

An External Archive Guided Multiobjective Evolutionary Algorithm Based on Decomposition for Combinatorial Optimization

Xinye Cai, *Member, IEEE*, Yexing Li, Zhun Fan, *Senior Member, IEEE*, and Qingfu Zhang, *Senior Member, IEEE*

Abstract—Domination-based sorting and decomposition are two basic strategies used in multiobjective evolutionary optimization. This paper proposes a hybrid multiobjective evolutionary algorithm integrating these two different strategies for combinatorial optimization problems with two or three objectives. The proposed algorithm works with an **internal (working) population and an external archive**. It uses a decomposition-based strategy for evolving its working population and uses a domination-based sorting for maintaining the external archive. Information extracted from the external archive is used to decide which search regions should be searched at each generation. In such a way, the domination-based sorting and the decomposition strategy can complement each other. In our experimental studies, the proposed algorithm is compared with a domination-based approach, a decomposition-based one, and one of its enhanced variants on two well-known multiobjective combinatorial optimization problems. Experimental results show that our proposed algorithm outperforms other approaches. The effects of the external archive in the proposed algorithm are also investigated and discussed.

Index Terms—Combinatorial multiobjective optimization, decomposition, Pareto optimality.

I. INTRODUCTION

A multiobjective optimization problem (MOP) can be stated as

$$\begin{aligned} &\text{maximize } F(x) = (f_1(x), \dots, f_m(x)) \\ &\text{subject to } x \in \Omega \end{aligned} \quad (1)$$

Manuscript received December 18, 2013; revised June 8, 2014; accepted August 2, 2014. Date of publication August 22, 2014; date of current version July 28, 2015. This work was supported in part by the National Natural Science Foundation of China under Grants 61300159, 61175073, 61332002, and 51375287; in part by the Natural Science Foundation of Jiangsu Province under Grant BK20130808; in part by the Research Fund for the Doctoral Program of Higher Education of China under Grant 20123218120041; in part by the Fundamental Research Funds for the Central Universities of China under Grant NZ2013306; and in part by the start up funds from City University of Hong Kong under Grant 7200386.

X. Cai and Y. Li are with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: xinye@nuaa.edu.cn; lyx19910329@126.com).

Z. Fan is with the Guangdong Provincial Key Laboratory of Digital Signal and Image Processing and the Department of Electronic Engineering, School of Engineering, Shantou University, Shantou 515063, China (e-mail: zfan@stu.edu.cn).

Q. Zhang is with the Department of Computer Science, City University of Hong Kong, Hong Kong, on leave from the School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park 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.2014.2350995

where Ω is the decision space, $F: \Omega \rightarrow R^m$ consists of m real-valued objective functions. The attainable objective set is $\{F(x)|x \in \Omega\}$. In the case when Ω is a finite set, (1) is called a discrete MOP.

Let $u, v \in R^m$, u is said to dominate v , denoted by $u \succ v$, if and only if $u_i \geq v_i$ for every $i \in \{1, \dots, m\}$ and $u_j > v_j$ for at least one index $j \in \{1, \dots, m\}$.¹ Given a set S in R^m , a point in it is called nondominated in S if no other point in S can dominate it. A point $x^* \in \Omega$ is Pareto-optimal if $F(x^*)$ is nondominated in the attainable objective set. $F(x^*)$ is then called a Pareto-optimal (objective) vector. In other words, any improvement in one objective of a Pareto optimal point must lead to deterioration to at least another objective. The set of all the Pareto-optimal points is called the Pareto set (PS) and the set of all the Pareto-optimal objective vectors is the Pareto front (PF) [17].

In many real-life applications, the PF is of great interest to decision makers for understanding the tradeoff relationship among different objectives and choosing their preferred solutions. Evolutionary algorithms have been recognized as a major method for approximating the PF. Along with the domination-based [6], [7], [31] and the performance-indicator-based algorithms [30], the multiobjective evolutionary algorithms based on decomposition (MOEA/D) [26] have been widely used and investigated in evolutionary computation community now. MOEA/D decomposes a MOP into a number of single objective optimization subproblems and then solves them in parallel. The objective function in each subproblem can be a linear or nonlinear weighted aggregation function of all the objective functions in the MOP in question. Two subproblems are called neighbors if their weight vectors are close to each other. MOEA/D explores correlation relationships among neighboring subproblems to speed up its search. A basic assumption in MOEA/D is that two neighboring problems should have similar optimal solutions. It could hold for most subproblems in a typical continuous MOP. It is very likely, however, that a nonnegligible number of subproblems do not satisfy this assumption in a combinatorial MOP. For this reason, MOEA/D alone may not perform well on some combinatorial MOPs. To address this issue, various combinations of MOEA/D and domination-based techniques have been investigated recently (see [1], [14], [16]). In these combinations,

¹In the case of minimization, the inequality signs should be reversed.

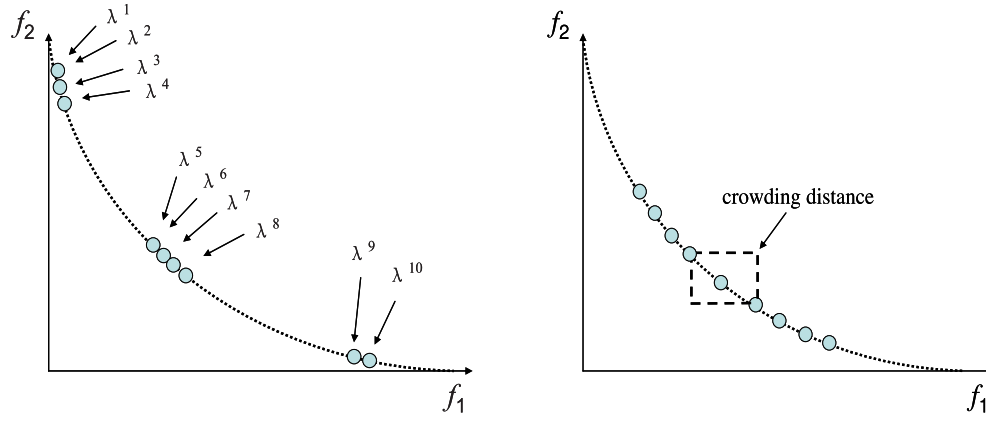


Fig. 1. (a) In a combinatorial MOP, some different subproblems such as ones with λ^1 and λ^2 can have the same optimal solutions. Even a large number of subproblems may not lead to a reasonably good approximation to the PF. (b) Principle of crowding distance assignment in NSGA-II [6], [7] can be used to select representative solution of good quality.

both the decomposition approach and the domination-based one are used for selecting good solutions.

Multiobjective evolutionary algorithms (MOEAs) generate and evaluate a population of solutions at each generation. From these solutions, one can use selection and genetic operators for producing new promising solutions. One can also use machine learning and other techniques to extract information from the previous search for guiding the further search. For example, a Bayesian rule miner is incorporated into a MOEA to identify promising search regions for dealing with noisy MOPs in [4]. In a variant of MOEA/D based on dynamic resource allocation (MOEA/D-DRA) [27], a utility value, defined and calculated for each subproblem based on the previous search performance on it, is used for guiding the allocation of computational resources among different subproblems. MOEA/D-DRA works well on continuous MOPs and won the CEC2009 competition. However, MOEA/D-DRA does not pay special attention to the population diversity, which may deteriorate its performance on combinatorial MOPs [16].

The rest of this paper is organized as follows. Section II explains our motivation of this paper. Section III describes the proposed scheme, called the external archive guided MOEA based on decomposition. Section IV introduces the multiobjective traveling salesman problem (MTSP) and the multiobjective next release problem (MNRP), two combinatorial MOPs used in our studies. Experimental studies and discussions are presented in Section V, where we compare our proposed algorithm with NSGA-II, MOEA/D, and MOEA/D-DRA. The effects of the external archive in our proposed algorithm are also investigated and discussed in Section V. Section VI concludes the paper.

II. MOTIVATION

A typical MOEA/D [26], decomposes a MOP into a number of single-objective subproblems and optimizes them in a collaborative way. The scalar objective function in each subproblem can be a linear or nonlinear weighted aggregation function of $f_1(x), \dots, f_m(x)$. MOEA/D defines a neighborhood relationship among subproblems

based on distances among their weight vectors. During its evolutionary process, MOEA/D maintains one solution for each subproblem. To generate a new one, it applies reproduction operators to current solutions of some of its neighboring subproblems. This generated new solution can also replace several current solutions of its neighboring subproblems when it outperforms them. A number of MOEA/D variants have been developed and used in various application domains [3], [10]–[12], [15], [18]–[20].

In this paper, a new variant of MOEA/D is proposed for multiobjective combinatorial optimization problems with two or three objectives. This paper is mainly motivated by the following considerations.

- 1) *Solution Distribution in MOEA/D*: Suppose that N subproblems are considered in MOEA/D and their weight (or direction) vectors are uniformly distributed in some sense. If the MOP is continuous and its PF is continuous and convex, the optimal solutions of these subproblems can constitute a good approximation to the PF when the number of weight vectors is large enough. However, it is often not the case for discrete MOPs as illustrated in Fig. 1(a). Some different subproblems such as ones with λ^1 and λ^2 can have the same optimal solutions. Even a large number of subproblems may not lead to a reasonably good approximation to the PF. This phenomenon, which has also been observed in [1] and in our experimental studies in Section V-C, explains why MOEA/D does not work well for some discrete MOPs. Moreover, search in one subproblem using MOEA/D for a discrete MOP at some stages can generate several diverse good solutions. Therefore, one can assume that each subproblem in MOEA/D may have varied contribution to the search process at different search stages. For this reason, different subproblems should NOT be allocated the same amounts of computational resources. This paper attempts to propose a mechanism of allocating computational resources to each subproblem dynamically.
- 2) *Subproblem's Utilities*: To decide how the computational resources should be distributed among different subproblems, one can first define an utility value for

each subproblem and then use these utility values to guide resource allocation at each generation. In MOEA/D-DRA [27], the utility of a subproblem is defined as the ratio of its objective function improvement to its allocated amount of computational effort over the last several generations. These utility functions work well for continuous MOPs. However, it may not be suitable for discrete MOPs since each utility function only measures the progress of one single subproblem but does not consider the contributions of all subproblems. This paper attempts to propose an utility function that considers the contributions of all subproblems to the search process in history.

- 3) *Nondominated Sorting and Crowding Distance*: Arguably, the nondominated sorting and crowding distance assignment in NSGA-II [6], [7] can effectively select representative solutions of good quality, in terms of both convergence and diversity, from a large set of solutions in the case of two or three objectives. The principle of crowding distance assignment is illustrated in Fig. 1(b). Recent studies (see [9], [25]) show that NSGA-II and MOEA/D are suitable for problems of complementary nature, which makes the combination of them very desirable. This paper attempts to utilize the idea from NSGA-II to help evaluate the contribution of each individual to the search process in the proposed new MOEA/D framework.

III. ALGORITHM

A. Framework

Like other MOEA/D variants, the proposed algorithm, termed the external archive guided MOEA/D (EAG-MOEA/D), decomposes the MOP into N single objective optimization subproblems. In principle, any aggregation methods can be used for this purpose. For simplicity, the weight sum approach is adopted in this paper, it requires N weight vectors

$$\lambda^j = (\lambda_1^j, \dots, \lambda_m^j) \quad j = 1, \dots, N \quad (2)$$

where $\lambda^j \in R_+^m$ and $\sum_{i=1}^m \lambda_i^j = 1$, and m is the number of objectives. The Subproblem k is

$$\begin{aligned} &\text{maximize } g_k(x) = \sum_{i=1}^m \lambda_i^j f_i(x) \\ &\text{subject to } x \in \Omega. \end{aligned} \quad (3)$$

For each $k = 1, \dots, N$, let $B(k)$ be the set containing the indexes of the T closest weight vectors to λ^k in terms of the Euclidean distance. If $i \in B(k)$, Subproblem i is called a neighbor of Subproblem k .

At each generation, EAG-MOEA/D maintains two populations.

- 1) $P = \{x^1, \dots, x^N\}$, where x^k is the best solution found so far to Subproblem k .
- 2) A , which has N solutions selected by using NSGA-II selection (the nondominated sorting approach and crowding distance assignment [6], [7]).

The algorithm works as follows.

- Step 1 *Initialization*: Initialize P and A .
- Step 2 *New Solution Generation*: Generate a set of N new solutions Y .
- Step 3 *Population Update*: Use Y to update P and A .
- Step 4 *Stopping Condition*: If a preset stopping condition is met, output A . Otherwise, go to Step 2.

The pseudocode of the algorithm is given in Algorithm 1. In Step 3, the method for updating P is based on decomposition while the method for A is NSGA-II selection.

The details of Steps 1–3 are given as follows.

B. Initialization

x^i in P can be generated randomly or by using a single objective heuristic on the Subproblem i . For simplicity, A is initialized to be P .

C. New Solution Generation

A new solution is called successful if it enters A in Step 3. Note that a new solution is generated by searching on a selected single objective subproblem and whether or not a new solution can enter A is determined by the NSGA-II selection. To guide the search by both subproblem search directions (i.e., decomposition) and domination-based sorting, we record $s_{i,k}$, the number of successful solutions generated by search on the Subproblem i at each generation k , and compute the total number of the successful solutions over the L previous generations

$$S_{i,G} = \sum_{k=G-L}^{G-1} s_{i,k}$$

where G is the current generation.

At each generation $G > L - 1$, the probability of selecting the Subproblem i is defined as

$$p_{i,G} = \frac{D_{i,G}}{\sum_{i=1}^N D_{i,G}} \quad (4)$$

where

$$D_{i,G} = \frac{S_{i,G}}{\sum_{j=1}^N S_{j,G}} + \epsilon, \quad (i = 1, 2, \dots, N). \quad (5)$$

$D_{i,G}$ is the proportion of successful solutions generated by the search on Subproblem i over the previous L generations. $\epsilon = 0.002$ is used to make all the $D_{i,G} > 0$. The above way for computing the probability has also been used in the ensemble of neighborhood size [29] for MOEA/D.

New solutions are generated in Step 2, detailed in Algorithm 1. In Step 2a, a subproblem is selected according to the probability defined in (4). For the selected subproblem, two parent solutions are selected from its T neighboring subproblems in Step 2b. In Step 2c, genetic operators are applied to the parents to generate an offspring y_j . By repeating this procedure (Step 2a–2c) N times, an offspring population $Y = \{y_1, \dots, y_N\}$ is generated. A subproblem may be selected for generating solutions more than one time at one generation.

Algorithm 1 EAG-MOEA/D**Input:**

- 1) a combinatorial MOP;
- 2) a stopping criterion;
- 3) N : the number of subproblems; the population size of P and A ;
- 4) $\lambda^1, \dots, \lambda^N$: a set of N weight vectors;
- 5) T : the size of the neighborhood of each subproblem.

Output: A set of nondominated solutions A ;**Step 1: Initialization:**

- a) Decompose the MOP into N subproblems associated with $\lambda^1, \dots, \lambda^N$.
- b) Generate an initial population $P = \{x^1, \dots, x^N\}$ randomly.
- c) Set $A = P$.
- d) Compute the Euclidean distance between any two weight vectors and obtain T closest weight vectors to each weight vector. For each $i = 1, \dots, N$, set $B(i) = \{i_1, \dots, i_T\}$, where $\lambda^{i_1}, \dots, \lambda^{i_T}$ are the T closest weight vectors to λ^i .

Step 2: New solution generation**for all** $j \in \{1, \dots, N\}$ **do**

- a) Select Subproblem i for search based on the probability defined in (4) and (5).
- b) Randomly select two indexes k and l from $B(i)$.
- c) Apply genetic operators on x_k and x_l to generate y_j for Subproblem i .

end for**Step 3: Population update****for all** $j \in \{1, \dots, N\}$ **do**

- a) If y_j is generated from subproblem i , for each index $k \in B(i)$, set $x_k = y_j$ if $g^{ws}(y_j|\lambda^k) \leq g^{ws}(x_k|\lambda^k)$.

end for/* update population A^* /

- b) Merge Y with A to obtain $Z = A \cup Y$; select N best solutions from the merged population Z by the NSGA-II selection to replace A .

Step 4: TerminationIf stopping criteria are satisfied, terminate the algorithm and output A . Otherwise, go to **Step 2**.**D. Population Update**

In Step 3, the solutions in Y are used to update both P and A . For solution y_j generated by search on subproblem i . Step 3a considers all the neighbors of Subproblem i , it replaces all neighbors x_k with y_j if y_j performs better than x_k with regard to the aggregated objective of the Subproblem k . The external population is updated in Step 3b. Y is merged with A first, and then the combined population Z is selected by the NSGA-II selection. The best N solutions are selected to form new A .

EAG-MOEA/D maintains two populations P and A . P stores the best solution found so far for each subproblem. A stores a population selected based on NSGA-II selection. A is used

for guiding the search, whereas the original MOEA/D doesn't do so.

In the original MOEA/D, each subproblem receives the same amount of computational resources. In EAG-MOEA/D, how likely a subproblem is selected for investment is determined on the contribution of the previous search on it to the external population. In MOEA/D-DRA, the probability that a subproblem is selected for investment is mainly determined by how well the previous search has performed on improving the solution to this subproblem, but not the whole population.

IV. TWO BENCHMARK COMBINATORIAL MOPS

Several combinatorial optimization problems have been widely used on testing MOEAs, (see [1], [2], [8], [14], [22]–[24], [28]). We consider two NP-hard combinatorial MOPs, the multiobjective software next release problem and the MTSP.

A. MNRP

Consider a software system with n requirements and U customers. Suppose the cost to implement requirement i is c_i , and the satisfaction score of customer u for requirement i is $s_{i,u}$.

The goal of the MNRP is to determine which requirements should be implemented in the next release of the software. The MNRP has two objectives to optimize. One objective is to minimize the required cost

$$C = \sum_{i=1}^n c_i \cdot x_i \quad (6)$$

and the other objective is to maximize the total satisfaction score

$$S = \sum_{u=1}^U \sum_{i=1}^n s_{i,u} \cdot x_i \quad (7)$$

where $x_i \in \{0, 1\}$. $x_i = 1$ means that requirement i is implemented, and $x_i = 0$ means that it is not implemented.

To apply MOEAs to this problem, we change the second objective to minimization of $-S$ and normalize both objectives so that their range is in $[0, 1]$.

A MNRP test instance with U customers and n requirements is denoted as $Cu - U/R - n$ in this paper. Eight randomly generated test instances are used in our studies, which include $Cu - 30/R - 300$, $Cu - 50/R - 200$, $Cu - 50/R - 500$, $Cu - 80/R - 800$, $Cu - 100/R - 1000$, $Cu - 120/R - 1200$, $Cu - 160/R - 1600$, and $Cu - 200/R - 2000$.

B. MTSP

Given n cities with edges connecting any two cities. Suppose that edge e has m distance metrics $d_{e,1}, \dots, d_{e,m}$. Each feasible solution is an edge subset which can form a Hamiltonian cycle. The i th objective in the MTSP is to minimize

$$f_i(x) = \sum_{e \in x} d_{e,i}. \quad (8)$$

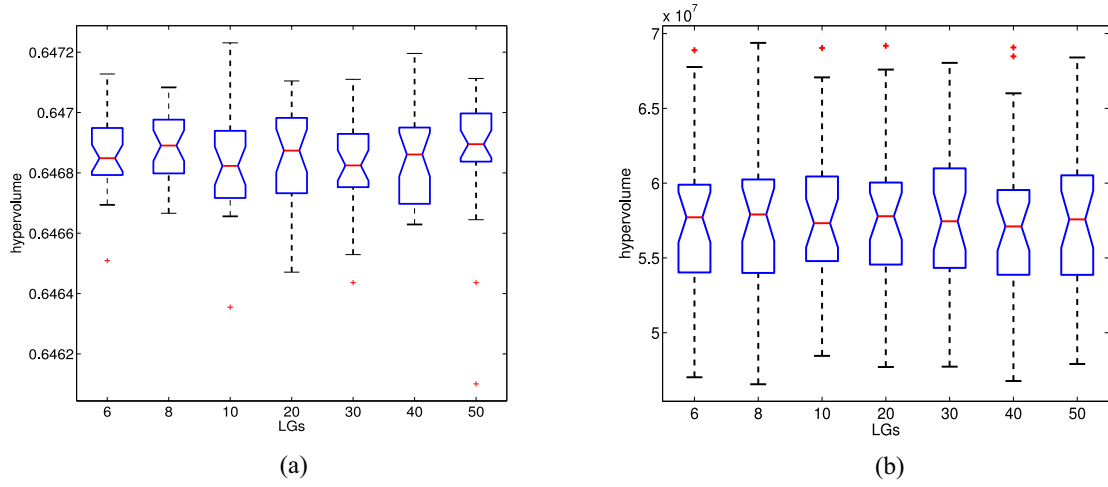


Fig. 2. Sensitivity tests for parameter LGs in EAG-MOEA/D on one MNRP and one MTSP instances. Instances show that parameter LGs are not very sensitive on both MNRP and MTSP instances. (a) $Cu = 50/R = 500$. (b) $c = 200/o = 2$.

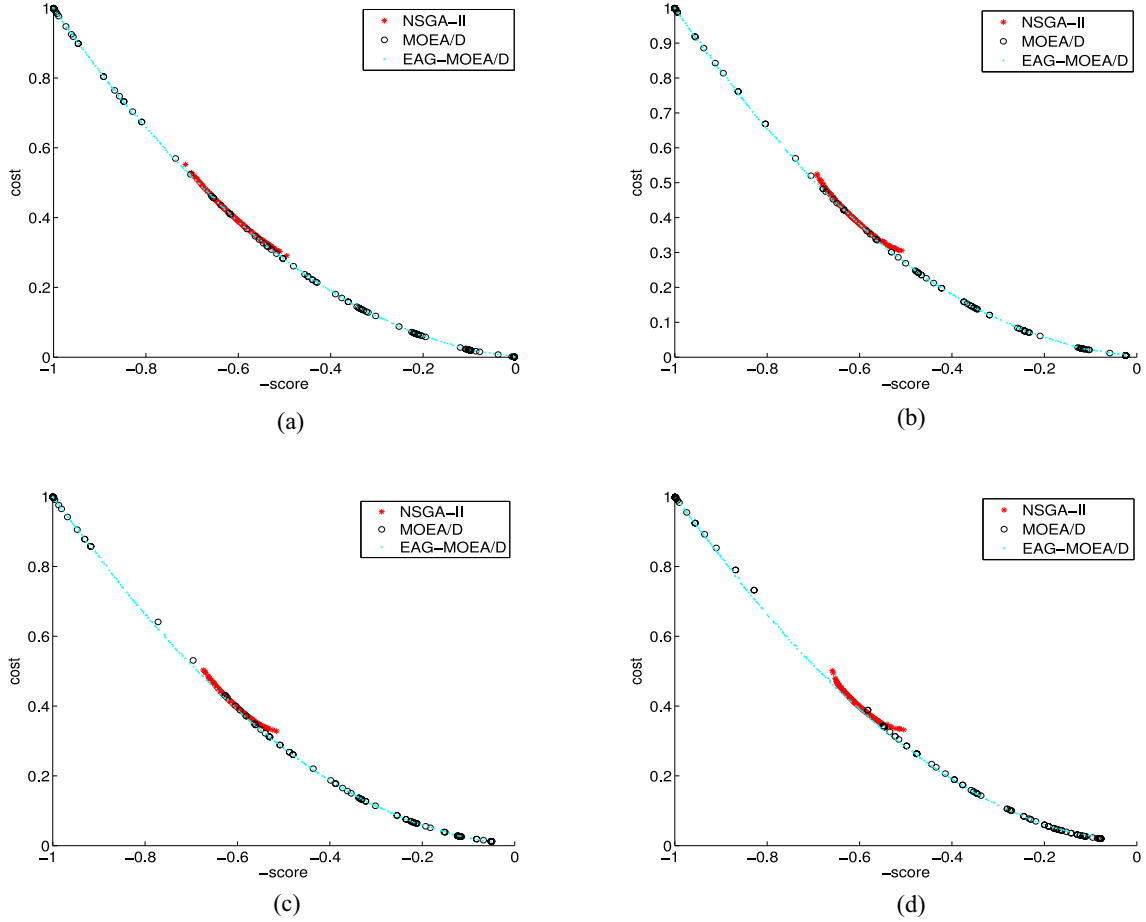


Fig. 3. Final nondominated solutions found by NSGA-II, MOEA/D, and EAG-MOEA/D on four MNRP instances. (a) $Cu = 80/R = 800$. (b) $Cu = 120/R = 1200$. (c) $Cu = 160/R = 1600$. (d) $Cu = 200/R = 2000$.

A MTSP with n cities and m objectives is denoted as $c - n/o - m$ in this paper. In our studies, we consider seven randomly generated test instance of MTSP with two objectives and two test instances of MTSP with three objectives, which include $c - 200/o - 2$, $c - 300/o - 2$, $c - 400/o - 2$, $c - 500/o - 2$, $c - 600/o - 2$, $c - 700/o - 2$, $c - 800/o - 2$, $c - 200/o - 3$, and $c - 300/o - 3$.

V. EXPERIMENTAL STUDIES AND DISCUSSION

To study the performance of EAG-MOEA/D and understand its behavior, this section conducts the following experimental work.

- 1) Comparison of EAG-MOEA/D with NSGA-II, MOEA/D, and MOEA/D-DRA on the two test problems.

TABLE I
PARAMETER SETTINGS IN EAG-MOEA/D, EA/MOEA/D, MOEA/D, AND NSGA-II

—	Population size		# of neighbors		Crossover rate		mutation rates		# of function evaluations	
	MNRP	MTSP	MNRP	MTSP	MNRP	MTSP	MNRP	MTSP	MNRP	MTSP
EAG-MOEA/D	200	100	20	10	1	1	1/n	0.01	50,000 - 10,000	400,000 - 1,500,000
EA-MOEA/D	200	100	20	10	1	1	1/n	0.01	50,000 - 10,000	400,000 - 1,500,000
MOEA/D	200	100	20	10	1	1	1/n	0.01	50,000 - 10,000	400,000 - 1,500,000
NSGA-II	200	100	20	10	1	0.8	1/n	0.01	50,000 - 10,000	400,000 - 1,500,000

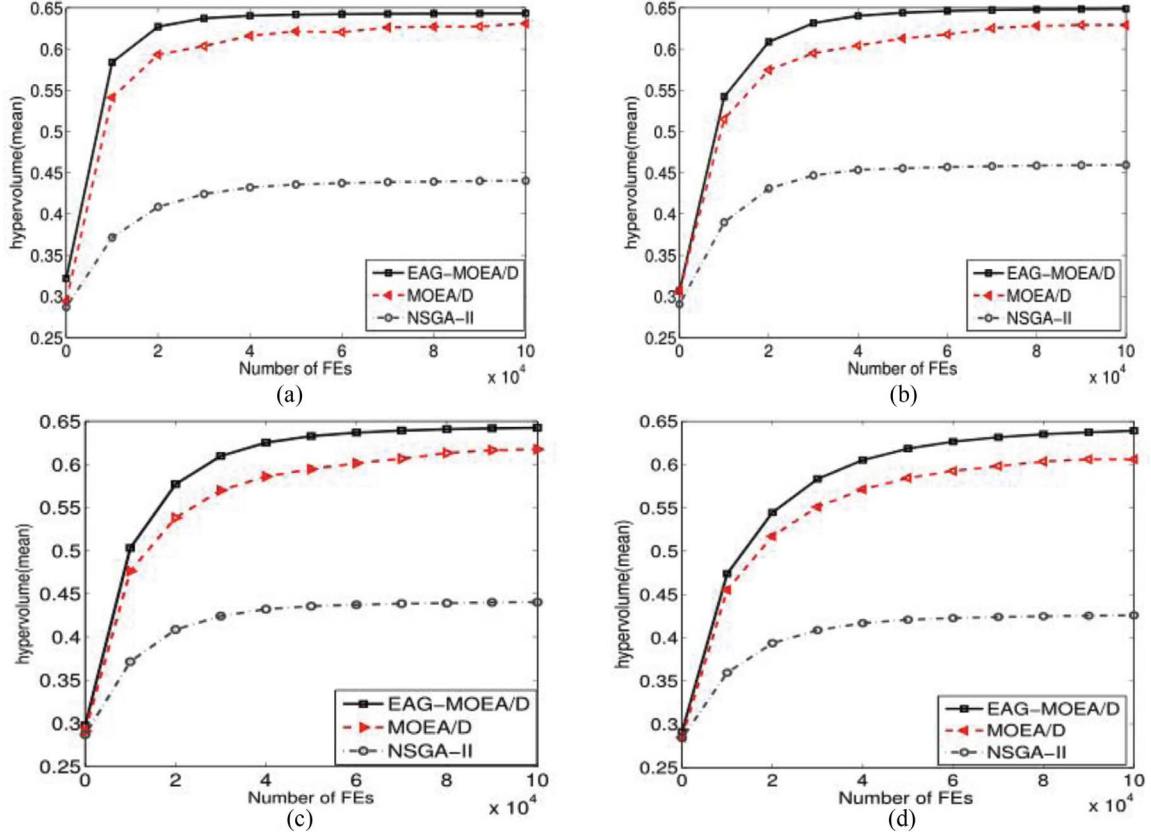


Fig. 4. Convergence graphs in terms of hypervolume (mean) obtained by NSGA-II, MOEA/D, and EAG-MOEA/D on MNRP instances. (a) $Cu - 80/R - 800$. (b) $Cu - 120/R - 1200$. (c) $Cu - 160/R - 1600$. (d) $Cu - 200/R - 2000$.

TABLE II
MEAN AND STANDARD DEVIATION VALUES OF I_H , IGD , AND Δ_p , OBTAINED BY NSGA-II, MOEA/D, AND EAG-MOEA/D ON MNRP INSTANCES

instance		I_H			IGD			Δ_p		
—		EAG-MOEA/D	MOEA/D	NSGA-II	EAG-MOEA/D	MOEA/D	NSGA-II	EAG-MOEA/D	MOEA/D	NSGA-II
$Cu - 30$	mean	0.6340	0.6237	0.5216	0.0021	0.0030	0.3287	0.0037	0.0085	0.3634
	std	1.4577e-4	0.0019	0.0076	0.0010	3.8483e-4	0.0175	0.0020	0.0016	0.0182
$/R - 300$	mean	0.6422	0.6343	0.5551	0.0018	0.0025	0.2575	0.0034	0.0065	0.3027
	std	1.5303e-4	0.0016	0.0087	3.6296e-4	1.8390e-4	0.0182	7.2887e-4	0.00063	0.0191
$Cu - 50$	mean	0.6469	0.6342	0.5061	0.0017	0.0048	0.3502	0.0032	0.0106	0.3975
	std	1.7725e-4	0.0021	0.0071	5.5025e-4	3.9634e-4	0.0191	5.9253e-4	0.0013	0.0215
$Cu - 80$	mean	0.6432	0.6304	0.4774	0.0021	0.0043	0.4000	0.0035	0.0099	0.4465
	std	1.1282e-4	0.0016	0.0059	4.3043e-4	4.3064e-4	0.0131	8.6642e-4	0.0014	0.0147
$/R - 800$	mean	0.6504	0.6338	0.4712	0.0021	0.0053	0.4054	0.0034	0.0120	0.4537
	std	1.3922e-4	0.0034	0.0066	5.3414e-4	9.7000e-4	0.0126	8.0093e-4	0.0034	0.0137
$Cu - 100$	mean	0.6488	0.6302	0.4596	0.0018	0.0064	0.4130	0.0027	0.0139	0.4635
	std	1.1935e-4	0.0041	0.0066	5.2261e-4	0.0012	0.0133	3.3594e-4	0.0042	0.0146
$/R - 1200$	mean	0.6424	0.6175	0.4404	0.0026	0.0106	0.3771	0.0032	0.0205	0.4355
	std	1.7933e-4	0.0047	0.0057	4.9009e-4	0.0026	0.0118	4.3582e-4	0.0072	0.0127
$Cu - 160$	mean	0.6389	0.6111	0.4258	0.0077	0.0162	0.3330	0.0128	0.0280	0.3990
	std	3.8820e-4	0.0074	0.0036	8.7019e-4	0.0037	0.0077	0.0017	0.0091	0.0084

2) Investigation of the contribution of the guide from the external archive in EAG-MOEA/D.

In our experiments, every algorithm is run independently 30 times on a test instance.

A. Parameter Settings

All the algorithms were implemented in MATLAB. Their parameter settings for the MNRP and the MTSP are listed in Table I. The uniform crossover and bit-flip mutation operators

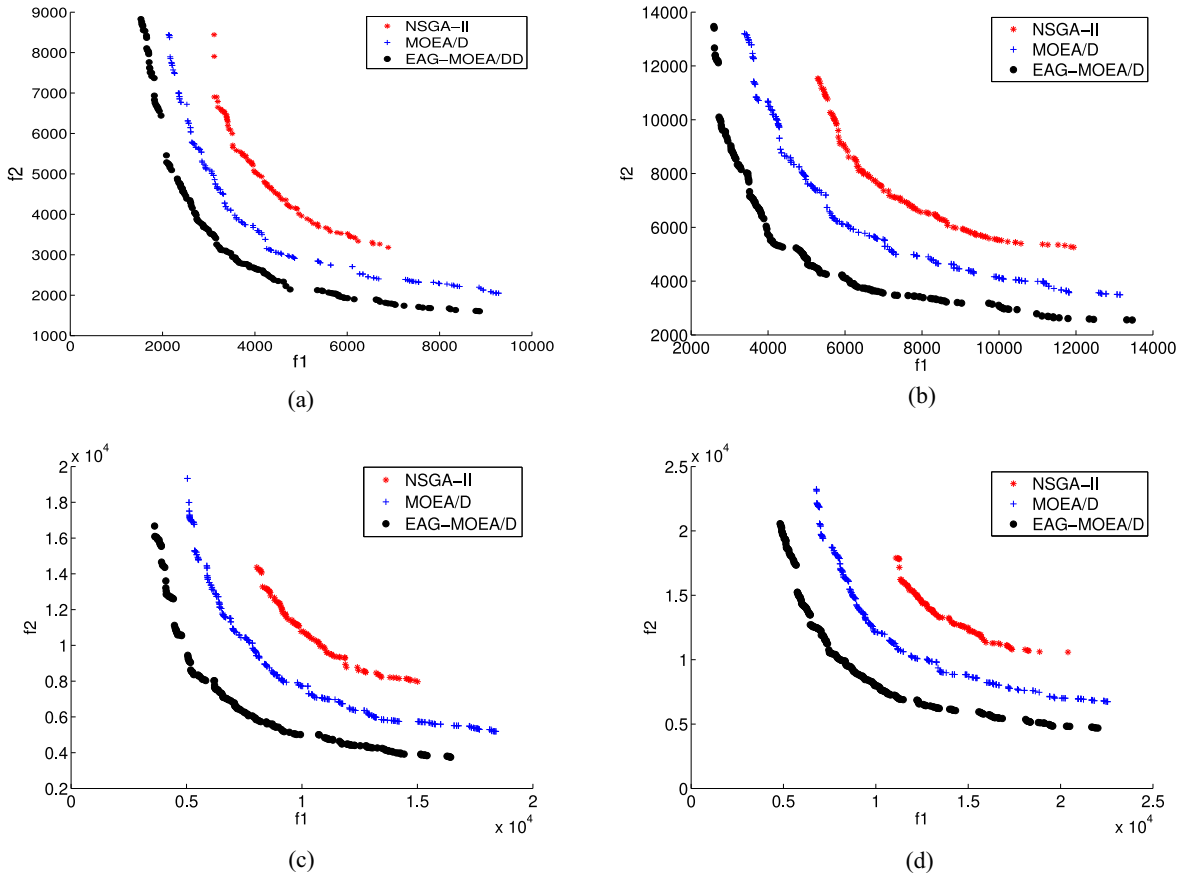


Fig. 5. Final nondominated solutions found by NSGA-II, MOEA/D, and EAG-MOEA/D on four MTSP instances. (a) $c = 200/o = 2$. (b) $c = 300/o = 2$. (c) $c = 400/o = 2$. (d) $c = 500/o = 2$.

TABLE III

MEAN AND STANDARD DEVIATION VALUES OF I_H , IGD , AND Δ_p , OBTAINED BY NSGA-II, MOEA/D, AND EAG-MOEA/D ON MTSP INSTANCES

instance		I_H			IGD			Δ_p		
		EAG-MOEA/D	MOEA/D	NSGA-II	EAG-MOEA/D	MOEA/D	NSGA-II	EAG-MOEA/D	MOEA/D	NSGA-II
$c = 200$ $/o = 2$	mean	6.0221e+7	4.8566e+7	3.9490e+7	396.1971	1.1846e+3	1.9777e+3	496.0815	1.2145e+3	1.9926e+3
	std	6.2366e+6	5.5574e+6	4.7260e+6	76.3018	91.0267	67.2064	91.4330	93.3633	68.0586
$c = 300$ $/o = 2$	mean	1.2625e+8	9.8752e+7	7.7264e+7	657.4949	1.9092e+3	3.2895e+3	845.0963	1.9452e+3	3.3052e+3
	std	1.0901e+7	9.8187e+6	8.5322e+6	73.0384	71.2707	79.7071	122.8488	72.3716	81.0038
$c = 400$ $/o = 2$	mean	2.3067e+8	1.7388e+8	1.2975e+8	913.9647	2.8536e+3	4.8921e+3	1.1463e+3	2.9068e+3	4.9071e+3
	std	1.9932e+7	1.7374e+7	1.4328e+7	111.0198	158.3540	109.4016	149.0998	167.8408	105.9919
$c = 500$ $/o = 2$	mean	3.4198e+8	2.5266e+8	1.7785e+8	1.1795e+3	3.8233e+3	6.7387e+3	1.3767e+3	3.6287e+3	6.5883e+3
	std	2.9319e+7	2.5590e+7	2.0991e+7	118.5997	131.4420	146.7671	172.0705	138.2165	134.4089
$c = 600$ $/o = 2$	mean	4.7920e+8	3.4367e+8	2.2982e+8	1.3839e+3	4.6418e+3	8.5539e+3	1.7014e+3	4.7045e+3	8.5688e+3
	std	3.5339e+7	3.1404e+7	2.2366e+7	144.0895	189.2634	197.0477	182.8116	195.6648	197.2284
$c = 700$ $/o = 2$	mean	6.7972e+8	5.2541e+8	3.5494e+8	1.4148e+3	4.4481e+3	9.1627e+3	1.8081e+3	4.5260e+3	9.1991e+3
	std	4.8865e+7	4.2883e+7	3.2194e+7	118.4250	174.1472	237.9123	192.8772	184.5599	242.9295
$c = 800$ $/o = 2$	mean	9.0322e+8	6.9174e+8	4.5605e+8	1.7655e+3	5.2921e+3	1.1026e+4	2.2474e+3	5.3949e+3	1.1082e+4
	std	6.1014e+7	5.3300e+7	4.1377e+7	190.1330	175.9593	254.7711	273.9604	193.6324	259.0512
$c = 200$ $/o = 3$	mean	3.2951e+11	2.5020e+11	2.8563e+10	921.4977	1.4006e+3	3.0186e+3	1.0468e+3	1.4702e+3	3.0409e+3
	std	9.0125e+9	6.9267e+9	1.2218e+10	62.3554	77.6346	132.7166	60.2810	83.7406	131.5414
$c = 300$ $/o = 3$	mean	7.7457e+11	5.2377e+11	4.4283e+11	1.4124e+3	2.2751e+3	5.0158e+3	1.6019e+3	2.3683e+3	5.0568e+3
	std	2.0881e+10	2.0033e+10	1.9174e+11	80.1275	131.0029	182.0547	100.0332	140.8277	182.7961

were used to generate new solutions for the MNRP test instances. Mutation rates for MNRP is set to $1/n$, where n is the length of a solution (the number of variables in a solution). For the MTSP instances, each candidate solution was represented as a permutation, and the position-based crossover and exchange mutation operators [13] were used for generating new solutions. The crossover rate for NSGA-II was set to 0.8 since we found that this setting was better than 1.

The setting of N weight vectors ($\lambda^1, \dots, \lambda^N$) is controlled by a positive integer parameter H , which specifies the

granularity or resolution of weight vectors, as in [26]. Each individual weight takes a value from

$$\left\{ \frac{0}{H}, \frac{1}{H}, \dots, \frac{H}{H} \right\}.$$

The number of weight vectors is determined by both parameter H and the number of objectives m : $N = C_{H+m-1}^{m-1}$.

The number of function evaluations are assigned to different problems based on their actual convergence conditions as

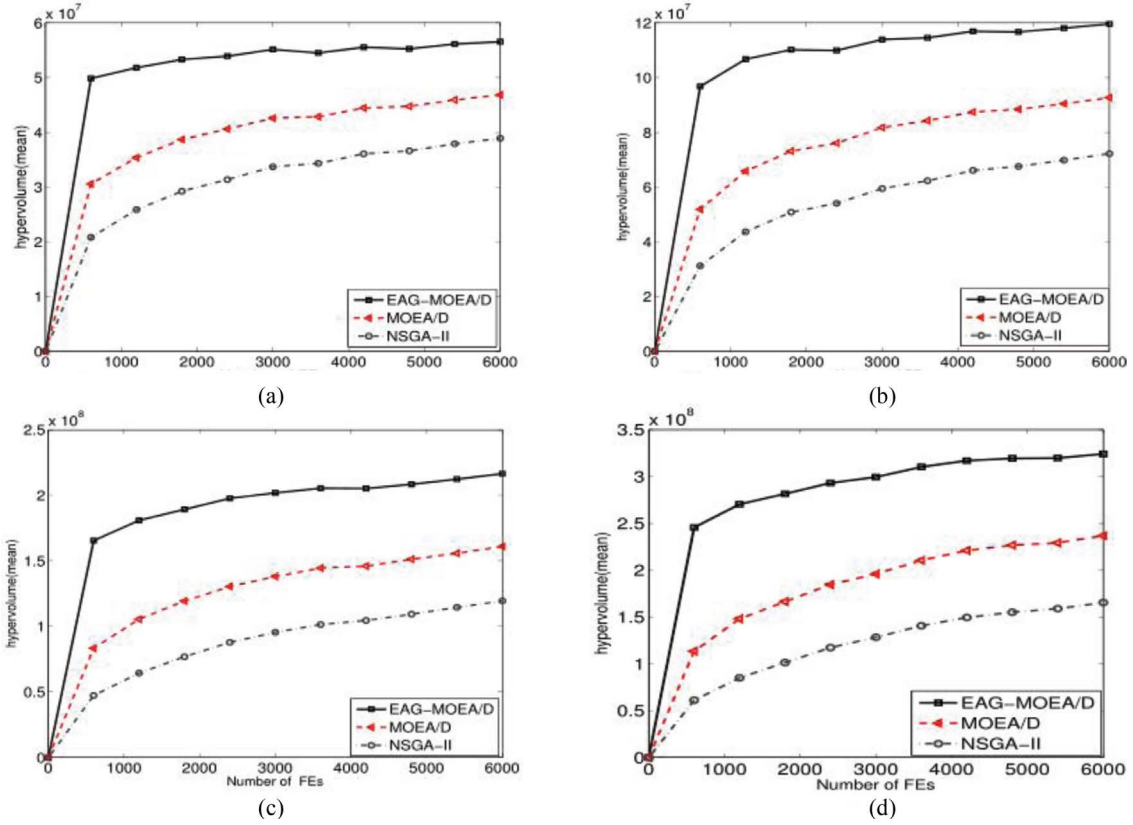


Fig. 6. Convergence graphs in terms of hypervolume (mean) obtained by NSGA-II, MOEA/D, and EAG-MOEA/D on four MTSP instances. (a) $c=200/o=2$. (b) $c=300/o=2$. (c) $c=400/o=2$. (d) $c=500/o=2$.

follows. For the MNRP, the number of function evaluations for all the algorithms is set to 50 000 for $Cu=30/R=300$, $Cu=50/R=500$, and $Cu=50/R=200$; and 100 000 for $Cu=80/R=800$, $Cu=100/R=1000$, $Cu=120/R=1200$, $Cu=160/R=1600$, and $Cu=200/R=2000$. For the MTSP, the number of function evaluation is 400 000 for $c=200/o=2$; 600 000 for $c=300/o=2$, $c=400/o=2$, $c=500/o=2$, $c=600/o=2$, $c=200/o=3$, $c=300/o=3$; and 1 500 000 for $c=700/o=2$ and $c=800/o=2$.

As to the number of learning generations (LGs) in EAG-MOEA/D, we conduct experiments to test its sensitivity. In the experiments, the number of function evaluations is set to 50 000 for $Cu=50/R=500$ and 300 000 for $c=200/o=2$. Other parameters for EAG-MOEA/D are set as in Table I. It is clear to see, from the Fig. 2, that the parameter LGs are not very sensitive for both MNRP and MTSP. In this paper, we set $LGs=8$ for all the problems.

B. Performance Metrics

The following three performance metrics are used in our studies.

- 1) *Hypervolume Indicator (I_H)* [32]: Let $y^* = (y_1^*, \dots, y_m^*)$ be a point in the objective space which is dominated by any Pareto optimal objective vectors. Let P be the obtained approximation to the PF in the objective space. Then the I_H value of P (with regard to y^*) is the volume of the region which is dominated by P and dominates y^* . The higher the hypervolume, the better the approximation. In our experiments,

we set $y^* = (0, 1)$ for MNRP test instances; $y^* = (f_1^{\max}, f_2^{\max})$ for bi-objective MTSP test instances and $y^* = (f_1^{\max}, f_2^{\max}, f_3^{\max})$ for three-objective ones, where f_i^{\max} indicates the maximum value of the i th objective in the obtained nondominated set.

- 2) *Inverted Generational Distance (IGD)* [5]: It measures the average distance from a set of reference points P^* in the PF to the approximation set P . It can be formulated as

$$IGD(P, P^*) = \frac{1}{|P^*|} \sum_{v \in P^*} \text{dist}(v, P) \quad (9)$$

where $\text{dist}(*, *)$ is the Euclidean distance. Ideally, the points in P^* should be uniformly distributed on the PF. However, the true PF is not known in both the MNRP and the MTSP test instances. In our experiments, P^* for a test instance is the set of by all nondominated solutions obtained by all the algorithms in all runs.

- 3) *Averaged Hausdorff Distance (Δ_p)* [21]: It is a recently proposed metric to measure how well P approximates the PF. It also needs P^* , a set of reference points as in the IGD. Δ_p is defined as

$$\Delta_p(X, Y) = \max \left\{ \left[\frac{1}{|P^*|} \sum_{v \in P^*} \text{dist}(v, P)^p \right]^{\frac{1}{p}}, \left(\frac{1}{|P|} \sum_{y \in P} \text{dist}(y, P^*)^p \right)^{\frac{1}{p}} \right\}. \quad (10)$$

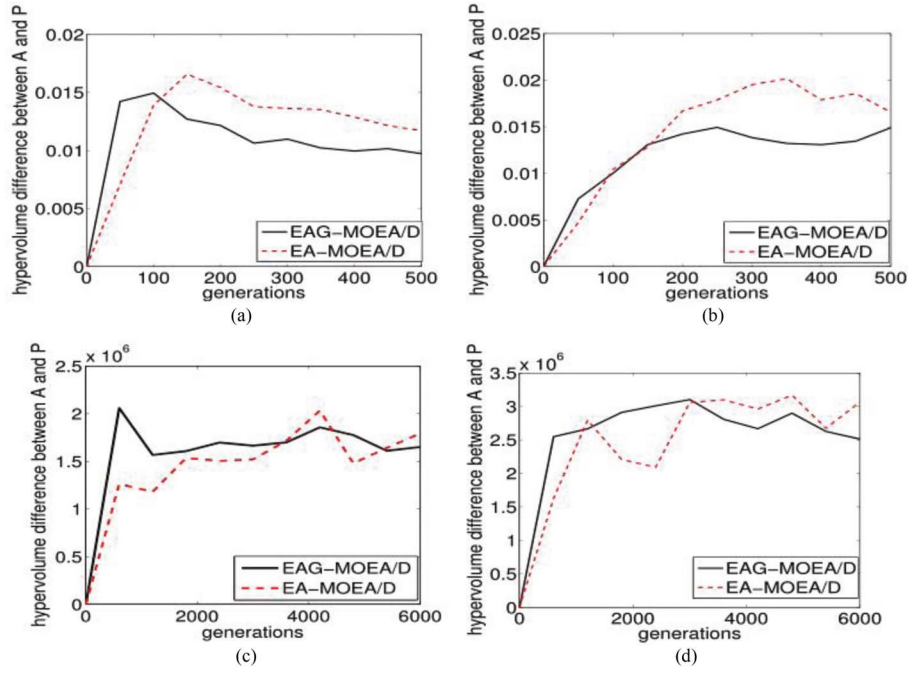


Fig. 7. Evolution of $I_H(A) - I_H(P)$ in EAG-MOEA/D and EA-MOEA/D on four MNRP or MTSP instances, where A is the external archive, and P is the working population. (a) $Cu - 120/R - 1200$. (b) $Cu - 160/R - 1600$. (c) $c - 300/o - 2$. (d) $c - 400/o - 2$.

TABLE IV
MEAN AND STANDARD DEVIATION VALUES OF I_H , IGD , AND Δ_p , OBTAINED BY EAG-MOEA/D
AND EA-MOEA/D ON MNRP AND MTSP INSTANCES

instance		I_H		t-test	IGD		t-test	Δ_p		t-test
-		EAG-MOEA/D	MOEA/D-DRA	p-value	EAG-MOEA/D	MOEA/D-DRA	p-value	EAG-MOEA/D	MOEA/D-DRA	p-value
$Cu - 30$ $/R - 300$	mean	0.6340	0.6254	1.23e-28	0.0021	0.0037	4.4687e-5	0.0037	0.0078	9.9403e-9
	std	1.46e-4	0.0010		0.0010	0.0015		0.0020	0.0016	
$Cu - 50$ $/R - 200$	mean	0.6422	0.6349	1.73e-27	0.0018	0.0023	0.0013	0.0034	0.0057	5.1994e-15
	std	1.53e-4	9.65e-4		3.63e-4	5.57e-4		7.2887e-4	0.0005	
$Cu - 200$ $/R - 500$	mean	0.6469	0.6406	2.92e-26	0.0017	0.0098	2.3147e-8	0.0032	0.0137	3.7033e-11
	std	1.77e-4	0.0015		1.36e-4	0.0057		5.9253e-4	0.0055	
$Cu - 80$ $/R - 800$	mean	0.6432	0.6331	8.38e-27	0.0021	0.0056	3.5088e-15	0.0035	0.0100	1.8124e-17
	std	1.13e-4	0.0014		4.30e-4	0.0011		8.6642e-4	0.0015	
$Cu - 100$ $/R - 1000$	mean	0.6504	0.6373	2.09e-22	0.0021	0.0089	1.7321e-13	0.0034	0.0141	2.0510e-18
	std	1.39e-4	0.0026		5.34e-4	0.0030		8.0093e-4	0.0029	
$Cu - 120$ $/R - 1200$	mean	0.6488	0.6335	3.46e-26	0.0018	0.0171	5.1218e-10	0.0027	0.0216	1.5770e-10
	std	1.19e-4	0.0022		5.23e-4	0.0091		3.3594e-4	0.0107	
$Cu - 160$ $/R - 1600$	mean	0.6424	0.6362	1.07e-27	0.0026	0.0366	1.9884e-23	0.0032	0.0491	3.3233e-22
	std	1.79e-4	0.0026		4.90e-4	0.0062		4.3582e-4	0.0092	
$Cu - 200$ $/R - 2000$	mean	0.6389	0.6156	4.97e-20	0.0077	0.0509	3.2089e-23	0.0128	0.0754	4.2886e-22
	std	3.89e-4	0.0055		8.70e-4	0.0081		0.0017	0.0129	
$c - 200$ $/o - 2$	mean	6.00e+7	5.08e+07	1.42e-26	396.1971	1.04e+3	5.6774e-23	496.0815	1.0925e+3	2.2195e-20
	std	6.20e+6	5.85e+06		92.9339	68.2400		91.4330	106.3723	
$c - 300$ $/o - 2$	mean	1.25e+8	1.03e+8	3.49e-30	657.4949	1.63e+3	6.1219e-31	845.0963	1.7258e+3	4.8766e-24
	std	1.08e+7	9.69e+6		73.0384	90.2135		122.8488	101.3367	
$c - 400$ $/o - 2$	mean	2.26e+8	1.81e+8	2.76e-30	913.9647	2.36e+3	2.8904e-27	1.1463e+3	2.4668e+3	7.4607e-24
	std	2.31e+7	2.11e+7		111.0198	127.2425		149.0998	131.8669	
$c - 500$ $/o - 2$	mean	3.34e+8	2.59e+8	3.07e-31	1.18e+3	3.24e+3	1.6723e-29	1.3767e+3	3.4116e+3	5.2304e-26
	std	3.13e+7	2.82e+07		118.5997	219.2380		172.0705	253.5896	
$c - 600$ $/o - 2$	mean	4.65e+8	3.38e+8	2.79e-32	1.38e+3	4.70e-33	4.70e-33	1.7014e+3	4.2411e+3	1.8265e-29
	std	2.78e+7	2.33e+7		144.0895	193.0609		182.8116	229.8847	
$c - 700$ $/o - 2$	mean	6.74e+8	4.43e+8	7.07e-37	1.41e+3	4.41e+3	1.3827e-39	1.8081e+3	4.3045e+3	2.8784e-37
	std	5.45e+7	4.43e+7		118.4250	241.0844		192.8772	235.6630	
$c - 800$ $/o - 2$	mean	9.03e+8	7.25e+8	1.61e-32	1.77e+3	4.89e+3	4.2072e-33	2.2474e+3	5.2121e+3	2.6421e-30
	std	5.25e+7	4.55e+7		190.1330	206.0012		273.9604	252.6829	
$c - 200$ $/o - 3$	mean	3.29e+11	2.66e+11	1.96e-20	921.4977	1.28e+03	7.5721e-13	1.0468e+3	1.3684e+03	3.0621e-11
	std	9.01e+9	1.37e+10		62.3554	160.2180		60.2810	170.7329	
$c - 300$ $/o - 3$	mean	7.75e+11	6.12e+11	3.2538e-22	1.41e+03	1.95e+03	8.8233e-19	1.6019e+3	2.0976e+3	3.0294e-16
	std	2.09e+10	2.54e+10		80.1275	122.6724		100.0332	125.7981	

In this metric, the higher the value of p , the more outliers are penalized. The value of p is set to 2 in our study. A smaller value of Δ_p indicates better quality of P .

C. EAG-MOEA/D Versus MOEA/D and NSGA-II

EAG-MOEA/D hybridizes NSGA-II and MOEA/D, therefore, it is necessary to compare it with both of them.

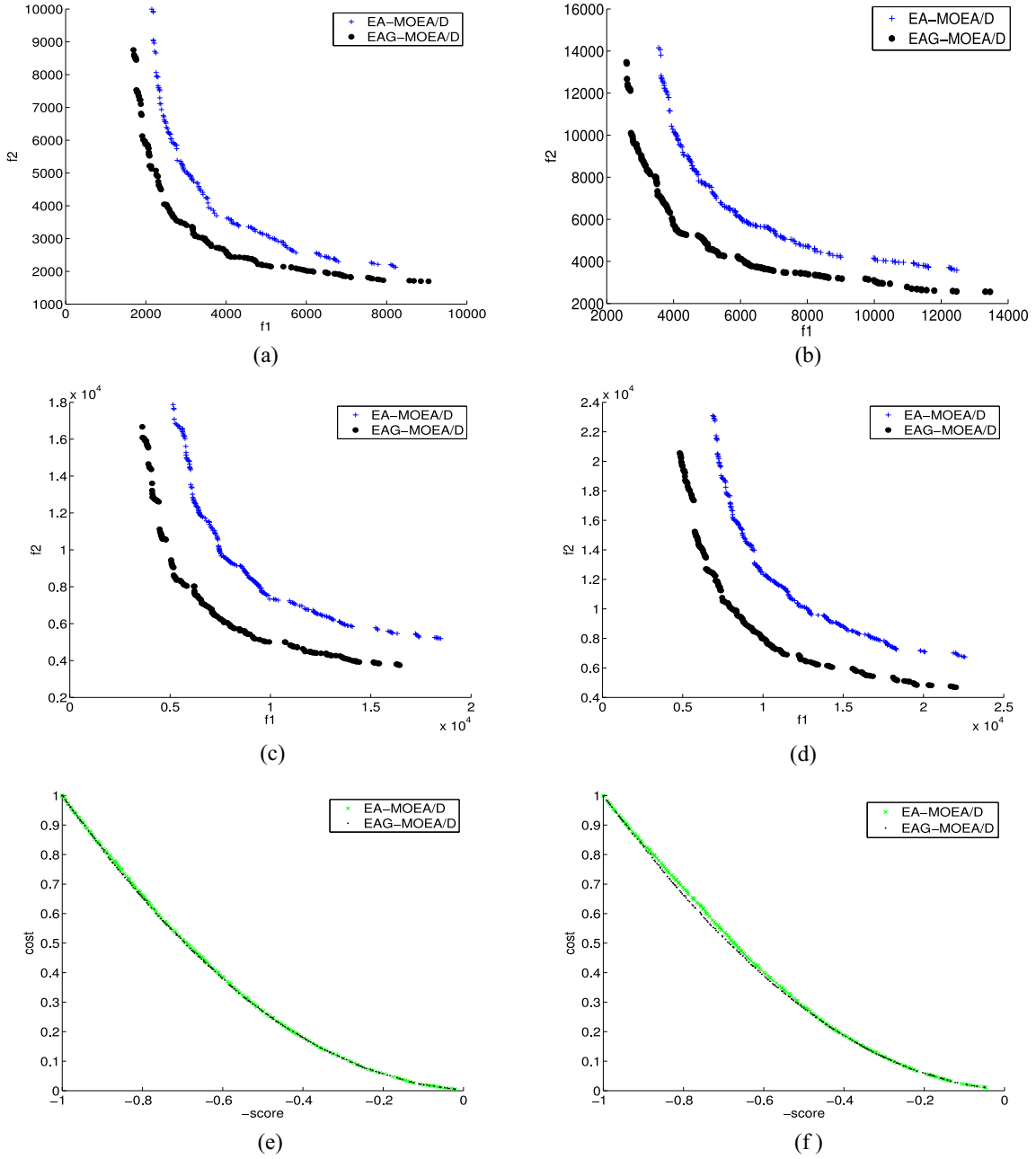


Fig. 8. Final nondominated solutions found by EAG-MOEA/D and EA-MOEA/D on six MNRP or MTSP instances. (a) $c - 200 / o - 2$. (b) $c - 300 / o - 2$. (c) $c - 400 / o - 2$. (d) $c - 500 / o - 2$. (e) $Cu - 120 / R - 1200$. (f) $Cu - 160 / R - 1600$.

Fig. 3 presents the final solution sets with the best (i.e., largest) hypervolume values obtained by each algorithm on MNRP instances over 30 runs. We can make the following observations.

- 1) The final solution sets obtained by the three algorithms are close to the PFs. Therefore, all the three algorithms perform well in terms of convergence.
- 2) NSGA-II produces uniformly spread solutions. However, these solutions are not widely spread.
- 3) MOEA/D produces widely but not uniformly spread solutions. It confirms our analysis in Section II, search on different subproblems with uniformly distributed weight vectors in MOEA/D may not produce a uniformly distributed solutions. This should be because

search efforts on different subproblems may have different contributions. Therefore, allocating the same amount of computational effort to every subproblem as in original MOEA/D may not be very effective and efficient.

- 4) EAG-MOEA/D outperforms both NSGA-II and MOEA/D, it can generate widely and uniformly distributed solutions. This confirms that the guide from the external archive does significantly improve the algorithmic performance.

To further compare the three algorithms, Tables II lists the mean and standard deviation values of I_H , IGD , and Δ_p metrics on eight MNRP instances. Note that a larger I_H indicates better performance while smaller IGD and Δ_p indicate better performance. Clearly, these results confirm that

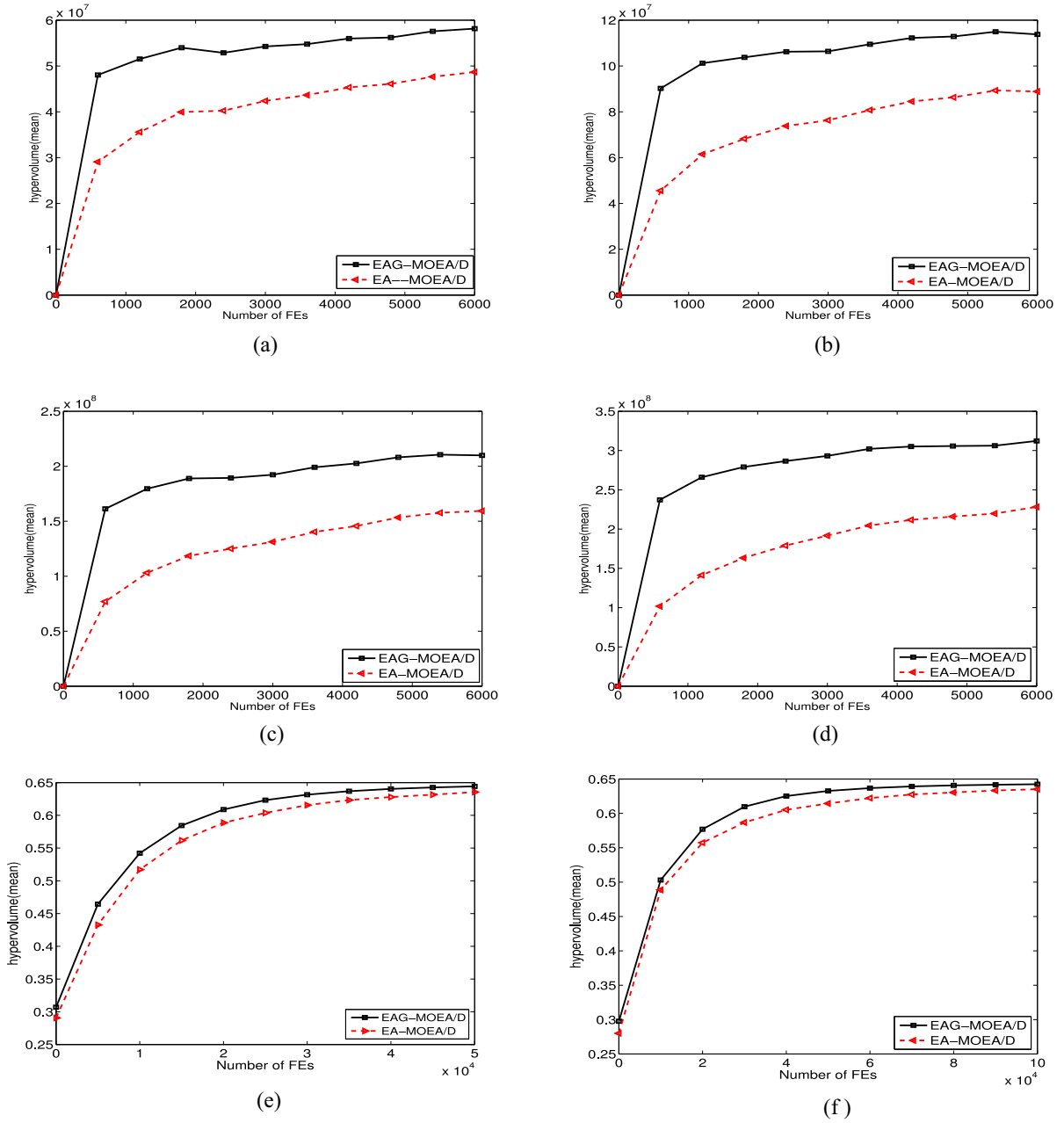


Fig. 9. Convergence graphs in terms of hypervolume (mean) obtained by EAG-MOEA/D and EA-MOEA/D on six MNRP or MTSP instances. (a) $c = 200 / o = 2$. (b) $c = 300 / o = 2$. (c) $c = 400 / o = 2$. (d) $c = 500 / o = 2$. (e) $Cu = 120 / R = 1200$. (f) $Cu = 160 / R = 1600$.

EAG-MOEA/D outperforms the other two algorithms on all eight MNRP test instances in terms of I_H , IGD , and Δ_p metrics. Apparently, NSGA-II is the worst mainly because the solution sets produced by NSGA-II concentrate on a small part of the PFs.

The evolution of the average hypervolume values with the numbers of function evaluations in the three algorithms on MNRP test instances are plotted in Fig. 4. These figure can show both the convergence speed of each algorithm as well as the quality of their final solution sets. It is evident that EAG-MOEA/D performs the best on both aspects.

Fig. 5 plots the final solution set with the largest hypervolume found by each algorithm for each MTSP test instance among 30 independent runs. It is very clear that

EAG-MOEA/D is the best in terms of both convergence and diversity on these test instances. Fig. 6 shows that EAG-MOEA/D converges faster than the two other algorithms. A detailed comparison in Table III also demonstrates the superiority of EAG-MOEA/D in terms of the three metrics on all the MTSP test instances.

D. EAG-MOEA/D Versus MOEA/D-DRA

MOEA/D-DRA [27], a variant of MOEA/D, does computational resource allocations based on the search progress on each subproblem. EAG-MOEA/D uses the external archive to guide the resource allocation among the subproblems. Therefore, it is interesting to compare these two different

TABLE V
MEAN AND STANDARD DEVIATION VALUES OF I_H , IGD , AND Δ_p , OBTAINED BY EA-MOEA/D AND MOEA/D ON MNRP AND MTSP INSTANCES

instance		I_H		t-test	IGD		t-test	Δ_p		t-test
—		EA-MOEA/D	MOEA/D	p-value	EA-MOEA/D	MOEA/D	p-value	EA-MOEA/D	MOEA/D	p-value
$Cu - 30$ / $R - 300$	mean	0.6332	0.6237	1.56e-22	0.0024	0.0030	4.11e-6	0.0044	0.0085	1.0956e-12
	std	2.91 e-4	0.0019		4.75e-4	3.85e-4		9.9865e-4	0.0016	
$Cu - 50$ / $R - 200$	mean	0.6421	0.6343	6.4014e-22	0.0021	0.0025	5.21e-4	0.0039	0.0065	2.2453e-10
	std	1.84e-4	0.0016		5.45e-4	1.84e-4		0.0013	0.00063	
$Cu - 50$ / $R - 500$	mean	0.6451	0.6342	1.3558e-22	0.0020	0.0048	3.88e-26	0.0035	0.0106	1.6318e-23
	std	3.40e-4	0.0021		1.24e-4	3.96e-4		3.6038e-4	0.0013	
$Cu - 80$ / $R - 800$	mean	0.6419	0.6304	4.6969e-26	0.0022	0.0043	1.80e-18	0.0036	0.0099	8.9791e-21
	std	2.73e-4	0.0016		4.48e-4	4.31e-4		9.1971e-4	0.0014	
$Cu - 100$ / $R - 1000$	mean	0.6484	0.6338	1.3458e-20	0.0023	0.0053	5.14e-16	0.0037	0.0120	6.6964e-14
	std	3.07e-4	0.0034		3.10e-4	9.70e-4		5.1083e-4	0.0034	
$Cu - 120$ / $R - 1200$	mean	0.6453	0.6302	1.6890e-18	0.0022	0.0064	1.90e-17	0.0043	0.0139	5.2931e-13
	std	4.20e-4	0.0041		2.63e-4	0.0012		5.2251e-4	0.0042	
$Cu - 160$ / $R - 1600$	mean	0.6362	0.6175	4.6978e-19	0.0044	0.0106	4.15e-14	0.0069	0.0205	1.7659e-11
	std	8.19e-4	0.0047		4.46e-4	0.0026		8.9902e-4	0.0072	
$Cu - 200$ / $R - 2000$	mean	0.6309	0.6111	3.6455e-14	0.0070	0.0162	1.47e-13	0.0098	0.0280	9.0782e-12
	std	0.0008	0.0074		6.20e-4	0.0037		0.0012	0.0091	
$c - 200$ / $o - 2$	mean	4.91e+7	4.86e+7	0.0284	1.19e+3	1.18e+3	0.7974	1.2205e+3	1.2145e+3	0.7856
	std	5.84e+6	5.56e+6		868.240	91.0267		70.3735	93.3633	
$c - 300$ / $o - 2$	mean	1.00e+8	9.86e+7	5.58e-05	1.88e+3	1.91e+3	0.1857	1.9210e+3	1.9452e+3	0.2744
	std	1.09e+7	9.84e+6		80.9255	71.2707		97.3896	72.3716	
$c - 400$ / $o - 2$	mean	1.77e+8	1.74e+8	2.51e-05	2.80e+3	2.85e+3	0.1233	2.8437e+3	2.9068e+3	0.0549
	std	1.79e+7	1.74e+7		98.8201	158.3540		101.9658	167.8408	
$c - 500$ / $o - 2$	mean	2.55e+8	2.53e+8	0.0200	3.86e+3	3.82e+3	0.3510	3.6611e+3	3.6287e+3	0.4521
	std	2.65e+7	2.60e+7		170.9247	131.4420		182.9899	138.2165	
$c - 600$ / $o - 2$	mean	3.45e+8	3.44e+8	0.3254	4.70e+3	4.64e+3	0.1246	4.7629e+3	4.7045e+3	0.1598
	std	3.05e+7	3.14e+7		146.8996	189.2634		160.3904	195.6648	
$c - 700$ / $o - 2$	mean	5.33e+8	5.25e+8	3.0777e-04	4.40e+3	4.45e+3	0.3071	4.4656e+3	4.5260e+3	0.2115
	std	4.21e+7	4.29e+7		167.0535	174.1472		174.3701	184.5599	
$c - 800$ / $o - 2$	mean	6.9864e+8	6.9174e+8	0.0073	5.25e+3	5.29e+3	0.3983	5.3360e+3	5.3949e+3	0.2669
	std	5.61e+7	5.33e+7		167.2215	175.9593		179.5077	193.6324	
$c - 200$ / $o - 3$	mean	2.5391e+11	2.5020e+11	0.0401	1.3966e+3	1.4006e+3	0.8411	1.4524e+3	1.4702e+3	0.3760
	std	6.5726e+09	6.9267e+09		79.4301	77.6346		80.0319	83.7406	
$c - 300$ / $o - 3$	mean	5.3529e+11	5.2377e+11	0.0543	2.2560e+3	2.2751e+3	0.5658	2.3383e+3	2.3683e+3	0.4077
	std	2.0628e+10	2.0033e+10		112.1540	131.0029		121.4289	140.8277	

approaches. In our comparison experiments, the parameter settings of both EAG-MOEA/D and MOEA/D-DRA are the same as in Table I.

Table IV compares EAG-MOEA/D and MOEA/D-DRA on the MNRP and MTSP test instances in terms of I_H , IGD , and Δ_p metrics. Clearly, EAG-MOEA/D is better than MOEA/D-DRA on all the instances. These results are very consistent with our considerations made in Section II. It is evident that the use of the external population for guiding resource allocation does have advantages on some combinatorial MOPs. However, one cannot exaggerate it. Each strategy is biased and has its own range of applications.

E. Effects of External Archive

The external archive is used in EAG-MOEA/D for two purposes. One is to guide the allocation of computational resources during the search, and the other is to be the output of the algorithm as the final solution set to the MOP. To understand the effects of the external archive, we conduct the following comparisons.

1) *MOEA/D Versus MOEA/D With External Archive (EA-MOEA/D)*: By EA-MOEA/D, we mean EAG-MOEA/D in which there is no dynamic allocation of computational resources and the external archive only serves as the final output. We want to study the difference between its external archive and its working population. For this purpose, we

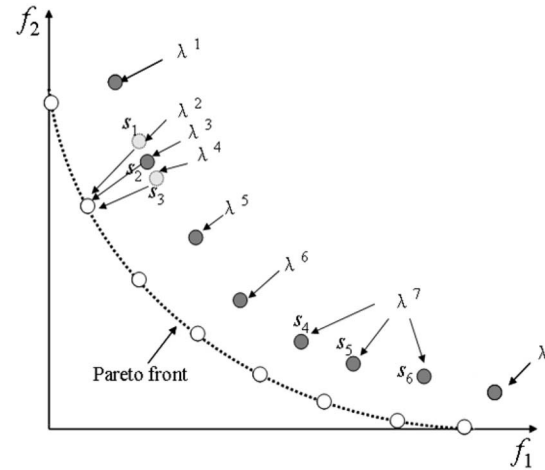


Fig. 10. Illustration of how the external archive A guides the working population P through the diversity information during the optimization process.

have run EA-MOEA/D and MOEA/D with the same parameter settings in Table I on all the test instances.

Table V presents the average values of I_H , IGD , and Δ_p over 30 independent runs. In terms of I_H , it can be observed from this table that EA-MOEA/D performs significantly better than MOEA/D on all the instances except for $c - 600 / o - 2$. On $c - 600 / o - 2$, EA-MOEA/D produces a larger I_H value, however, it is not statistically significant. In terms of IGD

TABLE VI
MEAN AND STANDARD DEVIATION VALUES OF I_H , IGD , AND Δ_p , OBTAINED BY EAG-MOEA/D
AND MOEA/D-DRA ON MNRP AND MTSP INSTANCES

instance		I_H		t-test	IGD		t-test	Δ_p		t-test
—		EAG-MOEA/D	EA-MOEA/D	p-value	EAG-MOEA/D	EA-MOEA/D	p-value	EAG-MOEA/D	EA-MOEA/D	p-value
$Cu - 30$	mean	0.6340	0.6332	9.45e-15	0.0021	0.0024	0.1112	0.0037	0.0044	0.0885
	std	1.46e-4	2.91e-4		0.0010	4.75e-4		0.0020	9.9865e-4	
$/R - 300$	mean	0.6422	0.6421	0.0737	0.0018	0.0021	0.0490	0.0034	0.0039	0.0887
	std	1.53e-4	1.84e-4		3.63e-4	5.45e-4		7.2887e-4	0.0013	
$Cu - 50$	mean	0.6469	0.6451	4.04e-21	0.0017	0.0020	0.0079	0.0032	0.0035	0.0121
	std	1.77e-4	3.40 e-4		1.36e-4	1.26e-4		5.9253e-4	3.6038e-4	
$/R - 500$	mean	0.6432	0.6419	2.9e-20	0.0021	0.0022	0.3731	0.0035	0.0036	0.7142
	std	1.13e-4	2.73e-4		4.30e-4	4.48e-4		8.6642e-4	9.1971e-4	
$Cu - 100$	mean	0.6504	0.6484	4.13e-26	0.0021	0.0023	0.0704	0.0034	0.0037	0.1200
	std	1.39e-4	3.07e-4		5.34e-4	3.10e-4		8.0093e-4	5.1083e-4	
$/R - 1000$	mean	0.6488	0.6453	2.50e-26	0.0018	0.0022	0.0013	0.0027	0.0043	7.7783e-17
	std	1.19e-4	4.20e-4		5.23e-4	2.63e-4		3.3594e-4	5.2251e-4	
$Cu - 120$	mean	0.6424	0.6362	1.07e-27	0.0026	0.0044	1.29e-16	0.0032	0.0069	1.6664e-18
	std	1.79e-4	8.19e-4		4.90e-4	4.46e-4		4.3582e-4	8.9902e-4	
$c - 200$	mean	6.02e+7	4.91e+7	3.89e-31	396.1971	1.19e+3	8.08e-29	496.0815	1.2205e+3	4.3551e-26
	std	6.24e+6	5.84e+6		76.3018	68.2400		91.4330	70.3735	
$/o - 2$	mean	1.26e+8	1.00e+8	1.03e-34	657.4949	1.88e+3	1.32e-34	845.0963	1.9210e+3	3.7187e-26
	std	1.09e+7	9.84e+6		73.0384	80.9255		122.8488	97.3896	
$c - 400$	mean	2.31e+8	1.77e+8	2.10e-35	913.9647	2.80e+3	4.18e-33	1.1463e+3	2.8437e+3	2.4253e-30
	std	1.99e+7	1.79e+7		111.0198	98.8201		149.0998	101.9658	
$/o - 2$	mean	3.42e+8	2.55e+8	3.20e-37	1.18e+3	3.86e+3	2.48e-33	1.3767e+3	3.6611e+3	7.5297e-29
	std	2.93e+7	2.65e+7		118.5997	170.9247		172.0705	182.9899	
$c - 600$	mean	4.79e+8	3.45e+8	5.58e-38	1.38e+3	4.70e+3	2.34e-37	1.7014e+3	4.7629e+3	2.7615e-33
	std	3.53e+7	3.05e+7		144.0895	146.8996		182.8116	160.3904	
$/o - 2$	mean	6.80e+8	5.32e+8	1.08e-35	1.41e+3	4.40e+3	1.74e-38	1.8081e+3	4.4656e+3	1.3120e-34
	std	4.89e+7	4.21e+7		118.4250	167.0535		192.8772	174.3701	
$c - 800$	mean	8.92e+8	6.99e+8	8.50e-36	1.77e+3	5.25e+3	1.90e-35	2.2474e+3	5.3360e+3	4.4802e-30
	std	6.10e+7	5.61e+7		190.1330	167.2215		273.9604	179.5077	
$c - 200$	mean	3.30e+11	2.54e+11	5.54e-25	921.4977	1.40e+03	4.16e-21	1.0468e+3	1.4524e+3	1.1666e-19
	std	9.01e+9	6.57e+9		62.3554	79.4301		60.2810	80.0319	
$/o - 3$	mean	7.75e+11	5.35e+11	2.71e-27	1.41e+3	2.26e+3	8.91e-27	1.6019e+3	2.3383e+3	2.1658e-24
	std	2.09e+10	2.06e+10		80.1275	112.1540		100.0332	121.4289	

and Δ_p , it is clear from the Table V that EA-MOEA/D constantly performs significantly better than MOEA/D on MNRP test instances, and the performances of the two algorithms in terms of IGD and Δ_p are about the same on the MTSP. Based on these results, we can claim that the external population is better or no worse than the working population for approximating the PF. Therefore, it is very reasonable to use the external population to guide the search.

2) *Difference Between the External Population and Working Population in EAG-MOEA/D*: To obtain a good understanding of the difference between the external archive A and the working population P , we also plot the evolution of $I_H(A) - I_H(P)$ in EA-MOEA/D and EAG-MOEA/D on two MNRP and two MTSP test instances, as shown in Fig. 7. Clearly, A is always better than P during the whole search process. Again, it demonstrates that A can guide the search.

3) *EA-MOEA/D Versus EAG-MOEA/D*: The experimental results are presented in Table VI and Figs. 8 and 9. The results in Table VI clearly shows that EAG-MOEA/D always outperforms EA-MOEA/D on almost all the instances except some MNRP ones in terms of all the three metrics. The final solution sets with the best I_H values plotted in Fig. 8 is consistent with the metric values in Table VI.

The evolutions of the I_H values with the numbers of function evolutions in the two algorithms for four MNRP test instances are plotted in Fig. 9. Clearly, EAG-MOEA/D converges faster than EA-MOEA/D.

These results show that the guide from the external population does help improve the algorithm efficiency.

4) *Effects of External Archive on Diversity Maintenance for EAG-MOEA/D*: Compared with MOEA/D-DRA, which uses the progress (convergence) of one single subproblem, EAG-MOEA/D further considers the contributions (determined by both convergence and diversity) of different subproblems to the search process, for dynamic allocation of computational resources. Fig. 10 shows some scenarios of how the external archive A guides the working population P through the diversity information during the optimization process. The hollow circles in Fig. 10 represent the Pareto optimal solutions and the solid circles represent the nondominated solutions in the external archive found by various subproblems within the LGs. In one extreme case, as shown in Fig. 10, subproblems associated with weight λ^2 , λ^3 , and λ^4 lead the search process to one same Pareto approximation solution. Three nondominated solutions (s_1 , s_2 , and s_3) have been found by these three subproblems, respectively. Among them, only one solution (s_2) generated by subproblem associated with λ^3 is able to enter the external archive and the other two solutions (s_1 and s_3) are eliminated by the crowding distance assignment due to the lack of diversity. As a result, subproblems associated with λ^2 and λ^4 contribute no solutions to the external archive. According to (4) and (5), they have very slim chance to be selected to generate new solutions. On the contrary, the subproblem associated with λ^7 leads to three nondominated solutions (s_4 , s_5 , and s_6) that successfully enter the external archive during the optimization process, which means it has a much larger probability to be selected to generate new solutions.

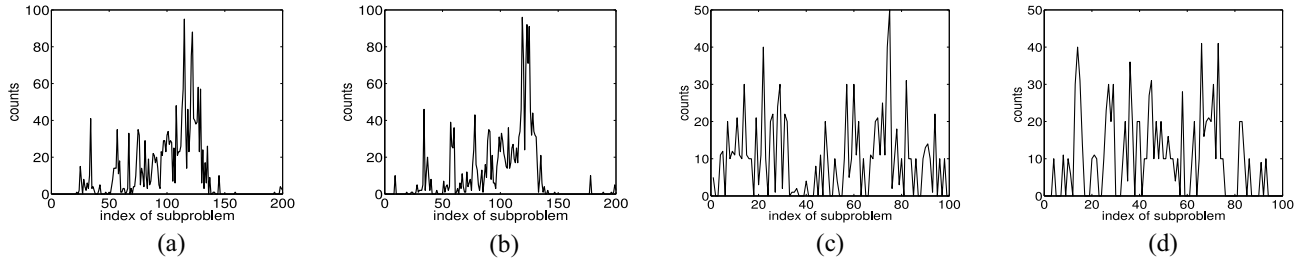


Fig. 11. EA-MOEA/D: counts of nondominated solutions all subproblem contribute to the external archive over the last ten generations of the optimization process for MNRP or MTSP instances. (a) $Cu = 80/R = 800$. (b) $Cu = 100/R = 1000$. (c) $c = 200/o = 2$. (d) $c = 300/o = 2$.

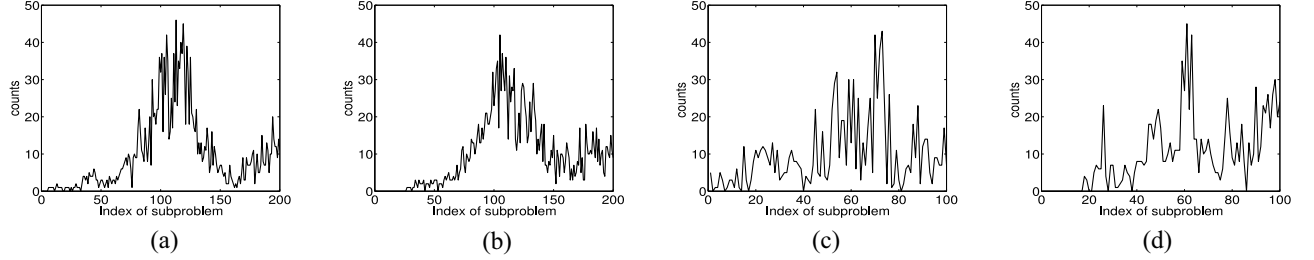


Fig. 12. EAG-MOEA/D: counts of nondominated solutions all subproblem contribute to the external archive over the last ten generations of the optimization process for MNRP or MTSP instances. (a) $Cu = 80/R = 800$. (b) $Cu = 100/R = 1000$. (c) $c = 200/o = 2$. (d) $c = 300/o = 2$.

5) Contributions of Search on Different Subproblems:

One assumption in this paper is that search on different subproblems should make different contributions to the external population. We record the number of the solutions in the final external archive generated by the search on each subproblem in both EA-MOEA/D and EAG-MOEA/D. We plot it in Figs. 11 and 12. It is clear to see that in these two MOEA/D variants, the contribution varies from subproblem to subproblem. This observation confirms our assumption in Section II that the search effort on each subproblem may have different contributions at different search stages.

6) *Effects of Dynamic Resource Allocation on MTSP With Unbalanced Objectives:* To further understand the behavior of EAG-MOEA/D, we further investigate the performance of EAG-MOEA/D on a MTSP instance with unbalanced objectives as follows.

For convenience, we use a three-objective TSP instance as our test functions and further categorize three objective function values in such a way that $f_1(x)$ has 1001 different values (e.g., 0, 1, 2, ..., 1000), $f_2(x)$ has 101 different values (e.g., 0, 10, 20, ..., 1000), and $f_3(x)$ has 11 different values (e.g., 0, 100, 200, ..., 1000).

Thus, in the constructed MTSP, $f_3(x)$ is very difficult to improve; $f_1(x)$ is very easy to improve and the difficulty of $f_2(x)$ is between $f_1(x)$ and $f_3(x)$.

Fig. 13 shows the effects of the dynamic allocation of computational resources on the three-objective TSP described above. It can be observed that the nondominated set obtained by EAG-MOEA/D has better convergence than that of EA-MOEA/D along $f_1(x)$, and the diversity of EAG-MOEA/D is worse than that of EA-MOEA/D along $f_3(x)$. These observations are consistent with our assumptions for the effects of the dynamic allocation of computational resources as follows. In the constructed MTSP, the NSGA-II population will

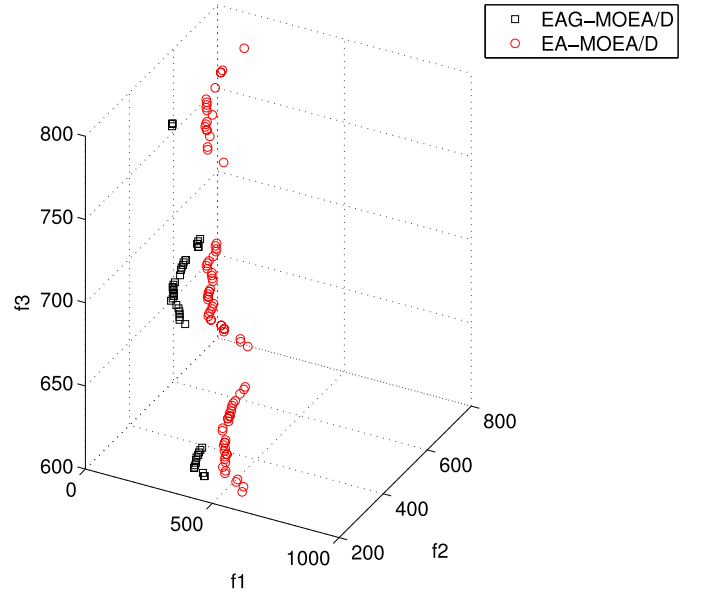


Fig. 13. Effects of the dynamic allocation of computational resources on the constructed unbalanced three-objective TSP, where $f_3(x)$ is very difficult to improve, $f_1(x)$ is very easy to improve, and the difficulty of $f_2(x)$ is between $f_1(x)$ and $f_3(x)$.

be frequently updated toward the direction of $f_1(x)$ because it's easy to improve. As a result, the search of the proposed algorithm will be biased toward the direction of $f_1(x)$.

VI. CONCLUSION

The paper has proposed EAG-MOEA/D which uses an external archive to guide dynamic allocation of computational resources among subproblems. It decomposes a MOP into a number of single optimization subproblems and optimizes them in a collective way. The EAG-MOEA/D framework

keeps two populations—one working population, and one external archive/population. Each subproblem provides an individual solution in the working population. The external archive in EAG-MOEA/D is maintained by the NSGA-II selection. Based on the contribution to the external archive from each subproblem in the search process, computational resources are allocated to each subproblem dynamically. EAG-MOEA/D has been compared with NSGA-II, MOEA/D, and MOEA/D-DRA on two multiobjective combinatorial optimization problems. Experimental results show that EAG-MOEA/D outperforms the other algorithms on these problems. We have also investigated and discussed the effects of the external archive in the search process.

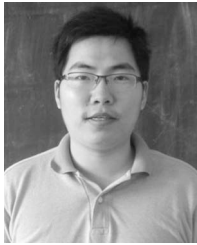
Further work includes investigations of applying EAG-MOEA/D for other optimization problems such as continuous, many objective, and constrained optimization problems. It is also interesting to use other techniques than NSGA-II selection to maintain the external archive.

ACKNOWLEDGMENT

The authors would like to thank Dr. K. C. Tan for his helpful comments and suggestions on the original manuscript.

REFERENCES

- [1] X. Cai and O. Wei, "A hybrid of decomposition and domination based evolutionary algorithm for multi-objective software next release problem," in *Proc. 10th IEEE Int. Conf. Control Autom.*, Hangzhou, China, 2013, pp. 412–417.
- [2] X. Cai, O. Wei, and Z. Huang, "Evolutionary approaches for multi-objective next release problem," *Comput. Informat.*, vol. 31, no. 4, pp. 847–875, 2012.
- [3] P. C. Chang, S. H. Chen, Q. Zhang, and J. L. Lin, "MOEA/D for flowshop scheduling problems," in *Proc. Congr. Evol. Comput. (CEC)*, Hong Kong, Jun. 2008, pp. 1433–1438.
- [4] J. Y. Chia, C. K. Goh, V. A. Shim, and K. C. Tan, "A data mining approach to evolutionary optimization of noisy multi-objective problems," *Int. J. Syst. Sci.*, vol. 43, no. 7, pp. 1217–1247, 2012.
- [5] C. A. C. Coello and N. C. Cortés, "Solving multiobjective optimization problems using an artificial immune system," *Genet. Program. Evol. Mach.*, vol. 6, no. 2, pp. 163–190, Jun. 2005.
- [6] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.
- [7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
- [8] A. Gaspar-Cunha, "A multi-objective evolutionary algorithm for solving traveling salesman problems: Application to the design of polymer extruders," in *Adaptive and Natural Computing Algorithms*, B. Ribeiro, R. F. Albrecht, A. Dobnikar, D. W. Pearson, and N. C. Steele, Eds. Vienna, Austria: Springer, Mar. 2005, pp. 189–193.
- [9] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of multi-objective evolutionary algorithms on many-objective knapsack problems," *IEEE Trans. Evol. Comput.*, to be published.
- [10] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multiobjective evolutionary algorithm," in *Proc. 5th Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, vol. 5467, Nantes, France, Apr. 2009, pp. 438–452.
- [11] A. Kafafy, A. Bounekkar, and S. Bonnevey, "Hybrid metaheuristics based on MOEA/D for 0/1 multiobjective knapsack problems: A comparative study," in *Proc. IEEE Congr. Evol. Comput.*, Brisbane, QLD, Australia, 2012, pp. 1–8.
- [12] A. Konstantinidis, K. Yang, Q. Zhang, and D. Zeinalipour-Yazti, "A multi-objective evolutionary algorithm for the deployment and power assignment problem in wireless sensor networks," *Comput. Netw.*, vol. 54, no. 6, pp. 960–976, 2010.
- [13] P. Larrañaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic, "Genetic algorithms for the travelling salesman problem: A review of representations and operators," *Artif. Intell. Rev.*, vol. 13, no. 2, pp. 129–170, 1999.
- [14] H. Li and D. Landa-Silva, "An adaptive evolutionary multi-objective approach based on simulated annealing," *Evol. Comput.*, vol. 19, no. 4, pp. 561–595, 2011.
- [15] 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.
- [16] 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.
- [17] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, MA, USA: Kluwer Academic, 1999.
- [18] T. Murata, H. Ishibuchi, and M. Gen, "Specification of genetic search directions in cellular multi-objective genetic algorithms," in *Proc. 1st Int. Conf. Evol. Multi-Criterion Optim.*, Zurich, Switzerland, 2001, pp. 82–95.
- [19] W. Peng and Q. Zhang, "Network topology planning using MOEA/D with objective-guided operators," in *Parallel Problem Solving from Nature—PPSN XII*, vol. 7492, Berlin, Germany: Springer, 2012, pp. 62–71.
- [20] W. Peng, Q. Zhang, and H. Li, "Comparison between MOEA/D and NSGA-II on the multi-objective traveling salesman problem," in *Multi-Objective Memetic Algorithms*, vol. 171, Berlin, Germany: Springer, 2009, pp. 309–324.
- [21] O. Schütze, X. Esquivel, A. Lara, and C. A. C. Coello, "Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 4, pp. 504–522, Aug. 2012.
- [22] 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, Appl. Rev.*, vol. 42, no. 5, pp. 682–691, Sep. 2012.
- [23] P. Wang, M. Emmerich, R. Li, K. Tang, T. Baeck, and X. Yao, "Convex hull-based multi-objective genetic programming for maximizing receiver operating characteristic performance," *IEEE Trans. Evol. Comput.*, to be published.
- [24] P. Wang, K. Tang, T. Weise, and X. Yao, "Multiobjective genetic programming for maximizing ROC performance," *Neurocomputing*, vol. 125, pp. 102–118, Feb. 2014.
- [25] G. G. Yen and Z. He, "Performance metrics ensemble for multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 131–144, 2014.
- [26] 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.
- [27] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," *School Comput. Sci. Electron. Eng., Univ. Essex, Colchester, U.K., Working Rep. CES-491*, Feb. 2009.
- [28] Y. Zhang, M. Harman, and S. A. Mansouri, "The multi-objective next release problem," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Jul. 2007, pp. 1129–1136.
- [29] S. Z. Zhao, P. N. Suganthan, and Q. Zhang, "Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes," *IEEE Trans. Evol. Comput.*, vol. 16, no. 3, pp. 442–446, Jun. 2012.
- [30] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature—PPSN VIII* (Lecture Notes in Computer Science), vol. 3242, X. Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervós, J. A. Bullinaria, J. E. Rowe, P. Tiño, A. Kabán, and H.-P. Schwefel, Eds. Berlin, Germany: Springer, Sep. 2004, pp. 832–842.
- [31] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," in *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, Eds. Barcelona, Spain: CIMNE, 2002, pp. 95–100.
- [32] 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.



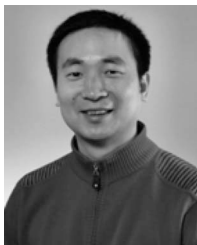
Xinye Cai (M'10) received the B.Sc. degree in information engineering from Huazhong University of Science and Technology, Wuhan, China, in 2004; the M.Sc. degree in electronic engineering from University of York, Heslington, U.K., in 2006; and the Ph.D. degree in electrical engineering from Kansas State University, Manhattan, KS, USA, in 2009.

He is an Associate Professor with the Department of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include evolutionary computation, optimization, data mining, and their applications.



Yexing Li is currently working toward the M.Sc. degree in computer science from the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China.

Her research interests include evolutionary computation and multiobjective optimization.



Zhun Fan (M'04–SM'10) received the B.S. and M.S. degrees in control engineering from Huazhong University of Science and Technology, Wuhan, China, in 1995 and 2000, respectively, and the Ph.D. degree in electrical and computer engineering from Michigan State University, East Lansing, MI, USA, in 2004.

He is currently the Director of the Guangdong Provincial Key Laboratory of Digital Signal and Image Processing, and a Full Professor and the Head of the Department of Electronic Engineering with Shantou University, Shantou, China. Prior to joining Shantou University, he was an Assistant Professor, and then an Associate Professor at the Technical University of Denmark, Kongens Lyngby, Denmark. He was also with the BEACON Center for Study of Evolution in Action, East Lansing, MI, USA, as a Researcher. His current research interests include intelligent control and robotic systems, computational intelligence, evolutionary computation, design automation, and optimization of mechatronic systems.



Qingfu Zhang (M'01–SM'06) received the B.Sc. degree in mathematics from Shanxi University, Taiyuan, China, in 1984, and the M.Sc. degree in applied mathematics and the Ph.D. degree in information engineering from Xidian University, Xi'an, China, in 1991 and 1994, respectively.

He is a Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong. He is also a Professor, on leave, from the School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K., and a Changjiang Visiting Chair Professor with Xidian University. He is currently leading the Metaheuristic Optimization Research Group with City University of Hong Kong. His research interests include evolutionary computation, optimization, neural networks, data analysis, and their applications. He holds two patents and has authored several research publications.

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