbf{x}$;
  else
    if $\mathbf{x}$ *prefers* p *to its current partner* $p\prime$ **then**
      Match $\mathbf{x}$ to p;
      Set $\mathbb{P} \leftarrow \mathbb{P} \cup p\prime \setminus p$;
    end
  end
end
Output the matching solution.

III. Two-Level Stable Matching-Based Selection

The stable matching-based selection finds a stable matching between subproblems and solutions by striking a balance between their preferences. Nevertheless, the balance between the preferences of subproblems and solutions does not mean a real balance between convergence and diversity. The definition of the preferences presented in Section II requires that more importance should be attached to match the solution to a subproblem that ranks higher in its preference list. Therefore, the stable matching-based selection reaches a balance closer to the side of convergence.

A straightforward way to overcome the lack of diversity is to restrict the number of preferred subproblems that a solution is allowed to be matched to. In other words, the preference list of a solution is reduced to a smaller size r, so that only its r most preferred subproblems remain. Then, this SMP is referred as an SMP with incomplete lists [14]. The modified stable matching algorithm is described in Algorithm 3. Notice that the only difference compared to Algorithm 2 is the condition of the while-loop. If the subproblem is still being unmatched after it has proposed matching requests to all solutions on its list, it means that the subproblem will not end up with any stable matching solution and it will be left being unmatched.

Therefore, the problem left by the stable matching with incomplete lists is that not all subproblems will be assigned a solution. The two-level stable matching-based selection is proposed to solve this problem. In the selection process, two levels of stable matching are conducted to select solutions. In the first-level stable matching, in order to strengthen the diversity of the solutions, the stable matching algorithm with solutions' incomplete preference lists is firstly applied to match

Algorithm 3: Stable matching with incomplete lists

Include all the subproblems into unmatched set $\mathbb{P}$;
Include all the solutions into unmatched set $\mathbb{S}$;
while $\mathbb{P} \neq \emptyset$ **do**
  Select a subproblem from $\mathbb{P}$, denoted as p, and propose a matching request to the first solution $\mathbf{x}$ on p's preference list, to which p has not yet proposed;
  if *p is on the preference list of* $\mathbf{x}$ **then**
    if $\mathbf{x} \in \mathbb{S}$ **then**
      Match $\mathbf{x}$ to p;
      Set $\mathbb{P} \leftarrow \mathbb{P} \setminus p$ and $\mathbb{S} \leftarrow \mathbb{S} \setminus \mathbf{x}$;
    else
      if $\mathbf{x}$ *prefers* p *to its current partner* $p\prime$ **then**
        Match $\mathbf{x}$ to p;
        Set $\mathbb{P} \leftarrow \mathbb{P} \cup p\prime \setminus p$;
      end
    end
  end
  if $\mathbf{x}$ *is the last solution on the preference list of* p **and** $p \in \mathbb{P}$ **then**
    $\mathbb{P} \leftarrow \mathbb{P} \setminus p$;
  end
end
Output the matching solution.

Algorithm 4: Two-level stable matching-based selection

Calculate the preference values of all subproblems and solutions;
First-level of stable matching:
Generate the full preference lists of all subproblems and solutions by sorting their preference values in ascending order;
Reduce the size of the solutions' preference lists by keeping the first r elements;
Find a stable matching between the subproblems and solutions by **Algorithm 3**;
Second-level of stable matching:
Generate the full preference lists of all the remaining subproblems and solutions by sorting their preference values in ascending order;
Find a stable matching between the remaining subproblems and solutions by **Algorithm 2**;
Output the combined matching solution.

the solutions to its closest subproblems. After the first-level stable matching, the subproblems remaining not matched are most likely to be far away from any solution. Thus, the main purpose of the second-level stable matching is to select a solution for the remaining subproblems. Whether the solution matched to a subproblem is closer to it or not is not as important as in the first-level stable matching. Therefore, in the second-level stable matching, the stable matching algorithm with full preference lists are applied to find a stable matching between all remaining subproblems and solutions. The algorithmic structure of the two-level stable matching-based selection is demonstrated in Algorithm 4.

IV. Experimental Settings

The two-level stable matching-based selection presented in Section III is integrated into the MOEA/D-DRA framework presented in Section II to form the new algorithm MOEA/D-STM2L. In order to evaluate the performance of MOEA/D-STM2L, it is compared with its base algorithm MOEA/D-DRA, its predecessor MOEA/D-STM and MOEA/D-IR [10] with the inter-relationship based selection. All algorithms are implemented in MATLAB. This section introduces the experimental settings for the experimental studies.

A. Test instances

17 benchmark unconstrained test MOPs are chosen for experimental studies. They are UF1-UF10 from CEC2009 MOEA competition [15] and MOP1-MOP7 [16]. The number of the decision variables for the UF test instances is set to be 30. For MOP1-MOP7, it is set to be 10.

B. Performance Metric

The Inverted Generational Distance (IGD) metric [17] is used to comprehensively investigate the performance of the proposed algorithm. It is defined as the average distance of a set of points uniformly sampled on the PF to their nearest objective vectors of the solution set found by the MOEA. It is formulated as:

$$IGD(P, P^*) = \frac{\sum_{\mathbf{x} \in P^*} dist(\mathbf{x}, P)}{|P^*|} \tag{7}$$

where $dist(\mathbf{x}, P)$ is the Euclidian distance of the objective vector to its nearest neighbor. IGD metric is a commonly used indication of the convergence and diversity of the solutions. The lower is the IGD value, the better is the approximation of the whole PF. The uniformly sampled objective vectors on the PF are provided from the inventors of these test instances. In addition, Wilcoxon's rank sum test at a 5% significance level is conducted between the metric values of MOEA/D-STM2L and its comparing MOEAs.

C. General Parameter Settings

Since the four variants of MOEA/D share most of the parameters in common, their parameters settings in this experimental studies are tried to be kept in consistent. Details of the same parameters settings are summarized as follows:

- N is set to be 600 for UF1-UF7, $1,000$ for UF8-UF10, 100 for MOP1-MOP5 and 300 for MOP6 and MOP7.
- For the DE operator, $CR = 1$ and $F = 0.5$. For the polynomial mutation, the mutation probability is set as $1/n$ and its distribution index $\mu = 20$ [18].
- δ is set to be 0.5.
- The maximal number of function evaluations is set to be $300,000$ for all test instances. Each algorithm is run 20 times on each test instance.

The settings of other parameters are listed below:

- The neighborhood size $T = 20$ for MOEA/D-STM, MOEAD-STM2L and MOEA/D-IR according to [8]. MOEA/D-DRA uses the settings suggested by its original authors in [13], i.e. $T = 0.1 \times N$.
- For MOEA/D-STM2L, r is set to be 8 for UF test instances and 4 for MOP test instances.
- For MOEA/D-IR, the number of related subproblems for a solution is set to be 2 and the number of related solutions for a subproblem is set to be 8.

V. Experimental studies

A. Performance Comparisons with other MOEAs

The mean and standard variance of the IGD results of MOEA/D-STM2L together with three comparing MOEAs are shown in Table I.

In terms of the UF test instances, the mean IGD value of MOEA/D-STM2L is better for most of the test instances comparing to each of the other variants of MOEA/D, including its predecessor MOEA/D-STM. Especially for UF6 and UF9, the performance of MOEA/D-STM2L is significantly better than MOEA/D-DRA and MOEA/D-IR. Even for test instances where MOEA/D-STM2L does not perform the best, its difference to the best mean value is negligible. The effective performance of MOEA/D-STM2L is mainly because of the farer balance between the convergence and diversity of the population. The only exception is UF10, where the stable matching-based selection is not effective anymore compared to MOEA/D-DRA and MOEA/D-IR. The reason might be that the trade-off between convergence and diversity is closer to the side of diversity for the stable matching-based selection. Therefore, when optimizing UF10, MOEA/D-STM2L and MOEA/D-STM tend to get stuck in some local Pareto optimal solutions to maintain the diversity of the population.

For the MOP test instances, the advantage of MOEA/D-STM2L on diversity is much clearer. MOEA/D-STM2L outperforms MOEA/D-DRA and MOEA/D-STM to a large extent on all MOP test instances. MOEA/D-IR has slightly better results only for 2 test instances.

TABLE I: IGD Results Of MOEA/D-STM2L And Three Comparing MOEAs On 17 Test Instances

Instance	MOEA/D-STM2L	MOEA/D-DRA	MOEA/D-STM	MOEA/D-IR
UF1	**1.009E-03**	1.632E-03†	1.058E-03†	1.021E-03
	5.98E-05	1.72E-04	5.62E-05	7.87E-05
UF2	**2.569E-03**	3.730E-03†	3.031E-03	2.763E-03
	5.55E-04	1.31E-03	1.19E-03	1.42E-03
UF3	**4.688E-03**	7.443E-03	5.176E-03	3.310E-02†
	3.86E-03	8.24E-03	4.17E-03	2.41E-02
UF4	5.267E-02	5.708E-02†	**5.097E-02**	5.221E-02
	2.98E-03	3.22E-03	**3.46E-03**	3.50E-03
UF5	2.548E-01	3.163E-01	**2.490E-01**	2.532E-01
	2.34E-02	1.11E-01	**1.62E-02**	2.27E-02
UF6	**6.399E-02**	2.866E-01†	7.055E-02	1.237E-01†
	2.15E-02	2.52E-01	3.41E-02	5.83E-02
UF7	1.146E-03	1.797E-03†	1.169E-03	**1.082E-03**‡
	7.99E-05	2.94E-04	5.91E-05	**1.80E-04**
UF8	3.611E-02	3.576E-02	3.959E-02	**3.151E-02**‡
	6.72E-03	9.99E-03	1.12E-02	**4.78E-03**
UF9	**2.506E-02**	1.005E-01†	2.513E-02	4.125E-02†
	1.55E-03	5.39E-02	1.27E-03	3.26E-02
UF10	1.430E+00	**4.403E-01**‡	1.715E+00†	6.010E-01‡
	2.38E-01	**7.63E-02**	2.52E-01	7.92E-02
MOP1	2.322E-02	3.445E-01†	3.367E-01†	**2.197E-02**‡
	1.74E-03	3.85E-02	5.40E-02	**1.83E-03**
MOP2	**2.868E-02**	2.816E-01†	2.496E-01†	3.822E-02†
	4.42E-02	6.64E-02	6.30E-02	7.85E-02
MOP3	**4.263E-02**	4.842E-01†	4.850E-01†	4.529E-02
	9.48E-02	2.85E-02	3.17E-02	8.31E-02
MOP4	**4.914E-02**	3.118E-01†	2.911E-01†	4.987E-02
	7.28E-02	1.95E-02	3.13E-02	7.60E-02
MOP5	1.999E-02	3.145E-01†	3.122E-01†	**1.994E-02**
	1.92E-03	9.44E-03	1.23E-02	**1.28E-03**
MOP6	**5.299E-02**	3.045E-01†	2.989E-01†	5.384E-02‡
	1.17E-03	2.28E-08	1.57E-02	3.41E-03
MOP7	**7.259E-02**	3.484E-01†	3.512E-01†	7.592E-02†
	2.63E-03	1.19E-02	1.30E-07	3.02E-03

The best mean value for each test instance is highlighted in boldface. Wilcoxons rank sum test at 5% significance level is conducted between MOEA/D-STM2L and its comparing MOEAs. $\dagger$ and $\ddagger$ indicates that MOEA/D-STM2L is significantly better or worse than the corresponding MOEA, respectively.

B. Impact of Parameter Settings

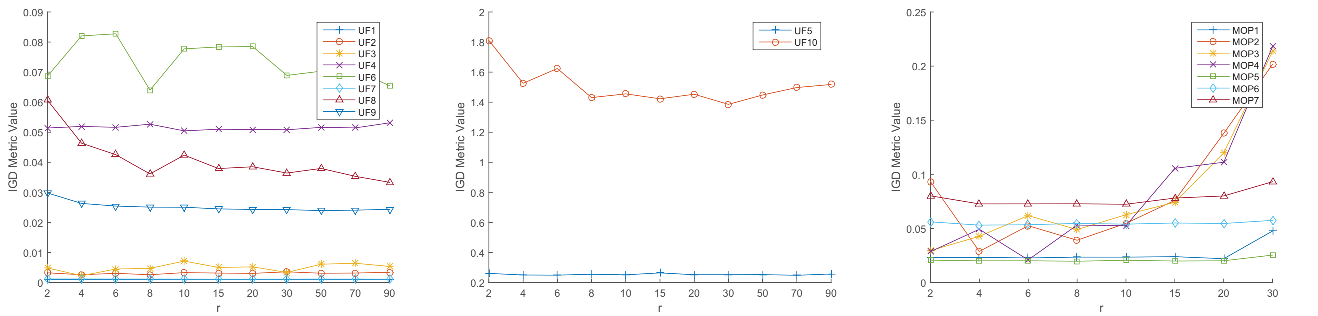
The MOEA/D-STM2L introduces a new parameter r to the stable matching based-selection. It controls to what degree the first stable matching limits the preferred subproblems by solutions. In order to study the impact of r on the effectiveness of the two-level stable matching-based selection, MOEA/D-STM2L is run for all test instances using different r settings. The mean IGD metric values of 20 independent runs are shown in Fig. 1(a)-Fig. 1(c).

The UF test instance set is a set of complicated MOPs with diverse features. Fig. 1(a) and Fig. 1(b) shows that UF1-UF3 and UF10 have limited suitable region of r. Either too high or too low has a bad influence on the performance of the MOEA/D-STM2L. For other UF test instances, the performance MOEA/D-STM2L is less dependent on the setting of r. But it is certainly not desirable if the setting of r is too low, resulting in a lack of convergence. Nonetheless, when the r is too high, the two-level of stable matching is then reduced to one-level. Taken all UF test instances into account, r is set to be 8 for UF test instances.

In terms of the MOP test instances, the effects of the two-level stable matching-based selection is more significant. In order to find a set of evenly distributed solutions to estimate the whole PF of the MOP test instances, more importance should be attached to guarantee the diversity of the solutions. Fig. 1(c) indicates that when the r goes too high, the performance of MOEA/D-STM2L drops dramatically. Therefore, r is set to be a smaller value for MOP1-MOP7, i.e. $r = 4$.

VI. Conclusion

In this paper, the two-level stable matching-based selection scheme is proposed for MOEA/D. The stable matching-based



(a) UF1-UF4 and UF6-UF9. (b) UF5 and UF10. (c) MOP1-MOP7.

Fig. 1: Parameter sensitivity studies of parameter r.

selection models the selection of solutions to subproblems as a matching process by defining the preferences between them. It regards a stable matching between subproblems and solutions as a balance between convergence and diversity. In the two-level stable matching-based selection, the first-level stable matching further moves the balance towards the side of diversity by restricting the number of preferred subproblems of a solution. Then, a second-level stable matching matches the remaining unmatched subproblems and solutions. Experimental studies show that the proposed MOEA/D-STM2L outperforms other state-of-the-art variants of MOEA/D on most of the test instances and provides comparable results to the best one in other cases.

The two-level stable matching-based selection achieves a significant improvement to the original stable matching-based selection. In future, we would like to further investigate the inability of stable matching-based selection on some test instances. Furthermore, reproduction operator is also important in MOEA design. It is interesting to investigate our newly developed reproduction operators, e.g., adaptive operator selection [19]–[21], local search [22] and manifold learning [23] in MOEA/DSTM2L.

Acknowledgment

This work is partially supported by Hong Kong RGC General Research Fund GRF grant 9042038 (CityU 11205314).

References

[1] K. Li, S. Kwong, J. Cao, M. Li, J. Zheng, and R. Shen, Achieving Balance Between Proximity and Diversity in Multi-objective Evolutionary Algorithm. *Information Sciences*, vol. 182, no. 1, pp. 220-242, Jan.2012.

[2] K. Li, S. Kwong, K. Deb, A Dual-Population Paradigm for Evolutionary Multiobjective Optimization. *Information Sciences*, vol. 309, pp. 50-72, 2015.

[3] 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.

[4] E. Zitzler, M. Laumanns, and L. Thiele, SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN)*, pp. 95-100, 2002.

[5] N. Beume, B. Naujoks, and M. Emmerich, SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653-1669, Sept. 2007.

[6] 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, 1999.

[7] 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.

[8] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, Stable matching-based selection in evolutionary multiobjective optimization. *IEEE Trans. Evol. Comput.*, vol. 18, no. 6, pp. 909-923, Dec. 2014.

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

[10] K. Li, S. Kwong, Q. Zhang, and K. Deb, Interrelationship based selection for decomposition multiobjective optimization. *IEEE Trans. Cybern.*, vol. PP, no. 99, pp. 1-13, Dec. 2014.

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

[12] I. Giagkiozis, and P. J. Fleming, Methods for multi-objective optimization: An analysis. *Information Sciences*, vol. 293, pp. 338-350, Feb. 2015.

[13] 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.*, pp. 203-208, May 2009.

[14] D. Gusfield, and R. W. Irving, *The stable marriage problem: structure and algorithms*, vol. 54. Cambridge: MIT press, 1989.

[15] 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. University of Essex and Nanyang Technological University, Tech. Rep. CES-487, 2008.

[16] H. Liu, F. Gu, and Q. Zhang, Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450-455, Jun. 2014.

[17] 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.

[18] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, Inc., 2001.

[19] K. Li, A. Fialho, S. Kwong, Q. Zhang, Adaptive Operator Selection With Bandits for a Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Tran. Evol. Comput.*, vol. 18, no. 1, pp. 114-130, Feb, 2014.

[20] K. Li, S. Kwong, R. Wang, J. Cao, I. Rudas, Multi-objective differential evolution with self-navigation. *SMC'12: Proc. of 2012 IEEE Conference on Systems, Man, and Cybernetics*, pp. 508-513, 2012.

[21] K. Li, A. Fialho, S. Kwong, Multi-objective differential evolution with adaptive control of parameters and operators. *LION'5: Proc. of the 5th International Conference on Learning and Intelligent Optimization*, pp. 473-487, Jan. 2011.

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

[23] K. Li, S. Kwong, A general framework for evolutionary multiobjective optimization via manifold learning. *Neurocomputing*, vol. 146, pp. 65-74.