Letters

Decomposition of a Multiobjective Optimization Problem into a Number of Simple Multiobjective Subproblems

Hai-Lin Liu, Fangqing Gu, and
Qingfu Zhang, *Senior Member, IEEE*

Abstract—**This letter suggests an approach for decomposing a multiobjective optimization problem (MOP) into a set of simple multiobjective optimization subproblems. Using this approach, it proposes MOEA/D-M2M, a new version of multiobjective optimization evolutionary algorithm-based decomposition. This proposed algorithm solves these subproblems in a collaborative way. Each subproblem has its own population and receives computational effort at each generation. In such a way, population diversity can be maintained, which is critical for solving some MOPs. Experimental studies have been conducted to compare MOEA/D-M2M with classic MOEA/D and NSGA-II. This letter argues that population diversity is more important than convergence in multiobjective evolutionary algorithms for dealing with some MOPs. It also explains why MOEA/D-M2M performs better.**

Index Terms—**Decomposition, hybrid algorithms, multiobjective optimization.**

I. Introduction

This letter considers the following continuous multiobjective optimization problem (MOP):

$$\begin{aligned} &\text{minimize} \quad F(x) = (f_1(x), \dots, f_m(x)) \\ &\text{subject to} \quad x \in \prod_{i=1}^{n}[a_i, b_i] \end{aligned} \tag{1}$$

where $\prod_{i=1}^{n}[a_i, b_i]$ is the decision space. $f : \prod_{i=1}^{n}[a_i, b_i] \to R^m$ consists of m real-valued continuous objective functions $f_1, \dots, f_m$. R^m is the objective space. Since these objectives conflict with one another, no single solution can optimize them at the same time. Let $u = (u_1, ..., u_m)$ and $v = (v_1, ..., v_m)$, u dominates v if and only if $u_i \leq v_i$ for every i and there exists one index j such that $u_j < v_j$. If there is no $x \in [a, b]^n$ such that $F(x)$ dominates $F(x^*)$, x^* is called a (globally) Pareto-optimal point and $F(x^*)$ is a Pareto-optimal objective vector. In other words, any improvement in one objective at a Pareto-optimal point must lead to a deterioration in at least one other objective. A Pareto-optimal solution is an optimal tradeoff candidate among all objectives. The set of all Pareto-optimal points is called the Pareto set (PS) and the set of all Pareto-optimal objective vectors is the Pareto front (PF) [3]. Under mild conditions, both the PS and the PF are $(m-1)$-D piecewise continuous manifolds [1]. Very often, a decision maker requires an approximation to the PF for having a good insight to the problem and making his or her final choice.

A number of multiobjective optimization evolutionary algorithms (MOEAs) have been developed for finding a set of solutions to approximate the PF in a single run [2]–[6]. Most MOEAs, such as NSGA-II [4], mainly rely on Pareto dominance to guide their search, particularly, their selection operators. In contrast, multiobjective evolutionary algorithm-based on decomposition (MOEA/D) [6] makes use of traditional aggregation methods to transform the task of approximating the PF into a number of single objective optimization subproblems. Then, a population-based algorithm is employed to solve these subproblems in a collaborative way. Some MOEA/D variants have been proposed for dealing with various MOPs [7], [8]. MOEA/D has also been used as a basic element in some hybrid algorithms [9]–[11].

Typically effective MOEAs require good population diversity since their goal is to approximate a set instead of a single point. The current MOEA/D algorithms achieve their population diversity via the diversity of their subproblems. These algorithms adopt an elite strategy in their selection. Whether or not a new solution replaces an old solution is completely determined by their aggregation function values. In some cases, such replacement can cause a severe loss of population diversity. As a result, they may miss some search regions and slow down their search. Another shortcoming is that a good performance of MOEA/D on a particular problem often requires a user to choose an appropriate aggregation method and a suitable setting of weight vectors, which may not always be an easy task. To reduce these shortcomings, this letter generalizes the original MOEA/D and proposes an approach for decomposing (1) into a number of simple multiobjective optimization subproblems. Based on this decomposition, a new MOEA/D algorithm, MOEA/D-M2M, is designed. We show that MOEA/D-M2M can significantly outperform MOEA/D and NSGA-II on a set of test instances used in our experimental studies. Taking one instance as an example, we explain why MOEA/D-M2M can maintain better population diversity than the other two algorithms.

II. Main Idea and Algorithm

A. Decomposition

For simplicity, this letter assumes that all the individual objective functions $f_1, \dots, f_m$ are nonnegative.[1] Therefore, all the objective vectors and, thus, the PF of (1) are in R^m_+.

Manuscript received July 18, 2012; revised March 13, 2013; accepted April 23, 2013. Date of publication September 11, 2013; date of current version May 27, 2014. This work was supported in part by the Natural Science Foundation of China under Grant 60974077, in part by the Natural Science Foundation of Guangdong Province under Grant S2012010008813, in part by the Programme of Science and Technology of Guangdong Province under Grant 2012B091100033, and in part by the Programme of Science and Technology, Department of Education of Guangdong Province under Grant 2012KJCX0042.

H. L. Liu and F. Gu are with Guangdong University of Technology, Guangdong 510520, China (email: hlliu@gdut.edu.cn; gufangqing84@163.com).

Q. Zhang is with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong SAR (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).

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.2013.2281533

[1] Otherwise, we can replace f_i by f_i+M, where M is a large enough positive number so that $f_i + M > 0$ for all i.

The main idea behind MOEA/D-M2M is to decompose (1) into a set of simple multiobjective optimization subproblems and then solve them in one single run. For this purpose, we first choose K unit vectors $v^1, \dots, v^K$ in R^m_+. Then, divide R^m_+ into K subregions $\Omega_1, \dots, \Omega_K$, where Ω_k $(k = 1, \dots, K)$ is

$$\Omega_k = \{u \in R^m_+ | \langle u, v^k \rangle \leq \langle u, v^j \rangle \text{ for any } j = 1, \dots, K\} \tag{2}$$

where $\langle u, v^j \rangle$ is the acute angle between u and v^j. In other words, u in Ω_k if and only if v^k has the smallest angle to u among all the K direction vectors. Based on this division, (1) can be transformed into K constrained multiobjective optimization subproblems. Subproblem k is

$$\begin{aligned} \text{minimize} \quad & F(x) = (f_1(x), \dots, f_m(x)), \\ \text{subject to} \quad & x \in \prod_{i=1}^{n} [a_i, b_i], \\ & F(x) \in \Omega_k. \end{aligned} \tag{3}$$

We would like to make the following comments on the above transformation.

1) This transformation is equivalent in the sense that the PFs of all these subproblems constitute the PF of (1). In contrast, the transformation used in the original MOEA/D is approximate since the optimal solutions of all the single objective optimization subproblems are just an approximation to the PF of (1).
2) Even when the PS of (1) has a nonlinear geometric shape, the PS of (3) could be close to linear due to the constraint $F(x) \in \Omega_k$ since it is just a small part of the PS of (1). Therefore, (3) could be simpler than (1), at least in terms of PS shapes. Note that Pareto-based MOEAs, such as NSGA-II, are suitable for dealing with simple PSs [1] and, thus, elements from these algorithms can be used for tackling these subproblems.
3) This transformation does not require any aggregation methods. What a user needs to do is to choose a set of direction vectors. To some extent, it requires less human labor than the transformation used in original MOEA/D framework.

B. Algorithm

MOEA/D-M2M uses the decomposition strategy proposed above to decompose (1) into K multiobjective optimization subproblems and solves them in a collaborative manner. At each generation, MOEA/D-M2M maintains K subpopulations: $P_1, \dots, P_K$, where P_k is for subproblem k. Each subpopulation contains S individual solutions. The F-value (i.e., objective function vector) of each solution is also recorded.

MOEA/D-M2M is described in Algorithm 1. Both lines 1 and 13 in Algorithm 1 use a number of individual solutions to set $P_1, \dots, P_K$. We use the simple approach described in Algorithm 2 to do it.

Algorithm 1: MOEA/D-M2M

```
   Input :
     1) MOP (1);
     2) A stopping criterion;
     3) $K$: the number of the subproblems;
     4) $K$ unit direction vectors: $v^1, \dots, v^K$;
     5) $S$: the size of subpopulation;
     6) Genetic operators and their associated parameters.
   Output: $\Psi$: a set of nondominated solutions
1  Initialization: Uniformly randomly choose $K \times S$ points
   from $[a, b]^n$, compute their $F$-values and then use them
   to set $P_1, \dots, P_K$.
2  while the stopping criterion is not met do
3      Generation of New Solutions:
4      Set $R = \emptyset$;
5      for $k \leftarrow 1$ to $K$ do
6          foreach $x \in P_k$ do
7              Randomly choose $y$ from $P_k$;
8              Apply genetic operators on $x$ and $y$ to
               generate a new solution $z$;
9              Compute $F(z)$;
10             $R := R \cup \{z\}$;
11         end
12         $Q := R \cup (\cup_{k=1}^{K} P_k)$;
13         use $Q$ to set $P_1, \dots, P_K$.
14     end
15     Find all the nondominated solutions in $\cup_{k=1}^{K} P_k$ and
       output them.
16 end
```

Algorithm 2: Allocation of Individuals to Subpopulations

```
Input  : $Q$: a set of individual solutions in $[a, b]^n$ and
         their $F$-values.
Output: $P_1, \dots, P_K$.
for $k \leftarrow 1$ to $K$ do
    Initialize $P_k$ as the solutions in $Q$ whose $F$-values
    are in $\Omega_k$;
    if $|P_k| < S$ then
        randomly select $S - |P_k|$ solutions from $Q$ and
        add them to $P_k$.
    end
    if $|P_k| > S$ then
        rank the solutions in $P_k$ using the nondominated
        sorting method [4] and remove from $P_k$ the
        $S - |P_k|$ lowest ranked solutions.
    end
end
```

Algorithm 2 ensures that each P_k has S solutions at each generation and, thus, promotes population diversity during the search. This is very important when some multiobjective subproblems are much more difficult than others. The MOEA/D variant in [1] and most Pareto dominance algorithms may not be able to solve such MOPs in an efficient way. The MOEA/D in [1] makes its selection by using aggregation functions and these Pareto dominance algorithms do nondominance ranking globally in their selection. Therefore, some hard multiobjective subproblems may receive very few or even no solutions for the next generation if, as it is very likely to happen, the current solutions in their feasible regions are completely dominated by other solutions. As a result, a great loss of population diversity will be made.

Line 7 of Algorithm 2 uses the nondominated sorting method in NSGA-II. Actually, any other selection strategies can be used for this purpose.

Some parallel MOEAs and multiobjective particle swarm optimization [12]–[15] and multiple populations-based MOEAs [16]–[19] also make use of multiple populations to target different parts of the PF explicitly or implicitly. Among these existing algorithms, cone-separated NSGA-II (CS-NSGA-II) [19] works in a similar way to our proposed MOEA/D-M2M. The major differences between CS-NSGA-II and MOEA/D-M2M include the following.

1) The decomposition method in MOEA/D-M2M is much simpler than the one in CS-NSGA-II. CS-NSGA-II normalizes the objective functions so that all the current solutions are in a unit m-D cube, and then divides this cube into several cones. Branke *et al.* [19] only explained how to do it in the case of two and three objectives. It is not trivial to use their method to do space division in the case of more than three objectives. Whereas the decomposition in MOEA/D-M2M is quite straightforward and only needs several unit direction vectors.
2) The space division in CS-NSGA-II is intended for a parallel computing environment with several processors. At some generations, a processor shares its infeasible solutions with some other processors. Whereas decomposition in MOEA/D-M2M is mainly for balancing diversity and convergence, at each generation, it merges all the subpopulations and then reallocates them to different subproblems.

III. Experimental Studies

We have compared MOEA/D-M2M with MOEA/D-DE and NSGA-II in our experimental studies. MOEA/D-DE is an efficient and effective implementation of MOEA/D for continuous MOPs proposed in [1].

A. Experimental Setting

1) The crossover and mutation operators with the same control parameters in [20] are used in the three algorithms for generating new solutions.
2) For the biobjective instances, the population size is 100 in both NSGA-II and MOEA/D-DE. In the case of the three objectives, it is 300 for NSGA-II and MOEA/D-DE.
3) For the biobjective instances, $K = S = 10$ in MOEA/D-M2M. For the triobjective instances, $K = S = 17$ in MOEA/D-M2M.
4) Stopping criterion: all three algorithms stop after 3000 generations.
5) Other control parameters in MOEA/D-DE: $T = 20$, $\delta = 0.9$ and $n_r = 2$, which are the same as in [1].
6) The K direction vectors are uniformly selected from the unit sphere in the first octant in MOEA/D-M2M.

Clearly, with the above settings, MOEA/D-M2M evaluates slightly fewer solutions at each generation than MOEA/D and NSGA-II in the case of three objectives.

B. Performance Metrics

1) *IGD-Metric:* The IGD-metric is used to measure the quality of a solution set P in our experiments. Suppose that P^* is a set of points, which are uniformly distributed along the PF in objective space, and P is an approximation to the PF. The distance between the P^* and P can be defined as

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|}$$

where $d(v, P)$ is the minimum Euclidean distance from the point v to P. Obviously, the smaller the value of IGD is, the better the algorithm performs. Five hundred points for two-objective test instances and 1000 points for three-objective test instances are uniformly sampled on the PF to form P^*.

2) *H-Metric:* Let $y^* = (y_1^*, \ldots, y_m^*)$ be a point in the objective space, which is dominated by any Pareto-optimal objective vectors. Let S be the obtained approximation to the PF in the objective space. Then, the I_H value of S (with regard to y^*) is the volume of the region, which is dominated by S and dominates y^*. In our experiments, $y^* = (1, \ldots, 1)$. The larger the H-metric is, the better the approximation is.

C. Test Instances

The following modified ZDT and DTLZ instances are used [21]. $g(x)$ functions used in our modified instances are different from those in their original versions. Their search space is $[0, 1]^n$. $n = 10$ unless stated otherwise. All these instances are for minimization.

$$\text{MOP1: } \begin{cases} f_1(x) = (1 + g(x))x_1 \\ f_2(x) = (1 + g(x))\left(1 - \sqrt{x_1}\right) \end{cases}$$

where

$g(x) = 2\sin(\pi x_1)\sum_{i=2}^{n}\left(-0.9t_i^2 + |t_i|^{0.6}\right)$;
$t_i = x_i - \sin(0.5\pi x_1)$;
its PF is $f_2 = 1 - \sqrt{f_1}$, $0 \leq f_1 \leq 1$;
its PS is $\{(x_1, \cdots, x_n)|0 < x_1 < 1, x_j = \sin(0.5\pi x_1)$;
$j = 2, \ldots, n$; or $x_1 = 0, 1.\}$.

$$\text{MOP2: } \begin{cases} f_1(x) = (1 + g(x))x_1 \\ f_2(x) = (1 + g(x))\left(1 - x_1^2\right) \end{cases}$$

where

$g(x) = 10\sin(\pi x_1)\sum_{i=2}^{n}\frac{|t_i|}{1+e^{5|t_i|}}$;
$t_i = x_i - \sin(0.5\pi x_1)$;
its PF is $f_2 = 1 - f_1^2$, $0 \leq f_1 \leq 1$;
its PS is $\{(x_1, \cdots, x_n)|0 < x_1 < 1, x_j = \sin(0.5\pi x_1)$;
$j = 2, \ldots, n$; or $x_1 = 0, 1.\}$.

$$\text{MOP3: } \begin{cases} f_1(x) = (1 + g(x))\cos(\frac{\pi x_1}{2}) \\ f_2(x) = (1 + g(x))\sin(\frac{\pi x_1}{2}) \end{cases}$$

where

$g(x) = 10\sin(\frac{\pi x_1}{2})\sum_{i=2}^{n}\frac{|t_i|}{1+e^{5|t_i|}}$;
$t_i = x_i - \sin(0.5\pi x_1)$;
its PF is $f_2 = \sqrt{1 - f_1^2}$, $0 \leq f_1 \leq 1$;
its PS is $\{(x_1, \cdots, x_n)|0 < x_1 \leq 1, x_j = \sin(0.5\pi x_1)$;
$j = 2, \ldots, n$; or $x_1 = 0, 1.\}$.

TABLE I
MEAN AND BEST OF IGD-METRIC VALUES OF MOEA/D-M2M, MOEA/D-DE, AND NSGA-II IN 20 INDEPENDENT RUNS FOR EACH TEST INSTANCE

IGD-metric	MOEA/D-M2M		MOEA/D-DE		NSGA-II	
Instance	best	mean	best	mean	best	mean
MOP1	**0.0151**	**0.0179**	0.2897	0.3239	0.2129	0.2206
MOP2	**0.0103**	**0.0118**	0.2167	0.2342	0.2103	0.2121
MOP3	**0.0116**	**0.0123**	0.4437	0.4798	0.2611	0.2660
MOP4	**0.0091**	**0.0102**	0.2662	0.2738	0.2745	0.2826
MOP5	**0.0153**	**0.0209**	0.2657	0.2925	0.2419	0.2442
MOP6	**0.0513**	**0.0526**	0.3039	0.3040	0.3040	0.3044
MOP7	**0.0623**	**0.0780**	0.3507	0.3507	0.3505	0.3505

TABLE II
MEAN AND BEST OF H-METRIC VALUES OF MOEA/D-M2M, MOEA/D-DE, AND NSGA-II IN 20 INDEPENDENT RUNS FOR EACH TEST INSTANCE

H-metric	MOEA/D-M2M		MOEA/D-DE		NSGA-II	
Instance	best	mean	best	mean	best	mean
MOP1	**0.6500**	**0.6499**	0.1326	0.1208	0.1862	0.1503
MOP2	**0.3057**	**0.2947**	0.0812	0.0678	0.0570	0.0375
MOP3	**0.1871**	**0.1837**	0.0012	0.0004	0.0545	0.0286
MOP4	**0.5079**	**0.5077**	0.1346	0.1026	0.1641	0.1326
MOP5	**0.6361**	**0.6340**	0.2143	0.1801	0.2156	0.1841
MOP6	**0.7532**	**0.7257**	0.5433	0.5431	0.5432	0.5404
MOP7	**0.3773**	**0.3762**	0.2014	0.2014	0.2014	0.2014

$$\text{MOP4: } \begin{cases} f_1(x) = (1+g(x))x_1 \\ f_2(x) = (1+g(x))(1-x_1^{0.5}\cos^2(2\pi x_1)) \end{cases}$$

where

$g(x) = 10\sin(\pi x_1)\sum_{i=2}^{n}\frac{|t_i|}{1+e^{5|t_i|}}$;
$t_i = x_i - \sin(0.5\pi x_1)$;
its PF is discontinuous;
its PS is $\{(x_1,\cdots,x_n)|0 < x_1 < 1, x_j = \sin(0.5\pi x_1);$
$j = 2,\ldots,n;$ or $x_1 = 0, 1.\}$.

$$\text{MOP5: } \begin{cases} f_1(x) = (1+g(x))x_1 \\ f_2(x) = (1+g(x))\left(1-\sqrt{x_1}\right) \end{cases}$$

where

$g(x) = 2|\cos(\pi x_1)|\sum_{i=2}^{n}\left(-0.9t_i^2 + |t_i|^{0.6}\right)$;
$t_i = x_i - \sin(0.5\pi x_1)$;
its PF is $f_2 = 1-\sqrt{f_1}$, $0 \le f_1 \le 1$;
its PS is $\{(x_1,\cdots,x_n)|0 \le x_1 \le 1, x_j = \sin(0.5\pi x_1);$
$j = 2,\ldots,n;$ or $x_1 = 0.5\}$.

$$\text{MOP6: } \begin{cases} f_1(x) = (1+g(x))x_1x_2 \\ f_2(x) = (1+g(x))x_1(1-x_2) \\ f_3(x) = (1+g(x))(1-x_1) \end{cases}$$

where

$g(x) = 2\sin(\pi x_1)\sum_{i=3}^{n}\left(-0.9t_i^2 + |t_i|^{0.6}\right)$;
$t_i = x_i - x_1x_2$;
its PF is $f_1 + f_2 + f_3 = 1$, $0 \le f_1, f_2, f_3 \le 1$;
its PS is $\{(x_1,\cdots,x_n)|0 < x_1 < 1, 0 \le x_2 \le 1;$
$x_j = x_1x_2, j = 3,\ldots,n;$ or $x_1 = 0, 1.\}$.

$$\text{MOP7: } \begin{cases} f_1(x) = (1+g(x))\cos(\frac{x_1\pi}{2})\cos(\frac{x_2\pi}{2}) \\ f_2(x) = (1+g(x))\cos(\frac{x_1\pi}{2})\sin(\frac{x_2\pi}{2}) \\ f_3(x) = (1+g(x))\sin(\frac{x_1\pi}{2}) \end{cases}$$

where

$g(x) = 2\sin(\pi x_1)\sum_{i=3}^{n}\left(-0.9t_i^2 + |t_i|^{0.6}\right)$;
$t_i = x_i - x_1x_2$;
its PF is $f_1^2 + f_2^2 + f_3^2 = 1$, $0 \le f_1, f_2, f_3 \le 1$;
its PS is $\{(x_1,\cdots,x_n)|0 < x_1 < 1, 0 \le x_2 \le 1;$
$x_j = x_1x_2, j = 3,\ldots,n;$ or $x_1 = 0, 1.\}$.

D. Results

Table I shows the smallest (i.e., best) and the mean of the IGD-metric values of the algorithms for each test instance in 20 independent runs. Table II presents the largest (i.e., best) and the mean of the H-metric values in 20 runs. It is clear from these two tables that MOEA/D-M2M is much better than MOEA/D-DE and NSGA-II in terms of both metrics on all the test instances.

Fig. 1 plots in the objective space, the distribution of the final solutions obtained in the run with the median IGD-metric value of each algorithm for each test instance. Obviously, both MOEA/D-DE and NSGA-II cannot locate the global PF on any instance. In contrast, MOEA/D-M2M can approximate the PFs of these instances quite well.

To understand why MOEA/D-M2M performs much better than the two other algorithms, we take MOP1 with $n = 2$ as an instance and analyze the Pareto dominance relations among all the points in the search space $[0, 1]^2$. In Fig. 2, region A in the search space is corresponding to a very tiny area including $(1, 0)$ in the objective space, every point in A dominates any point in region B. Since both MOEA/D and NSGA-II adopt the convergence first and diversity second selection strategy, they can be easily trapped in region A. By contrast, MOEA/D-M2M always allocates the same number of solutions to each subproblems, for this reason, it will not spend its most efforts in region A and, thus, be able to find the global PF.

$N = K \times S$ can be regarded as the whole population size in MOEA/D-M2M. If we fix N, then MOEA/D-M2M has one extra control parameters K compared with NSGA-II. To investigate the effect of the parameter on the algorithm performance, we have conducted experiments with different values of K on MOP1. In our experiments, S is set to be $\frac{100}{K}$ to keep $N = 100$, all the other settings are the same as in Section III-A. The smallest value of S is two since the crossover operator requires at least two different solutions in each subpopulation. Therefore, the largest value of K is 50. The experimental results from 20 runs are summarized in Fig. 3 and 4.

It is evident from these two figures that the performance of the algorithm does not change dramatically when $K \ge 10$. These results indicate that MOEA/D-M2M is not very sensitive to this parameter. We should point out that a large value of K requires a large number of direction vectors, which can increase the computational overhead in allocation of individual solutions to subproblems.

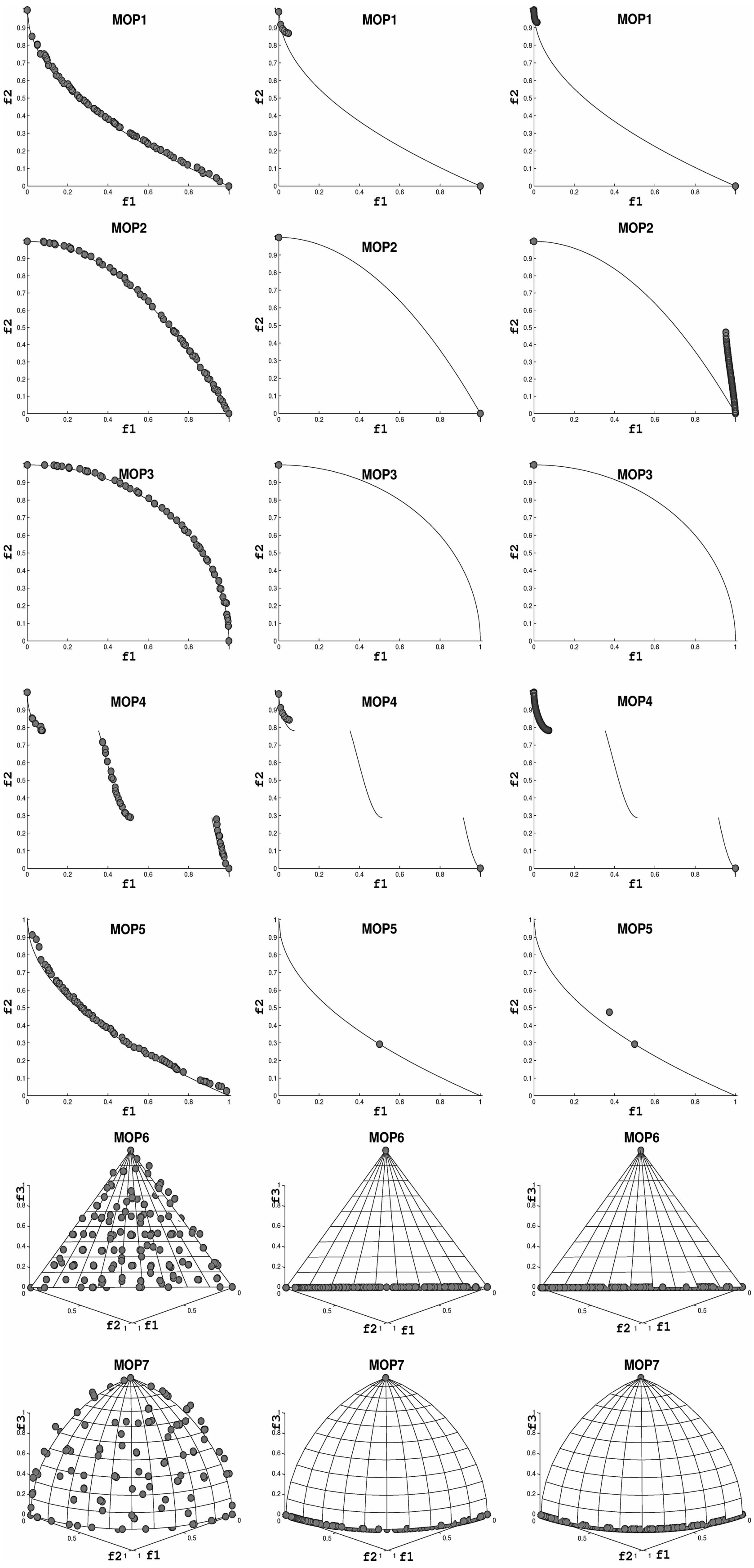

Fig. 1. Plot of the nondominated front with the median IGD-metric value found by MOEA/D-M2M (the left panel), MOEA/D-DE (the middle panel), and NSGA-II (the right panel).

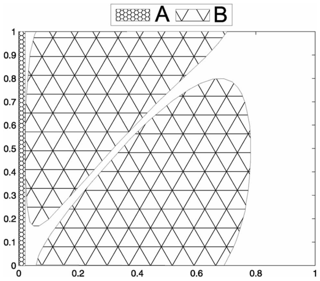

Fig. 2. Plot of the search space of MOP1 with $n = 2$. A point in region A dominates any point in region B.

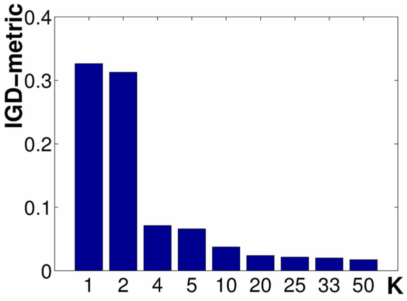

Fig. 3. Plot of the bar of the IGD-metric for different values of K for MOP1.

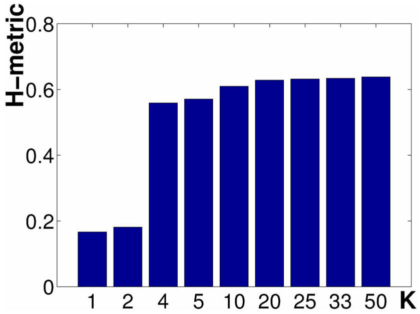

Fig. 4. Plot of the bar of the H-metric for different values of K for MOP1.

IV. Conclusion

This letter has proposed a simple way to decompose a MOP into a number of simple multiobjective subproblems. It also designed an algorithm called MOEA/D-M2M, which generalizes the MOEA/D framework. MOEA/D-M2M aims at solving these multiobjective subproblems in a collaborative manner. Each subproblem has a subpopulation and always receives computational effort during the search. In such a way, good population diversity can be achieved, which is essential for solving some MOPs. Experimental studies on a set of test instances have implied that MOEA/D-M2M can significantly outperform MOEA/D-DE and NSGA-II on a set of test instances.

Future work will include combinations of MOEA/D-M2M with other evolutionary search techniques and investigation of its performance on other hard multiobjective optimization problems.

References

[1] 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.

[2] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: Wiley, 2001.

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

[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] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. Evol. Meth. Design Optimiz. Control Applicat. Ind. Problems*, 2001, pp. 95–100.

[6] 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.

[7] Y. Mei, K. Tang, and X. Yao, "Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem," *IEEE Trans. Evol. Comput.*, vol. 15, no. 2, pp. 151–165, Apr. 2011.

[8] V. A. Shim, K. C. Tan, and C. Y. Cheong, "A hybrid estimation of distribution algorithm with decomposition for solving the multiobjective multiple traveling salesman problem," *IEEE Trans. Syst., Man, Cybern. C*, vol. 42, no. 5, pp. 682–691, Sep. 2012.

[9] S. Z. MartInez and Carlos. A. C. Coello, "A direct local search mechanism for decomposition-based multi-objective evolutionary algorithms," in *Proc. IEEE World Congr. Comput. Intell.*, Jun. 2012, pp. 3431–3438.

[10] K. Sindhya, K. Miettinen, and K. Deb, "A hybrid framework for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 495–511, Aug. 2012.

[11] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Effects of using two neighborhood structures on the performance of cellular evolutionary algorithms for many-objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, May 2009, pp. 2508–2515.

[12] T. Hiroyasu, M. Miki, and S. Watanabe, "The new model of parallel genetic algorithm in multi-objective optimization problems, divided range multi-objective genetic algorithm," in *Proc. Congr. Evol. Comput.*, vol. 1. Jul. 2000, pp. 333–340.

[13] A. L. Jaimes and C. A. C. Coello, "MRMOGA: Parallel evolutionary multiobjective optimization using multiple resolutions," in *Proc. Congr. Evol. Comput.*, vol. 3. 2005, pp. 2294–2301.

[14] D. VanVeldhuizen, J. Zydallis, and G. Lamont, "Considerations in engineering parallel multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 144–173, Apr. 2003.

[15] S. Mostaghim, J. Branke, and H. Schmeck, "Multi-objective particle swarm optimization on computer grids," in *Proc. Genetic Evol. Comput. Conf.*, 2007, pp. 869–875.

[16] T. Hiroyasu, M. Miki, and S. Watanabe, "Distributed genetic algorithms with a new sharing approach in multiobjective optimization problems," in *Proc. Congr. Evol. Comput.*, vol. 1. 1999, pp. 69–76.

[17] K. Deb, P. Zope, and A. Jain. "Distributed computing of Pareto-optimal solutions with evolutionary algorithms," in *Proc. 2th Conf. Evol. Multi-Criterion Optimiz.*, 2003, pp. 534–549.

[18] H. Liu and F. Gu, "A improved NSGA-II algorithm based on subregional search," in *Proc. Congr. Evol. Comput.*, 2011, pp. 1906–1911.

[19] J. Branke, H. Schmeck, K. Deb, and M. Reddy, "Parallelizing multiobjective evolutionary algorithms: Cone separation," in *Proc. Congr. Evol. Comput.*, 2004, pp. 1952–1957.

[20] H. Liu and X. Li, "The multiobjective evolutionary algorithm based on determined weight and sub-regional search," in *Proc. Congr. Evol. Comput.*, vol. 15. 2009, pp. 361–384.

[21] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.