A Multi-phase Adaptively Guided Multiobjective Evolutionary Algorithm Based on Decomposition for Travelling Salesman Problem

Ning Zhang
School of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics
Jiangsu,China
Email: zhangning_0920@163.com

Xinye Cai
School of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics
Jiangsu,China
Email: xinye@nuaa.edu.cn

Zhun Fan
School of Engineering
Shantou University
Guangdong,China
Email: zfan@stu.edu.cn

***Abstract*—In this paper, a multi-phase strategy for dynamic resource allocation is proposed for some special optimization problems where the evolutionary process cannot be explicitly divided into two phases, under the decomposition-based multiobjective evolutionary optimization framework. Based on the evolutionary status, a switching mechanism is adopted to adaptively use either convergence or diversity information in the external archive, to guide the evolutionary search in the working population. The proposed algorithm is compared with six well-known multiobjective evolutionary algorithms on multiobjective travelling salesman problem (MOTSP). Experimental results show that our proposed algorithm performs better than other compared algorithms.**

I. Introduction

In real-world applications, many optimization problems are, by nature, multiobjective, where the objectives are usually conflicting with each other. Different from a single objective optimization problem where a single optimal solution exists, multiobjective optimization problems (MOPs) aim to obtain a set of Pareto-optimal solutions, which represent the best tradeoff candidate solutions. The Pareto-optimal solutions can be very helpful for decision makers to understand the tradeoff relationship among different objectives and choose their preferred solutions. Over the past decades, multiobjective evolutionary algorithms (MOEAs) have been widely recognized as a major methodology to approximate the Pareto-optimal solutions [1], [2], [3].

Selection, which plays a key role in the performance of MOEAs, is desirable to balance between the distance of solutions towards the Pareto-optimal solutions and the uniform spread of solutions [4], [5]. The former is usually called convergence and latter is called diversity in the field of MOEAs.

Based on different selection mechanisms, MOEAs can be further divided into the domination-based (e.g., [6], [7], [8]), the indicator-based (e.g., [9], [10], [11]) and the decomposition-based MOEAs (e.g., [12], [13], [14], [15]). Among all the MOEAs, multiobjective evolutionary algorithm based on decomposition (MOEA/D) [13], which can be regarded as a generalization of cMOGA [16], becomes a representative of decomposition-based MOEAs. MOEA/D decomposes a MOP into a set of single-objective (or simple multiobjective) subproblems, by aggregating all the objectives with different linear or nonlinear weighted aggregation functions and solves all the subproblems in parallel. In MOEA/D, each solution is associated with a subproblem, and two subproblems are called neighbors if their weight vectors are close to each other. By using information from the neighboring subproblems, MOEA/D is able to conduct more effective search [13]. The diversity of MOEA/D is implicitly achieved by specifying a wide spread of the directions in the objective space.

A considerable number of MOEA/D variants have been proposed to improve MOEA/D in various aspects (e.g., [17], [18], [19], [20], [21], [22], [23], [24]). For example, it has been observed in the experiments that some parts of the PF in a MOP can be more difficult to approximate than others [17], [25], [26]. Therefore, it is necessary to provide different computational resources to different subproblems during the optimization process. Under this circumstance, Zhang et.al. [17] proposed MOEA/D-DRA, where a dynamic resource allocation strategy based on the utility function (convergence information) is adopted for allocating different computational resources to different subproblems. Similar to MOEA/D-DRA, Zhou et.al. [26] proposed a generalized resource allocation (GRA) strategy for dynamic resource allocation for MOEA/D. Each subproblem is selected to invest according to a probability of improvement vector, maintained and updated by an offline/online measurements of the subproblems' hardness. In [19], [27], both of a decomposition-based working population and a domination-based external population are used. The proposed algorithm, called EAG-MOEA/D [19] allocates different computational resources to different subproblems based on the both the convergence and diversity information extracted from the external archive, to guide the search in the working population. Very recently, an extension of EAG-

978-1-5090-0623-6/16/$31.00

MOEA/D, two-phase external archive guided multiobjective evolutionary algorithm (2EAG-MOEA/D) [27] was proposed. Different from EAG-MOEA/D, the evolutionary process is explicitly divided into two phases: convergence and diversity phase. 2EAG-MOEA/D utilizes either convergence or diversity information, depending on the evolutionary status.

Based on our recent experimental studies, for some special MOPs, e.g., multiobjective travelling salesman problem (MOTSP), the evolutionary status cannot explicitly be divided into two phases. Therefore, it is natural to extend the evolutionary statuses into multiple phases. In this paper, we propose a multi-phase adaptively guided multiobjective evolutionary algorithm based on decomposition (AG-MOEA/D). Different from EAG-MOEA/D or 2EAG-MOEA/D, AG-MOEA/D adaptively switches between using convergence or diversity information to guide the search in MOEA/D. The experimental studies have shown that AG-MOEA/D outperforms both EAG-MOEA/D and 2EAG-MOEA/D in the MOTSP instances.

The rest of the paper is organized as follows. Section II explains the background knowledge and motivations of AG-MOEA/D. Section III details the procedures of AG-MOEA/D. Experimental studies and discussions are presented in Section IV and V, where we compare AG-MOEA/D with two classical MOEAs, NSGA-II [8], MOEA/D [13], and four state-of-the-art MOEAs, MOEA/D-DRA [17], MOEA/D-GRA [26], EAG-MOEA/D [19] and 2EAG-MOEA/D [27]. Finally, Section VI concludes the paper.

II. Backgrounds and Motivations

A. Multiobjective optimization problem

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

$$\begin{aligned} \text{minimize} \quad & F(x) = (f_1(x), \ldots, f_m(x)) \\ \text{subject to} \quad & x \in \Omega. \end{aligned} \tag{1}$$

where $x = (x_1, x_2, \ldots, x_n)$ is a n-dimensional decision variable, and Ω is the decision space; $F : \Omega \to R^m$ consists of m objective functions, mapping decision space Ω to objective space R^m.

Definition 1: u is said to *dominate* v, denoted by $u \preceq v$, if and only if $u_i \leq v_i$ for every $i \in \{1, \ldots, m\}$ and $u_j < v_j$ for at least one index $j \in \{1, \ldots, m\}$.

Definition 2: A solution $x^* \in \Omega$ is *Pareto-optimal* if there is no other solution in Ω can dominate it.

Definition 3: The set of all the Pareto-optimal solutions is called the *Pareto set* (*PS*).

Definition 4: The set of all the Pareto-optimal objective vectors is the *Pareto front* (*PF*).

B. Motivation

2EAG-MOEA/D, which explicitly divides the evolutionary process into the (first) convergence phase and (second) diversity phase, is proposed in [27]. At the beginning, only the convergence information is used to guide the search in the working population. A utility, the relative decrease of the scalarizing function value for each subproblem over the past learning period, is used to determine the activation of the switch from convergence to diversity phase. Once the utility becomes less than a preset value, the algorithm considers that the evolutionary process steps into the diversity phase and diversity information is used to guide the search. Obviously, an underlying assumption in 2EAG-MOEA/D is that the convergence is much more important at the first stage; while the diversity is much more important at the second stage.

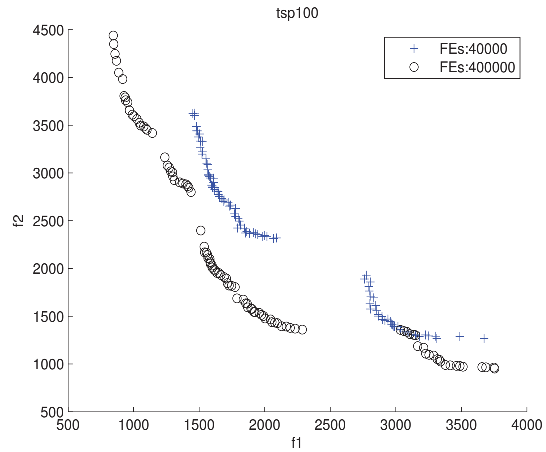


Fig. 1: The non-dominated solutions obtained by 2EAG-MOEA/D on $c - 100/o - 2$ instance at (a) 40,000FEs, (b) 400,000FEs.

2EAG-MOEA/D has shown its advantages in problems, such as multiobjective software next release problem [27]. However, for some other problems, such as MOTSP, we find it is not reasonable to divide the evolutionary status into just two phases. Fig. 1 shows the non-dominated solutions obtained by 2EAG-MOEA/D at 40,000 and 400,000 function evaluations (FEs) respectively on a MOTSP instance. After 40,000 FEs, 2EAG-MOEA/D assumes the evolutionary process is already in the last diversity phase. However, the population keeps converging and spreading until 400,000 FEs, which indicates both convergence and diversity play an equally important role from 40,000 to 400,000 FEs.

Based on the observations in Fig. 1, we propose a multi-phase MOEA, which divides the evolutionary status into multiple phases. In other words, the proposed algorithm is able to switch between convergence and diversity phase multiple times and uses the corresponding information to guide the search in MOEA/D.

III. The Proposed Algorithm

The proposed algorithm, multi-phase adaptively guided multiobjective evolutionary algorithm (AG-MOEA/D), is based on MOEA/D-DE. Several approaches can be used for decomposition [28]. Weighted sum approach is used for simplicity, as follows.

$$g^{ws}(x|\lambda^i) = \sum_{j=1}^{m} \lambda_j^i f_j(x), \tag{2}$$

where $\lambda^i = (\lambda_1^i, \ldots, \lambda_m^i)$ is the weight vector of the i-th subproblems, and $f(x) = (f_1^i(x), \ldots, f_m^i(x))$ is the objective vector of the solution x.

The algorithm flow is given as follows:

Input:

1) A working population P;
2) An external population A;
3) N: the number of subproblems; the population size of P or A;
4) $\lambda^1, \ldots, \lambda^N$: a set of of N weight vectors;
5) T: the size of the neighborhood of each subproblem;
6) n_r: the maximal number of solutions replaced by each child solution;
7) δ: the probability that parent solutions are selected from the neighborhood;
8) LG: learning generations;
9) $phase$: 0 represents the convergence phase and 1 represents the diversity phase;

Output: A set of non-dominated solutions;

Step 1: Initialization:

Step 1.1 For each $i = 1, \ldots, N$, select the neighbors for i-th subproblem based on the Euclidean distance. Set $B(i) = \{i_1, \ldots, i_T\}$, where $\lambda^{i_1}, \ldots, \lambda^{i_T}$ are the T closest weight vectors to λ^i.

Step 1.2 Generate an initial population P : $\{x^1, \ldots, x^N\}$ randomly, and make $A = P$.

Step 1.3 Set $gen = 0$.

Step 2: Reproduction:

Step 2.1 Set $j = 1$.

Step 2.2 Select i-th subproblem based on the probability pro by using Roulette wheel selection.

Step 2.3 Obtain the neighboring solutions as the mating pool for i-th subproblem:

$$R(j) = \begin{cases} B(i) & if(rand < \delta), \\ 1 \ldots, N & otherwise. \end{cases} \tag{3}$$

Step 2.4 Select two indexes k and l from $R(j)$ randomly.

Step 2.5 Generate a new solution y_j for i-th subproblem from x_k and x_l by the position based crossover operators [29] and exchange mutation operators [29].

Step 2.6 $j = j + 1$.

Step 2.7 If $j > N$, go to **Step 3**; otherwise go to **Step 2.2**.

Step 3: Updating solutions:

Step 3.1 For each $j \in \{1, \ldots, N\}$, do the following:

1) Set $q = 0$;
2) If $q = n_r$ or $R(j)$ is empty, go to **Step 3.2**.
3) Randomly pick an index m from $R(j)$, set $x_m = y_j$ if $g^{ws}(y_j|\lambda^m) \leq g^{ws}(x_m|\lambda^m)$.
4) Remove index m from $R(j)$ and go to 2).

Step 3.2 $C = \{y_1, \ldots, y_N\} \bigcup A$.

Step 3.3 $(F_1, F_2, \ldots)$ =non-dominated sorting(C) [8].

Step 3.4 Set $A = \emptyset, t = 1$, do the following:

1) If $|A \bigcup F_t| > N$, go to **Step 3.5**.
2) $A = A \bigcup F_t$, $t = t + 1$ and go to 1).

Step 3.5 If $|A| = N$, go to **Step 4**; otherwise do the following:

1) Calculate the minimum distance for each solution in F_t to A.
2) The solution in F_t with the maximum distance is moved from F_t to A.
3) If $|A| = N$, go to **Step 4**; otherwise go to 1).

Step 4: Calculating subproblem-selection-probability:

Step 4.1 For each subproblem $i \in \{1, \ldots, N\}$, calculate the convergence information $con_{i,gen}$.

Step 4.2 For each subproblem $i \in \{1, \ldots, N\}$, calculate the diversity information $div_{i,gen}$.

Step 4.3 According to (4) or (5), if $\Delta_u > 0.005$ or $\Delta_n \neq 0$, set $phase = 0$; otherwise set $phase = 1$.

Step 4.4 If $phase = 0$, calculate pro based on (6)(7); else, calculate pro based on (6)(8).

Step 4.5 $gen = gen + 1$;

Step 5: Termination:

Step 5.1 If stopping criteria are satisfied, terminate the algorithm and output A. Otherwise, go to **Step 2**.

More details of Steps 1-4 are given as follows:

A. Initialization

Each weight vector λ^i in $(\lambda^1, \ldots, \lambda^N)$ takes values from

$$\{\frac{0}{H}, \frac{1}{H}, \ldots, \frac{H}{H}\}.$$

where H is a user-defined parameter, $\sum_{j=1}^{m} \lambda_j^i = 1$, m is the number of the objectives. The number of the weight vectors N is determined by H and m ($N = C_{H+m-1}^{m-1}$). In this paper H is set to be 99 for 2-objective the instance and 13 for 2-objective the instance. Each weight vector has a neighborhood which consists of T nearest weight vectors.

The initial working population $P = \{x^1, \ldots, x^N\}$ is randomly sampled from the decision space. The external population A is initialized with P.

B. Reproduction

The position based crossover operators [29] and exchange mutation operators [29] are used as the reproduction operators. A total number of N solutions, $Y = \{y_1, \ldots, y_N\}$ are generated.

C. Updating solutions

The working population is updated in Step 3.1. Each new solution at most replaces its n_r neighboring solutions. Step 3.2-3.5 is used to update the external archive. The non-dominated sorting [8] and L_p-norm-based distance [30] is adopted to be the select promising solutions form the combined population. In this paper p is set to be $1/m$, where m is the number of objectives.

D. Calculating subproblem-selection-probability

In Step 4.1, the number of successful solutions for i-th subproblem at gen-th generation $con_{i,gen}$ is calculated as the convergence information. Note that a solution is called successful if it dominates at least one solution in the external archive and it keeps in the external archive at the end of the gen-th generation. In Step 4.2, $div_{i,j,gen}$ is defined as the minimum L_p-norm ($p = 1/m$) of the j-th solution in the external archive generated by the i-th subproblem to all the other solutions in the external archive, at the gen-th generation; and $div_{i,gen}$, which indicates the average of $div_{i,j,gen}$, is considered as the diversity information.

AG-MOEA/D adopts either one of the following two mechanisms to determine whether the evolution is in the convergence or diversity phase. One mechanism, termed WP, is based on the convergence information extracted from the working population and the other one, termed EA, is based on the convergence information from the external archive.

The former mechanism uses the average of the utility values for all the subproblems (Δ_u) to determine whether the evolution is in the convergence phase as follows.

$$\Delta_u = \frac{1}{N}\sum_{i=1}^{N}\frac{g^{ws}(t_{i,old}|\lambda^i) - g^{ws}(t_{i,new}|\lambda^i)}{g^{ws}(t_{i,old}|\lambda^i)} \quad (4)$$

In (4), $t_{i,old}$ is the best solution for i-th subproblem in $(gen - LG + 1)$-th generation and $t_{i,new}$ is the current best solution for i-th subproblem. If the utility value is greater than a preset value (0.005), the evolution is considered to be in the convergence phase; otherwise it is considered to be in the diversity phase.

Alternatively, the latter mechanism uses the convergence information extracted from the external archive (Δ_n) to determine whether the evolution is in the convergence phase. More specifically, Δ_n is defined as the sum of $con_{i,gen}$ for all the subproblems over the last LG generations. Once $\Delta_n \neq 0$, the evolution is considered to be in the convergence phase; otherwise, it is in the diversity phase.

$$\Delta_n = \sum_{G=gen-LG+1}^{gen}\sum_{i=1}^{N} con_{i,G} \quad (5)$$

For convenience, AG-MOEA/D with the former mechanism is called AG-MOEA/D (WP) and AG-MOEA/D with the latter mechanism is called AG-MOEA/D (EA).

The subproblem-selection-probability-vector pro is updated as follows:

$$pro_{i,gen} = \frac{D_{i,gen}}{\sum_{i=1}^{N} D_{i,gen}}, (i = 1, 2, \ldots, N, gen \geq LG). \quad (6)$$

In the convergence phase, $D_{i,gen}$ above is defined as follows:

$$D_{i,gen} = \frac{\sum_{G=gen-LG+1}^{gen} con_{i,G}}{\sum_{i=1}^{N}\sum_{G=gen-LG+1}^{gen} con_{i,G}} + \epsilon. \quad (7)$$

$\epsilon = 0.005$ is used to avoid that the probabilities of any subproblems are zeros. Apparently the subproblems, produced more successful solutions before, have more opportunity to obtain new solutions; In the diversity phase, $D_{i,gen}$ is calculated as (8).

$$D_{i,gen} = \frac{\sum_{G=gen-LG+1}^{gen} div_{i,G}}{\sum_{i=1}^{N}\sum_{G=gen-LG+1}^{gen} div_{i,G}} + \epsilon. \quad (8)$$

IV. Experimental Studies

A. Test problems

We consider one of the NP-hard combinatorial problems–multiobjective travelling salesman problem (MOTSP). Given a network $L = (V, C)$,where $V = v_1, v_2, \ldots, v_n$ represents a set of n cities and $C = c_k : k \in 1, 2, \ldots, m$ is formed by m distance/cost matrices between cities ($c_k : V \times V$). In this paper, the m distance/cost matrices are uncorrelated to each other. Each solution is a sequence of cities e which can form a Hamiltonian cycle. The i-th objective in the MOTSP is to minimize:

$$f_i(x) = \sum_{j=1}^{n} c_i(e(j), e(j+1)) \quad (9)$$

A MOTSP with n cities and m objectives is recorded as $c - n/o - m$ in this paper. We consider 7 test instances of MTSP with two objectives, which include $c - 100/o - 2$, $c - 200/o - 2$, $c - 300/o - 2$, $c - 400/o - 2$, $c - 500/o - 2$, $c - 600/o - 2$, $c - 700/o - 2$, and 2 test instances with three objectives $c - 100/o - 3$ and $c - 200/o - 3$.

B. Performance Metrics

The following performance metrics are used in this paper:

1) Hypervolume indicator: The Hypervolume metric (HV)[31] is a popular indicator due to good theoretical properties, calculating the volume of the region between the obtained approximation P and a reference point $y^* = (y_1^*, \ldots, y_m^*)$ in the objective space as follows:

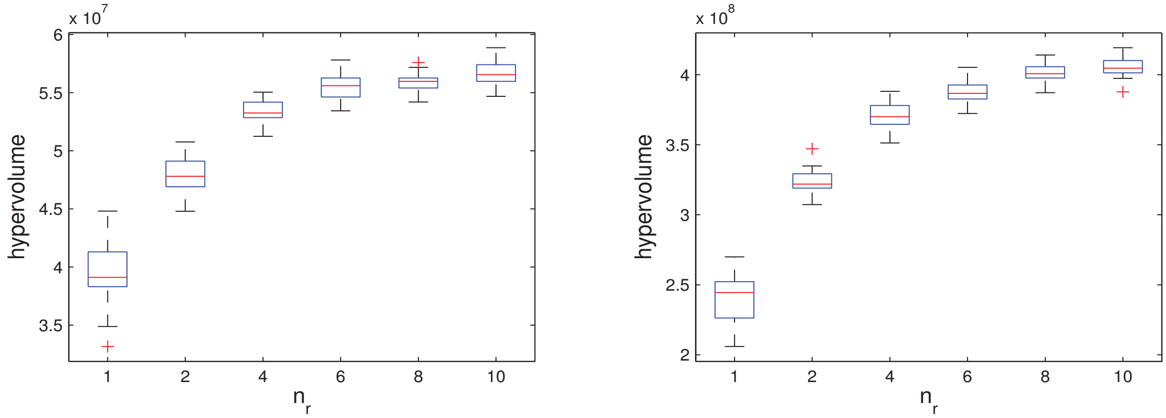
$$I_H(P, y^*) = volume(\bigcup_{f \in P}[f_1, y_1^*] \times \ldots [f_m, y_m^*]) \quad (10)$$

The higher the hypervolume, the better the approximation. In our experiments, we set $y^* = 1.1 * (f_1^{max}, f_2^{max})$ for MTSP test instances, where f_i^{max} is the maximum value of the i-th objective in the obtained solutions.

2) Inverted Generational Distance: Inverted Generational Distance (IGD) [32] is used to reflect how close the obtained solutions to the true Pareto Front, measuring the average distance from the true Pareto front to the closest solutions in the obtained solutions. Let P^* be a set of uniformly distributed points along the PF and P be the obtained set. It can be calculated as follows.

$$IGD(P, P^*) = \frac{1}{|P^*|}\sum_{v \in P^*} dist(v, P) \quad (11)$$

where $dist(*, *)$ is the Euclidean distance. The lower the IGD, the better the approximation. Note that *the ture PF* of the MTSP is unknown. So we use all non-dominated solutions obtained by all the algorithms in all runs to be *the reference Pareto front* P^* to calculate IGD.



(a) (b)

Fig. 2: Examination of the influence of n_r on the performance of AG-MOEA/D on two test instances. (a) $c-200/o-2$, (b) $c-600/o-2$.

TABLE I: HV and IGD results obtained by AG-MOEA/D (WP) and AG-MOEA/D (EA) on all the test instances. The t-test is conducted with 5% significance level. The best results for each instance are highlighted in boldface.

instance		HV		t-test	IGD		t-test
		AG-MOEA/D (WP)	AG-MOEA/D (EA)	p-value	AG-MOEA/D (WP)	AG-MOEA/D (EA)	p-value
$c-100$ $/o-2$	mean	1.7771E+07	**1.7951E+07**	3.3573E-02	3.3481E+02	**2.9507E+02**	6.2472E-03
	std	2.8571E+05	2.9598E+05		4.9695E+01	5.4226E+01	
$c-200$ $/o-2$	mean	6.9154E+07	**7.0701E+07**	1.3391E-06	6.3838E+02	**5.3539E+02**	4.9286E-04
	std	7.8285E+05	9.8755E+05		7.4420E+01	9.4672E+01	
$c-300$ $/o-2$	mean	1.3932E+08	**1.4487E+08**	3.5577E-10	1.1348E+03	**9.3871E+02**	1.9194E-06
	std	1.7769E+06	2.2003E+06		9.0623E+01	1.5364E+02	
$c-400$ $/o-2$	mean	2.6454E+08	**2.7852E+08**	1.8292E-18	1.6430E+03	**1.2214E+03**	9.0144E-12
	std	3.0717E+06	3.0352E+06		1.5177E+02	1.6246E+02	
$c-500$ $/o-2$	mean	3.6592E+08	**3.8854E+08**	1.1864E-16	1.9776E+03	**1.6351E+03**	1.0235E-07
	std	4.4427E+06	5.0861E+06		1.9225E+02	1.8985E+02	
$c-600$ $/o-2$	mean	5.1802E+08	**5.5336E+08**	4.9536E-21	2.4743E+03	**1.9378E+03**	5.0741E-11
	std	6.6235E+06	6.3168E+06		1.6064E+02	2.8287E+02	
$c-700$ $/o-2$	mean	6.9384E+08	**7.4071E+08**	2.6838E-16	2.9737E+03	**2.2867E+03**	3.0324E-11
	std	1.1512E+07	8.5340E+06		3.0821E+02	2.4396E+02	
$c-100$ $/o-3$	mean	8.8016E+10	**9.1375E+10**	5.8874E-05	5.5227E+02	**4.8105E+02**	1.9053E-06
	std	2.1205E+09	1.6523E+09		4.5283E+01	4.5309E+01	
$c-200$ $/o-3$	mean	9.0385E+10	**9.4116E+12**	3.9835E-14	1.3127E+03	**1.1386E+03**	1.8443E-09
	std	1.9343E+10	1.1374E+10		6.7410E+01	9.3513E+01	

C. Experimental Setups

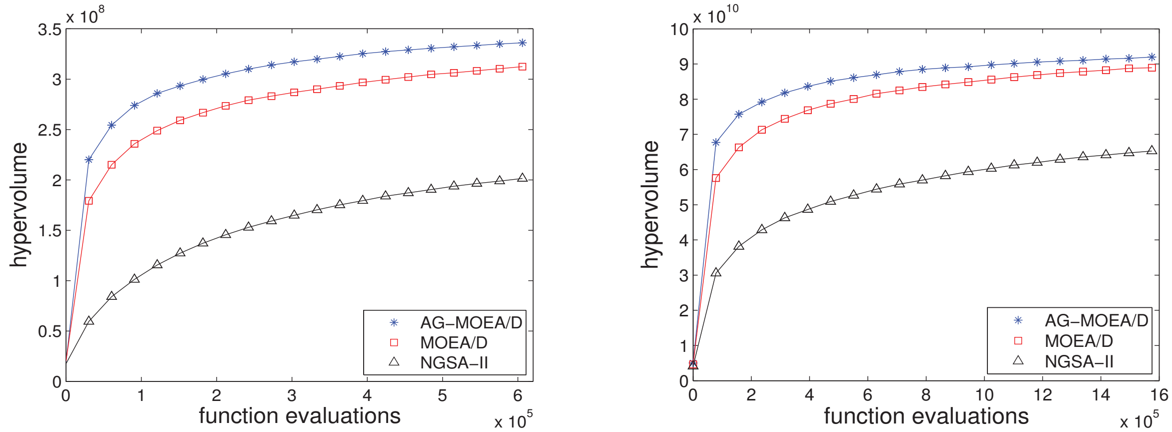
In this paper, seven algorithms are compared including NSGA-II [8], MOEA/D [13], MOEA/D-DRA [17], MOEA/D-GRA [26], EA-MOEA/D [19], EAG-MOEA/D [19] and 2EAG-MOEA/D [27]. It is worth to note that EA-MOEA/D can be considered as AG-MOEA/D without the dynamic resource allocation strategy. All the algorithms were implemented 30 independent runs on each test instance.

The crossover probability was set as $p_c = 1.0 (p_c = 0.8$ for NSGA-II) and the mutation probability was $p_m = 1/n$, where n denotes the number of decision variables. Set $LG = 10$ for all test problems. The neighborhood size T was 10, and the parameter δ was set to be 0.9. We set $n_r = 10$ for our proposed algorithm and MOEA/D-DRA ($n_r = 10$ is better than 1). The algorithms stop after a predefined function evaluations. The number of function evaluation is 400,000 for $c-100/o-2$ and $c-200/o-2$; 600,000 for $c-300/o-2$ and $c-400/o-2$; 800,000 for $c-500/o-2$, $c-600/o-2$ and $c-700/o-2$ and 1,500,000 for $c-100/o-3$ and $c-200/o-3$.

V. Experimental Results

A. Influence of Parameter n_r

To study the sensitivity of n_r to the proposed algorithms, we have tested the settings of $n_r \in [1, 2, 4, 6, 8, 10]$ on $c-200/o-2$ and $c-600/o-2$. It can be seen from Fig. 2 that the information from neighboring subproblems has a significant impact on the convergence of the problems for MOTSP. AG-MOEA/D with larger n_r values works better than those with small values.



(a) (b)

Fig. 3: The evolution of the median HV metric values versus the number of function evaluations on $c-400/o-2$ and $c-100/o-3$. (a) $c-400/o-2$, (b) $c-100/o-3$.

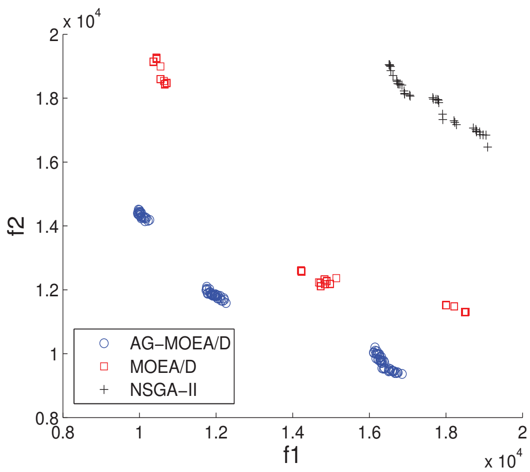


Fig. 4: The non-dominated solutions obtained by AG-MOEA/D, MOEA/D and NSGA-II on $c-400/o-2$ in the 100-th generation.

B. Comparisons of WP and EA mechanisms

Table I gives the comparative results of AG-MOEA/D (WP) and AG-MOEA/D (EA). Clearly, AG-MOEA/D (EA) outperforms AG-MOEA/D (WP) with statistical significance in all the test instances. The results indicate AG-MOEA/D using EA mechanism is able to better determine the right evolutionary phase.

C. Comparisons of AG-MOEA/D and classical algorithms

Fig. 3 presents the evolution of AG-MOEA/D, MOEA/D and NSGA-II on $c-400/o-2$ and $c-100/o-3$, in terms of hypervolume metric. AG-MOEA/D outperforms MOEA/D and NSGA-II along the evolutionary process. In addition, Fig. 4 shows the solutions obtained by three algorithms in 100-th generation, which indicates the effectiveness for the guidance of the external archive in terms of convergence at the early phase.

D. The Effect Of The Multi-phase Strategy

In order to verify the performance of the multi-phase strategy, the proposed algorithm is compared with EAG-MOEA/D [19], 2EAG-MOEA/D [27], MOEA/D-DRA [17] and MOEA/D-GRA [26], which are all MOEA/D variants with the dynamic allocation of computational resources. Similar to AG-MOEA/D, EAG-MOEA/D and 2EAG-MOEA/D use information extracted form the external archive to guide the search of the working population. MOEA/D-DRA and MOEA/D-GRA allocates the computational resource to different subproblems based on utility value of each subproblem: the improvement of a subproblem u_i and the hybrid utility function $u_{i,x}$ [26] is used as the utility value in MOEA/D-GRA. $u_{i,x}$ is the combination of the improvement of i-th subproblem and the population distribution of solution generated by i-th subproblem in the decision space. The former is called MOEA/D-GRA$_i$, and the latter is called MOEA/D-GRA$_{i,x}$.

Table II presents the experiment results of AG-MOEA/D, EA-MOEA/D, EAG-MOEA/D and 2EAG-MOEA/D on all test instances in terms of HV and IGD metrics. As can be seen from Table II, AG-MOEA/D performs the best on all the test instances, except for $c-400/o-2$. The comparison between AG-MOEA/D and EA-MOEA/D shows that the adopted mechanisms have successfully guided the working population. Compared with EAG-MOEA/D and 2EAG-MOEA/D, AG-MOEA/D makes a better balance between convergence and diversity in MOTSP. 2EAG-MOEA/D switches from convergence phase to diversity phase only once, whereas AG-MOEA/D switches back and forth between two phases for 24 times in our experiments at least.

Table III presents the experiment results of AG-MOEA/D, MOEA/D-DRA, MOEA/D-GRA$_i$ and MOEA/D-GRA$_{i,x}$ on MOTSP test instances. From table III, AG-MOEA/D shows a significant advantage over other compared algorithms with

TABLE II: HV and IGD results obtained by AG-MOEA/D, EA-MOEA/D, EAG-MOEA/D and 2EAG-MOEA/D on MOTSP instances. † and ‡ denotes the performance of the corresponding algorithm is significantly worse than or better than that of AG-MOEA/D according to the t-test with 5% significance level. The best results for each instance are highlighted in boldface.

instance		HV				IGD			
		AG-MOEA/D	EA-MOEA/D	EAG-MOEA/D	2EAG-MOEA/D	AG-MOEA/D	EA-MOEA/D	EAG-MOEA/D	2EAG-MOEA/D
$c-100$	mean	**1.8339E+07**	1.8091E+07 †	1.7835E+07 †	1.8181E+07 †	**2.9507E+02**	3.4540E+02 †	3.8142E+02 †	3.4298E+02 †
$/o-2$	std	2.9843E+05	3.2382E+05	3.1442E+05	3.0998E+05	5.4226E+01	5.5235E+01	5.7137E+01	5.3526E+01
$c-200$	mean	**6.3242E+07**	5.9900E+07 †	6.1908E+07 †	6.2421E+07 †	**5.3539E+02**	7.9134E+02 †	7.1393E+02 †	6.1347E+02 †
$/o-2$	std	9.6148E+05	7.4228E+05	1.3660E+06	9.6678E+05	9.4672E+01	8.1288E+01	1.5473E+02	8.7126E+01
$c-300$	mean	**1.5851E+08**	1.4973E+08 †	1.5646E+08 †	1.5674E+08 †	**9.3871E+02**	1.2737E+03 †	1.1855E+03 †	1.0179E+03 †
$/o-2$	std	2.2496E+06	1.6830E+06	1.9262E+06	2.5821E+06	1.5364E+02	9.7071E+01	2.0555E+02	1.3910E+02
$c-400$	mean	2.8796E+08	2.6532E+08 †	**2.8951E+08** ‡	2.8246E+08 †	**1.2214E+03**	1.9872E+03 †	1.2818E+03 †	1.4422E+03 †
$/o-2$	std	2.9950E+06	2.6032E+06	4.6707E+06	4.6671E+06	1.6246E+02	1.4290E+02	2.2650E+02	2.6008E+02
$c-500$	mean	**4.3252E+08**	3.9807E+08 †	4.2390E+08 †	4.1328E+08 †	**1.6351E+03**	2.2517E+03 †	1.9750E+03 †	1.9603E+03 †
$/o-2$	std	5.1869E+06	3.9111E+06	9.9077E+06	6.1677E+06	1.8985E+02	1.5742E+02	3.6409E+02	1.9951E+02
$c-600$	mean	**6.1196E+08**	5.5919E+08 †	6.0324E+08 †	5.8737E+08 †	**1.9378E+03**	2.8410E+03 †	2.5419E+03 †	2.3057E+03 †
$/o-2$	std	6.2684E+06	4.6137E+06	1.1310E+07	8.0325E+06	2.8287E+02	1.3013E+02	5.7238E+02	2.2223E+02
$c-700$	mean	**8.1148E+08**	7.3681E+08 †	8.0423E+08 †	7.7377E+08 †	**2.2867E+03**	3.5128E+03 †	2.6766E+03 †	2.9080E+03 †
$/o-2$	std	8.4132E+06	7.7932E+06	1.3562E+07	8.9050E+06	2.4396E+02	2.3774E+02	5.8396E+02	2.0834E+02
$c-100$	mean	**9.1976E+10**	8.9348E+10 †	8.9186E+10 †	8.8779E+10 †	**4.8105E+02**	5.5209E+02 †	5.9128E+02 †	5.9330E+02 †
$/o-3$	std	2.1247E+09	1.9504E+09	1.8646E+09	2.2832E+09	4.5283E+01	4.1360E+01	4.1164E+01	6.4449E+01
$c-200$	mean	**1.0044E+12**	9.6142E+11 †	9.8869E+11 †	9.6991E+11 †	**1.1386E+03**	1.3116E+03 †	1.2866E+03 †	1.3165E+03 †
$/o-3$	std	1.4768E+10	1.1344E+10	1.9440E+10	1.3674E+10	9.3513E+01	6.7448E+01	8.1012E+01	7.9770E+01

TABLE III: HV and IGD results obtained by AG-MOEA/D, MOEA/D-DRA, MOEA/D-GRA$_i$ and MOEA/D-GRA$_{i,x}$ on all the test instances. † and ‡ denotes the performance of the corresponding algorithm is significantly worse than or better than that of AG-MOEA/D according to the t-test with 5% significance level. The best results for each instance are highlighted in boldface.

instance		HV				IGD			
		AG-MOEA/D	MOEA/D-DRA	MOEA/D-GRA$_i$	MOEA/D-GRA$_{i,x}$	AG-MOEA/D	MOEA/D-DRA	MOEA/D-GRA$_i$	MOEA/D-GRA$_{i,x}$
$c-100$	mean	1.8339E+07	1.8589E+07 ‡	1.2658E+07 †	**2.2901E+07**‡	2.9507E+02	**2.8811E+02**	1.1687E+03 †	1.3300E+03 †
$/o-2$	std	2.9843E+05	3.4002E+05	5.5828E+05	9.0094E+06	5.4226E+01	6.1867E+01	1.1553E+02	2.4120E+02
$c-200$	mean	**6.3242E+07**	5.9646E+07 †	2.8315E+07 †	1.1719E+07 †	**5.3539E+02**	8.3078E+02 †	3.5079E+03 †	6.6542E+03 †
$/o-2$	std	9.6148E+05	1.2384E+06	3.2473E+06	1.2895E+06	9.4672E+01	1.0151E+02	3.8341E+02	1.3612E+03
$c-300$	mean	**1.5851E+08**	1.4875E+08 †	6.3908E+07 †	5.9159E+07 †	**9.3871E+02**	1.4276E+03 †	6.1830E+03 †	1.1050E+04 †
$/o-2$	std	2.2496E+06	2.1300E+06	1.7367E+07	1.8321E+07	1.5364E+02	1.5826E+02	1.4945E+03	1.7510E+03
$c-400$	mean	**2.8796E+08**	2.6489E+08 †	7.2717E+07 †	6.3613E+07 †	**1.2214E+03**	2.1367E+03 †	1.1359E+04	1.9063E+04 †
$/o-2$	std	2.9950E+06	5.1171E+06	4.0308E+07	3.9588E+07	1.6246E+02	2.5491E+02	3.1031E+03	3.1757E+03
$c-500$	mean	**4.3252E+08**	3.9593E+08 †	7.4790E+07 †	1.0101E+08 †	**1.6351E+03**	2.4437E+03 †	1.6147E+04 †	2.3731E+04 †
$/o-2$	std	5.1869E+06	6.0841E+06	6.4201E+07	6.3468E+07	1.8985E+02	2.5864E+02	4.2502E+03	4.1842E+03
$c-600$	mean	**6.1196E+08**	5.5535E+08 †	1.0201E+08 †	1.0137E+08 †	**1.9378E+03**	3.2657E+03 †	1.9371E+04 †	3.1492E+04 †
$/o-2$	std	6.2684E+06	1.2181E+07	8.1400E+07	8.3483E+07	2.8287E+02	3.2007E+02	4.8085E+03	4.8899E+03
$c-700$	mean	**8.1148E+08**	7.3135E+08 †	7.7598E+07 †	1.0886E+08 †	**2.2867E+03**	3.6448E+03 †	2.5609E+04 †	3.8406E+04 †
$/o-2$	std	8.4132E+06	1.2460E+07	7.4872E+07	9.5405E+07	2.4396E+02	3.4825E+02	4.2573E+03	5.5784E+03
$c-100$	mean	**9.1976E+10**	7.1578E+10 †	7.0227E+10 †	6.9368E+10 †	**4.8105E+02**	9.2125E+02 †	9.2329E+02†	9.4613E+02†
$/o-3$	std	2.1247E+09	2.4636E+09	2.0947E+09	2.1441E+09	4.5283E+01	7.6145E+01	5.0999E+01	6.2971E+01
$c-200$	mean	**1.0044E+12**	6.8101E+11 †	6.5061E+11 †	6.5735E+11 †	**1.1386E+03**	2.5041E+03 †	2.6107E+03 †	2.5505E+03†
$/o-3$	std	1.4768E+10	1.8337E+10	1.8913E+10	2.1742E+10	9.3513E+01	1.5112E+02	1.2703E+02	1.2332E+02

statistical significance in all the test instances, except for $c-100/o-2$. MOEA/D-DRA and MOEA/D-GRA$_i$ utilize the convergence information to allocate computing resources. MOEA/D-GRA$_{i,x}$ simply sums the convergence information and diversity information up. The comparative result indicates that the multi-phase strategy performs better than any of them.

In summary, the results show that AG-MOEA/D can find a better balance between convergence and diversity during the evolutionary process on the travelling salesman problem.

VI. Conclusion

The paper proposes a multi-phase strategy for dynamic allocation of computational resources in the framework of MOEA/D. Experiments verify that the proposed algorithm achieves better performance than NSGA-II, MOEA/D and its state-of-the-art variants in MOTSP.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (NSFC) under grant 61300159, 61175073 and 61473241, by the Natural Science Foundation of Jiangsu Province under grant BK20130808, by China Postdoctoral Science Foundation under grant 2015M571751 and by the Fundamental Research Funds for the Central Universities of China under grant NZ2013306.

References

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

[2] C. M. Fonseca and P. J. Fleming, "An Overview of Evolutionary Algorithms in Multiobjective Optimization," *Evolutionary Computation*, vol. 3, no. 1, pp. 1–16, Spring 1995.

[3] C. A. Coello Coello, "Evolutionary multiobjective optimization: A historical view of the field," *IEEE Computational Intelligence Magazine*, vol. 1, no. 1, pp. 28–36, February 2006.

[4] P. A. Bosman and D. Thierens, "The Balance Between Proximity and Diversity in Multiobjective Evolutionary Algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174–188, April 2003.

[5] J. J. Durillo, A. J. Nebro, F. Luna, and E. Alba, "On the Effect of the Steady-State Selection Scheme in Multi-Objective Genetic Algorithms," in *Evolutionary Multi-Criterion Optimization. 5th International Conference, EMO 2009*, M. Ehrgott, C. M. Fonseca, X. Gandibleux, J.-K. Hao, and M. Sevaux, Eds. Nantes, France: Springer. Lecture Notes in Computer Science Vol. 5467, April 2009, pp. 183–197.

[6] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," in *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, Eds., Athens, Greece, 2002, pp. 95–100.

[7] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester, UK: John Wiley & Sons, 2001.

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, April 2002.

[9] E. Zitzler and S. Künzli, "Indicator-based Selection in Multiobjective Search," in *Parallel Problem Solving from Nature - PPSN VIII*, X. Y. et al., Ed. Birmingham, UK: Springer-Verlag. Lecture Notes in Computer Science Vol. 3242, September 2004, pp. 832–842.

[10] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 16 September 2007.

[11] J. Bader and E. Zitzler, "Hype: An algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[12] J. D. Schaffer and J. J. Grefenstette, "Multiobjective Learning via Genetic Algorithms," in *Proceedings of the 9th International Joint Conference on Artificial Intelligence (IJCAI-85)*. Los Angeles, California: AAAI, 1985, pp. 593–595.

[13] Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, December 2007.

[14] E. J. Hughes, "Multiple Single Objective Pareto Sampling," in *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, vol. 4. Canberra, Australia: IEEE Press, December 2003, pp. 2678–2684.

[15] E. Hughes, "MSOPS-II: A general-purpose many-objective optimiser," in *IEEE Congress on Evolutionary Computation, CEC2007*, Sept 2007, pp. 3944–3951.

[16] T. Murata, H. Ishibuchi, and M. Gen, "Specification of Genetic Search Directions in Cellular Multi-objective Genetic Algorithms," in *First International Conference on Evolutionary Multi-Criterion Optimization*, E. Zitzler, K. Deb, L. Thiele, C. A. C. Coello, and D. Corne, Eds. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001, pp. 82–95.

[17] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained mop test instances," School of CS and EE, University of Essex, Working Report CES-491, Feburary 2009.

[18] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching based selection in evolutionary multiobjective optimization," *IEEE Transaction on Evolutionary Computation*, vol. 18, pp. 909–923, 2014.

[19] X. Cai, Y. Li, Z. Fan, and Q. Zhang, "An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization," *IEEE Transaction Evolutionary Computation*, vol. 19, no. 4, pp. 508–523, 2015.

[20] S. B. Gee, K. C. Tan, V. A. Shim, and N. R. Pal, "Online diversity assessment in evolutionary multiobjective optimization: A geometrical perspective," *IEEE Transaction on Evolutionary Computation*, vol. 19, no. 4, pp. 542–559, 2015.

[21] H. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Transaction Evolutionary Computation*, vol. 18, no. 3, pp. 450–455, 2014.

[22] M. Asafuddoula, T. Ray, and R. A. Sarker, "A decomposition-based evolutionary algorithm for many objective optimization," *IEEE Transaction on Evolutionary Computation*, vol. 19, no. 3, pp. 445–460, 2015.

[23] L. Ke, Q. Zhang, and R. Battiti, "MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and antcolony," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1845–1859, 2013.

[24] Y. Mei, K. Tang, and X. Yao, "Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem," *IEEE Transaction on Evolutionary Computation*, vol. 15, no. 2, pp. 151–165, 2011.

[25] T.-C. Chiang and Y.-P. Lai, "MOEA/D-AMS: Improving MOEA/D by an adaptive mating selection mechanism," in *IEEE Congress on Evolutionary Computation (CEC'2011)*, 2011, pp. 1473–1480.

[26] A. Zhou and Q. Zhang, "Are all the subproblems equally important? resource allocation in decomposition based multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, accepted, 2015.

[27] N. Zhang, Y. Huang, and X. Cai, "A two-phase external archive guided multiobjective evolutionary algorithm for the software next release problem," in *Bio-Inspired Computing – Theories and Applications: 10th International Conference, BIC-TA 2015*, September 2015, pp. 664–675.

[28] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston: Kluwer Academic Publishers, 1999.

[29] 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.

[30] H. Wang, L. Jiao, and X. Yao, "Two_arch2: An improved two-archive algorithm for many-objective optimization," *IEEE Trans. Evolutionary Computation*, vol. 19, no. 4, pp. 524–541, 2015.

[31] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[32] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance Assessment of Multiobjective Optimizers: An Analysis and Review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, April 2003.