

Multiobjective evolutionary algorithm for frequency assignment problem in satellite communications

Jiahai Wang · Yiqiao Cai

Published online: 24 June 2014
© Springer-Verlag Berlin Heidelberg 2014

Abstract Satellite communications technology leads to an important improvement in our life and world. The frequency assignment problem (FAP) is a fundamental problem in satellite communication system for providing high-quality transmissions. The whole goal of the FAP in satellite communication system is to minimize co-channel interference between two satellite systems by rearranging frequency assignment. Recently, many metaheuristics, including neural networks and evolutionary algorithms, are proposed for this NP-complete problem. All such algorithms formulate the FAP as a single-objective problem, although it obviously has two objectives and thus essentially is a multiobjective optimization problem. This study explicitly formulates FAP as a multiobjective optimization problem and presents a multiobjective evolutionary algorithm based on decomposition (MOEA/D) with a problem-specific subproblem-dependent heuristic assignment (SHA), called MOEA/D-SHA, for the multiobjective FAP. Simulation results show that the MOEA/D-SHA outperforms significantly general-purpose MOEA/D, and an off-the-shelf multiobjective algorithm, i.e., NSGA-II. The advantages of the MOEA/D-SHA over the state-of-the-art single-objective approaches are also shown.

Keywords Frequency assignment problem · Multiobjective optimization · Differential evolution · Decomposition · Subproblem-dependent heuristic assignment

1 Introduction

Wireless communication has received lots of attention these years due to its various applications including mobile telephone, satellite communications, TV broadcasting, wireless LANs, and military operations. Frequency assignment problems (FAPs) (Aardal et al. 2007) have arisen in many different situations in the field of wireless communications. In each of these situations, a FAP arises with application-specific characteristics. All these applications lead to many different FAP models and to many different types of instances within the models (Aardal et al. 2007). Different FAP models and solution techniques for each of the features of the problem, such as the handling of interference among radio signals, the availability of frequencies, and the optimization criterion or objective, are comprehensively reviewed in Aardal et al. (2007). Typical cases include the FAP in digital cellular phone standard GSM (general system for mobile communication) (Maximiano et al. 2009a,b, 2013; Luna et al. 2011), and the FAP in satellite communication (Pelton et al. 2004; Whalen 2008; Mizuike and Ito 1986, 1989; Kurokawa and Kozuka 1993; Funabiki and Nishikawa 1997; Liu et al. 2007; Wang et al. 2008, 2011; Salcedo-Sanz et al. 2004; Salcedo-Sanz and B-Calz' on 2005; Salman et al. 2010).

In this study, we focus on the FAP in satellite communication systems (Pelton et al. 2004; Whalen 2008; Mizuike and Ito 1986, 1989; Kurokawa and Kozuka 1993; Funabiki and Nishikawa 1997; Liu et al. 2007; Wang et al. 2008, 2011; Salcedo-Sanz et al. 2004; Salcedo-Sanz and

Communicated by V. Loia.

J. Wang (✉)
Department of Computer Science, Sun Yat-sen University,
Guangzhou 510006, People's Republic of China
e-mail: wjiahai@hotmail.com

Y. Cai
College of Computer Science and Technology, Huaqiao University,
Xiamen 361021, People's Republic of China

B-Calz'on 2005; Salman et al. 2010). Satellite communication technology has refined our life and world by playing a key role in creating a global culture by spreading worldwide entertainment, simulating technological and cultural interchange, and promoting trade around the world (Pelton et al. 2004; Whalen 2008). In satellite communication systems, the reduction of the co-channel interference has become a major factor for determining system design (Mizuike and Ito 1986, 1989; Kurokawa and Kozuka 1993; Funabiki and Nishikawa 1997; Liu et al. 2007; Wang et al. 2008, 2011; Salcedo-Sanz et al. 2004; Salcedo-Sanz and B-Calz'on 2005; Salman et al. 2010). Furthermore, due to the necessity of accommodating as many satellites as possible in geostationary orbit, this interference reduction has become an even more important issue with the increase of geostationary satellites (Mizuike and Ito 1986, 1989; Kurokawa and Kozuka 1993; Funabiki and Nishikawa 1997; Liu et al. 2007; Wang et al. 2008, 2011; Salcedo-Sanz et al. 2004; Salcedo-Sanz and B-Calz'on 2005; Salman et al. 2010). To deal with interference reduction in practical situations, the rearrangement of frequency assignments is considered as an effective measure (Mizuike and Ito 1986, 1989; Kurokawa and Kozuka 1993; Funabiki and Nishikawa 1997; Liu et al. 2007; Wang et al. 2008, 2011; Salcedo-Sanz et al. 2004; Salcedo-Sanz and B-Calz'on 2005; Salman et al. 2010).

There are two objectives for the FAP in satellite communications. The first objective (named f_L) is to minimize the largest interference of elements selected for the assignment, and the second objective (named f_T) is to minimize the total interference of all the selected elements (Mizuike and Ito 1986, 1989; Kurokawa and Kozuka 1993; Funabiki and Nishikawa 1997; Liu et al. 2007; Wang et al. 2008, 2011; Salcedo-Sanz et al. 2004; Salcedo-Sanz and B-Calz'on 2005; Salman et al. 2010). The FAP is proven to be NP-complete (Mizuike and Ito 1989), thus a good exact algorithm for its optimal solution in polynomial time is unlikely to exist. Mizuike and Ito (1986, 1989) proposed a segmentation of frequency band and an exact algorithm based on branch-and-bound for the FAP. But the branch-and-bound algorithm may fail when applied to large instances since the FAP is NP-complete.

Recently, a lot of metaheuristics are proposed for the FAP. Most of these algorithms are based on Hopfield neural network (Hopfield and Tank 1985). In 1993, Kurokawa and Kozuka (1993) firstly proposed a Hopfield neural network that consists of $M \times M$ neurons for the FAP with N -carrier M -segment. But this Hopfield neural network algorithm cannot be applied to practical large size systems because of its very small convergence rate and requirements of large number of neurons and computation time (Kurokawa and

Kozuka 1993; Funabiki and Nishikawa 1997). Funabiki and Nishikawa (1997) proposed a gradual neural network (GNN) that consists of $N \times M$ neurons. The GNN has the advantage over the Hopfield neural network algorithm in Kurokawa and Kozuka (1993) because the GNN reduces the required number of neurons. However, the multiphase searching leads to heavy computation burden, and the GNN also cannot be applied to the FAP with large size (Wang et al. 2008). In 2007, Liu et al. (2007) proposed a transiently chaotic neural network (TCNN) for the FAP. In the TCNN, transient chaotic dynamics are introduced to help Hopfield neural network escape from local minima. In 2008, Wang et al. (2008) proposed a noisy chaotic neural network with variable thresholds (NCNN-VT) for the FAP. In the NCNN-VT, noisy dynamics are introduced to the TCNN to further improve the performance. But in the NCNN-VT, it is difficult to control and balance the chaotic dynamics, stochastic dynamics and gradient ascent dynamics for converging to a stable equilibrium point corresponding to an acceptably near-optimal solution. Furthermore, the rate of convergence to feasible solution is not very high in particular when the problem size becomes large. Recently, Wang et al. (2011) proposed a multi-start stochastic competitive Hopfield neural network (MS-SCHNN) for the FAP. The MS-SCHNN can always converge to feasible solution due to competitive mechanism, and also can escape from local minima due to the stochastic dynamics and multi-start scheme. Therefore, the MS-SCHNN obtains better results than all previous Hopfield neural network-based approaches (Wang et al. 2011).

In 2004 and 2005, Salcedo-Sanz et al. (2004), Salcedo-Sanz and B-Calz'on (2005) proposed a hybrid simulated annealing and a hybrid genetic algorithm, respectively, for the FAP. These two hybrid algorithms fail to obtain solutions in large size problems due to the excessive computation time (Wang et al. 2008). Recently, Salman et al. (2010) proposed six versions of differential evolution (DE) (Storn and Price 1997; Das and Suganthan 2011) algorithms for the FAP. Among the previous algorithms for the FAP (Kurokawa and Kozuka 1993; Funabiki and Nishikawa 1997; Liu et al. 2007; Wang et al. 2008, 2011; Salcedo-Sanz et al. 2004; Salcedo-Sanz and B-Calz'on 2005; Salman et al. 2010), the MS-SCHNN (Wang et al. 2011) and the DE (Salman et al. 2010) are two best single-objective optimization algorithms proposed recently.

The FAP is handled as a single-objective optimization problem in previous works (Kurokawa and Kozuka 1993; Funabiki and Nishikawa 1997; Liu et al. 2007; Wang et al. 2008, 2011; Salcedo-Sanz et al. 2004; Salcedo-Sanz and B-Calz'on 2005; Salman et al. 2010). In the neural network-based algorithms (Kurokawa and Kozuka 1993; Funabiki and Nishikawa 1997; Liu et al. 2007; Wang et al. 2008, 2011),

only the largest interference f_L is encoded into the energy function to be minimized. The hybrid simulated annealing (Salcedo-Sanz et al. 2004) and hybrid genetic algorithm (Salcedo-Sanz and B-Calz'on 2005) also only concentrate on achieving minimum largest interference. Thus, the final results are strongly biased to the considered objective. In the DE (Salman et al. 2010), the weighted-sum approach is used and thus the aforementioned two objectives of the FAP are aggregated into a single-objective function as follows:

$$\min F = w_1 f_L + w_2 f_T \quad (1)$$

where w_1 and w_2 are two weight coefficients. The DE is performed for two different settings or bias: one with the aim of producing solution with minimum largest interference by setting $w_1 = 5.0$, and $w_2 = 0.01$, while the other with the objective of producing solution with minimum total interference by setting $w_1 = 0.5$, and $w_2 = 5.0$. This might be inappropriate because total interference and largest interference are generally of different scales. Summing them up, as a unified score, does not really provide meaningful information about the quality of solutions, and might cause difficulties in practice as it is hard to determine the appropriate values of weights. Furthermore, it is impossible that a single solution provided by the single-objective optimization algorithm is optimal in terms of both total interference and largest interference.

Although the FAP obviously has two objectives, it has never been explicitly investigated in a multiobjective fashion. This study explicitly formulates FAP as a multiobjective optimization problem and thus presents a multiobjective evolutionary algorithm based on decomposition (MOEA/D) with a problem-specific subproblem-dependent heuristic assignment (SHA), called MOEA/D-SHA, for the multiobjective FAP (MO-FAP). That is, we resort to balancing between total interference and largest interference, and seeking a good trade-off using multiobjective evolutionary algorithm. Simulation results show that the MOEA/D-SHA outperforms significantly general-purpose MOEA/D as well as the state-of-the-art in multiobjective evolutionary algorithms based on Pareto dominance, i.e., NSGA-II. The advantages of the MOEA/D-SHA over the state-of-the-art single-objective approaches for the FAP are also shown through simulation results.

The contribution of this study is threefold. First, the FAP in satellite communication is formulated explicitly as a multiobjective problem for the first time. Note that multiobjective optimization formulation of the particular FAP scenario in global system for mobile communications (GSM) networks is now starting to be researched (Max-

imiano et al. 2009a,b, 2013). But the FAP in satellite communication considered in this study is far different from those reported in the literature Luna et al. (2011), Maximiano et al. (2009a,b, 2013) with similar names because it comes from different application fields and thus has different problem features including constraints and objectives. Simulation results confirm the multiobjective nature of the FAP considered, and show some advantages in using the multiobjective problem formulation over single-objective approaches. Second, a multiobjective evolutionary algorithm, called MOEA/D-SHA, is proposed for the MO-FAP, where problem-specific SHA is designed. The proposed MOEA/D-SHA is the first multiobjective optimization-based algorithm for the FAP, and thus can be seen as a benchmark algorithm for future studies. Finally, the proposed MOEA/D-SHA is comprehensively compared with other popular multiobjective evolutionary algorithms and the state-of-the-art single-objective algorithms for the FAP. Simulation results show that the proposed algorithm can obtain better performance. The proposed algorithm not only manages to achieve almost the same or better solutions than those obtained by single-objective algorithms, but also finds simultaneously a set of alternative solutions. These solutions show different trade-offs between two objectives, which can be used to yield insight into the choices available to the decision maker and hence facilitate final decision making.

The remaining sections of this study are organized as follows. Section 2 gives the background, including the formulation of the MO-FAP, some concepts of multiobjective problem, and briefly introduction of the DE. In Sect. 3, the MOEA/D-SHA for the MO-FAP is given in detail. In Sect. 4, benchmark data sets are used to evaluate the MOEA/D-SHA. Last Section concludes this study.

2 Background

2.1 Formulation of FAP

Given two adjacent satellite systems, the whole goal of the FAP is to reduce the intersystem co-channel interference by rearranging the frequency assignment on carrier in system 2, while fixing the assignment in system 1. Since each carrier usually occupies a different length in a frequency band, the segmentation strategy of carriers is firstly proposed by Mizuike and Ito (1986, 1989). In the segmentation strategy, the commonly shared frequency band is divided into a number of segments so that every carrier can be described by a collection of consecutive unit segment. Then, an $M \times M$ interference matrix E can be adopted to describe the interference between two M -segment systems. The ij -th element e_{ij}

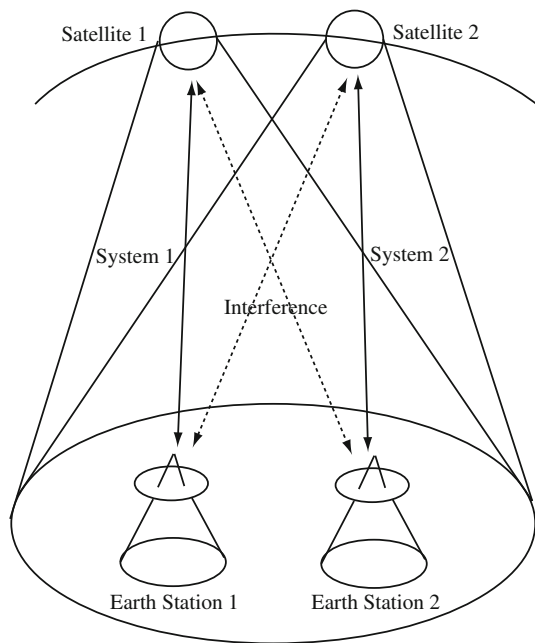


Fig. 1 Intersystem co-channel interference

in interference matrix E represents the co-channel interference when segment i in system 2 uses a common frequency with segment j in system 1.

The FAP can be formulated as a combinatorial optimization problem with three constraints and two objectives. The three constraints of the FAP are (Kurokawa and Kozuka 1993; Funabiki and Nishikawa 1997; Liu et al. 2007; Wang et al. 2008, 2011; Salcedo-Sanz et al. 2004; Salcedo-Sanz and B-Calz'on 2005; Salman et al. 2010):

Constraint 1: Every segment in system 2 must be assigned to a segment in system 1;

Constraint 2: Each segment in system 1 can be assigned by at most one segment in system 2;

Constraint 3: All segments of each carrier in system 2 must be assigned to consecutive segments in system 1 in the same order.

The two objectives (f_L, f_T) of the FAP are (Kurokawa and Kozuka 1993; Funabiki and Nishikawa 1997; Liu et al. 2007; Wang et al. 2008, 2011; Salcedo-Sanz et al. 2004; Salcedo-Sanz and B-Calz'on 2005; Salman et al. 2010):

f_L : Minimize the largest element of the interference matrix selected in the assignment;

f_T : Minimize the total interference of all the selected elements.

Next, a small size benchmark instance (denoted as BM1 in literature Kurokawa and Kozuka 1993; Funabiki and Nishikawa 1997; Liu et al. 2007; Wang et al. 2008, 2011; Salcedo-Sanz et al. 2004; Salcedo-Sanz and B-Calz'on 2005; Salman et al. 2010 and in this study) is given as an example

to explain the concepts of the FAP. The BM1 is firstly introduced by Mizuike and Ito (1986, 1989), and widely adopted by previous references (Kurokawa and Kozuka 1993; Funabiki and Nishikawa 1997; Liu et al. 2007; Wang et al. 2008, 2011; Salcedo-Sanz et al. 2004; Salcedo-Sanz and B-Calz'on 2005; Salman et al. 2010) as an example. Thus, Figs. 1, 2, 3 are also adopted or extended from previous references to describe the FAP concepts. Figure 1 illustrates the intersystem co-channel interference between two adjacent satellite systems. Sidelobe emission for earth station antennas can cause the intersystem interference (indicated by dotted arrows) when they are operated in the same frequency band. Figure 2 shows an example of the co-channel interference model. The communications are assumed to be operated in the frequency band between F_a and F_b . Each carrier can be described as a collection of consecutive unit segments in a frequency band. Three and four carriers are utilized there in each satellite system, respectively. The co-channel interference is evaluated by each pair of carriers using the same frequency. Thus, to secure the communication quality of all carriers, the largest interference must be minimized among all pairs. In Fig. 2, interference is indicated by plain arrows and dotted arrow indicates no interference. The total interference should be also minimized to improve the overall communication quality. Both systems have $M = 6$ segments, and system 2 has $N = 4$ carriers. To reduce the co-channel interference between two systems, the frequency assignments in system 2 are rearranged while the frequencies used in system 1 are fixed. The representation of the full assignment of segments can be described by an $M \times M$ matrix F , where element $f_{ij} = 1$ means that the segment i in system 2 has been reassigned to segment j in system 1. Figure 3a illustrates the segmentation of the system in Fig. 2, and the optimum assignment under the interference matrix in Fig. 3b. In the interference matrix, C_{ij} denotes the j -th carrier in system i , S_{ij} denotes the j -th segment in system i , and the asterisk element denotes infinity which indicates a penalty cost to avoid impossible assignment of segments for carrier allocation.

Thus, the first objective function can be described as follows:

$$f_L = \max(e_{ij} \times f_{ij}) \quad (2)$$

The second objective function can be described as follows:

$$f_T = \sum_{i=1}^M \sum_{j=1}^M e_{ij} \times f_{ij} \quad (3)$$

Fig. 2 Co-channel interference model of the system

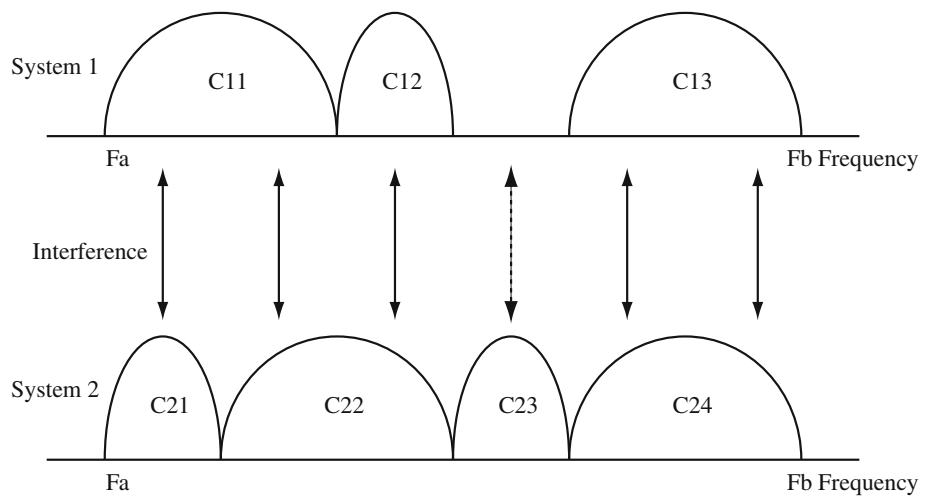
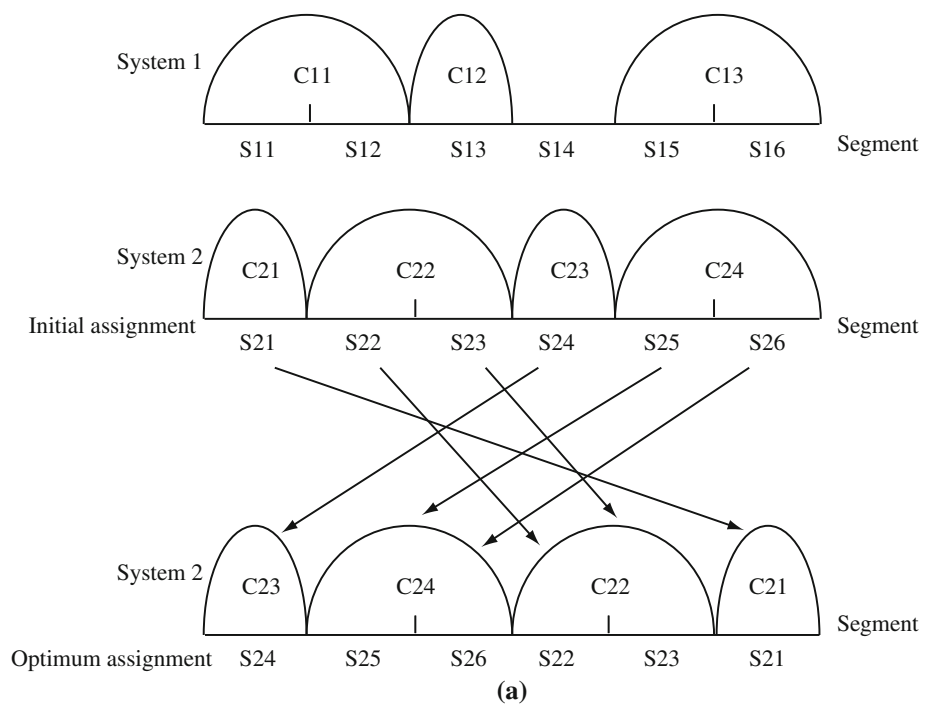


Fig. 3 **a** Segmentation of the system in Fig. 2. **b** Interference matrix of the system in Fig. 2



		C11		C12		C13	
		S11	S12	S13	S14	S15	S16
C21	S21	20	20	40	0	25	25
	S22	50	10	30	0	55	*
C22	S23	*	50	30	0	15	55
	S24	30	30	45	0	35	35
C23	S25	45	5	25	0	50	*
	S26	*	45	25	0	10	50

(b)

In this study, the FAP is handled as a multiobjective problem, named MO-FAP, where both largest interference and total interference are simultaneously optimized without bias as follows:

$$\min F = \{f_L, f_T\} \quad (4)$$

Our interpretation of the FAP as a multiobjective problem according to Eq. (4) represents a philosophically different view of the problem as the whole. This simple translation of the FAP into a multiobjective problem is surprisingly effective, which will be shown in simulation.

2.2 Multiobjective optimization

A multiobjective problem can be defined as follows:

$$\begin{aligned} \min F(X) &= \{f_1(X), \dots, f_m(X)\}, \\ \text{s.t. : } X &\in \Omega, \end{aligned}$$

where Ω is the decision variable space. $F : \Omega \rightarrow R^m$ consists of m objective functions that are often conflicting with each other. For a multiobjective problem, our goal is to seek a set of solutions that have good trade-offs among the objectives. It is also commonly required that this set of solutions is as diverse as possible.

The concepts of domination relationship and Pareto optimality in multiobjective optimization are defined as follows.

Definition 1 Pareto dominance: Let $u, v \in R^m$, u dominates v iff $u_i \leq v_i$ for each $i \in \{1, \dots, m\}$ and $u_j < v_j$ for at least one $j \in \{1, \dots, m\}$.

Definition 2 Pareto optimality: A decision variable $x^* \in \Omega$ is said to be *Pareto optimal* if there is no other $x \in \Omega$ so that $F(x)$ dominates $F(x^*)$. The set of all Pareto optimal solutions in the decision space is called the Pareto optimal set. The image of the Pareto optimal set in the objective space is called the Pareto optimal front.

An algorithm for a multiobjective problem should return a set of nondominated solutions (NDS) that can well approximate the Pareto optimal front (Pareto optimal set) as diverse as possible. Figure 4 illustrates the Pareto optimal front in the MO-FAP with $m = 2$ (i.e., total interference and largest interference). The Pareto optimal solutions in the Pareto optimal front (marked as solid circles) provide better total interference and/or largest interference than any other solution in the objective space. The remaining solutions (marked as open circles) are all dominated by at least one solution of the Pareto optimal front. Solutions X^A and X^B (marked as solid stars) are often called the extreme

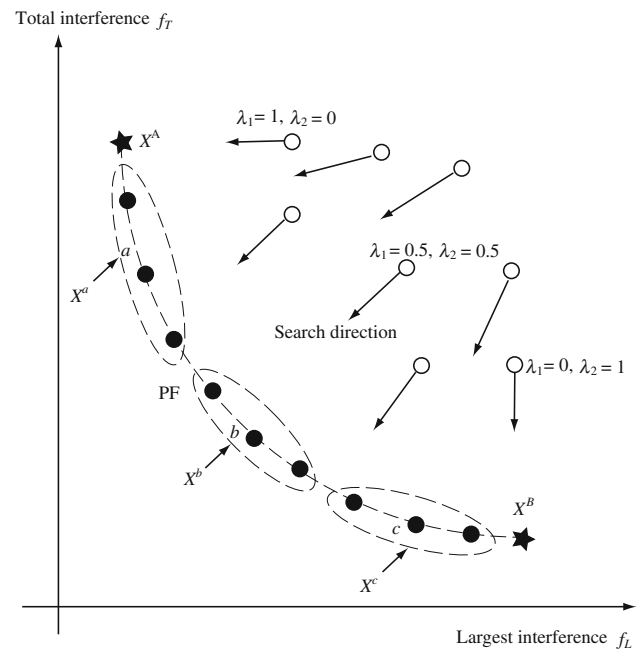


Fig. 4 Pareto optimal front of FAP and classification of the Pareto optimal assignments of FAP

points of the Pareto optimal front. These extreme points provide the lowest total interference and largest interference, respectively, than any other solution in the objective space.

Over the years, a number of metaheuristics, including evolutionary algorithms (Zhou et al. 2011), simulated annealing and tabu search (Talbi et al. 2012), have been extended to solve multiobjective problem. The evolutionary algorithms can find a good approximation of the Pareto optimal front in a single run due to their population-based nature, which makes them more natural candidates for multiobjective problems. Thus, multiobjective evolutionary algorithms have gained popularity and have been used extensively to solve the multiobjective problem over the last years (Zhou et al. 2011). The use of Pareto dominance for fitness evaluation has been the mainstream in multiobjective evolutionary algorithm for the last two decades. The NSGA-II (Deb et al. 2002) is a Pareto dominance-based multiobjective evolutionary algorithm which can be broadly regarded as a representative of multiobjective evolutionary algorithms and, probably, the most widely used. It is relatively simple, reliable, well-studied, and can readily incorporate constraints. The key characteristic of the NSGA-II is that it uses a fast nondominated sorting and a crowded distance estimation for comparing the quality of different solutions during selection. The NSGA-II has shown superior performance on not only benchmark problems (Deb et al. 2002), but also real-world application (Saha 2009;

Wang et al. 2010; Praditwong et al. 2011; Garcia-Najera and Bullinaria 2011), and thus has been an off-the-shelf choice for solving multiobjective problem. Recently, it has been pointed out in some studies that Pareto dominance-based algorithms do not always work well on multiobjective problem with many objectives (Peng et al. 2009; Li et al. 2010). Decomposition (or scalarizing function)-based fitness evaluation is a promising alternative to Pareto dominance especially for the many-objective problems and multiobjective combinatorial optimization problems (Peng et al. 2009; Li et al. 2010). Thus, the classical approach for multiobjective problem based on decomposition or aggregation is re-formularized into evolutionary algorithm recently (Zhang and Li 2007; Li and Zhang 2009; Zhang et al. 2009). A representative one of this kind of algorithms is MOEA/D (Zhang and Li 2007; Li and Zhang 2009; Zhang et al. 2009). The MOEA/D framework has been studied and used for dealing with a number of multiobjective problems (Mei et al. 2011; Konstantinidis and Yang 2011a,b; Li and Landa-Silva 2011; Ke et al. 2013; Shim et al. 2012; Czyzszak and Jaszkiwicz 1998). More details about multiobjective problem and other approaches to the multiobjective problems can be referred to the most recent review (Zhou et al. 2011; Talbi et al. 2012).

2.3 Differential evolution

The DE (Storn and Price 1997) starts with a population of NP D -dimensional individuals, which encode the candidate solution vectors, i.e., $X^i = (x_1^i, x_2^i, \dots, x_D^i)$, $i = 1, \dots, NP$. Then, the DE iteratively implements three operators: mutation, crossover and selection. Mutation and crossover are used to generate new trial vectors for each target vector, and selection then determines which vector will survive into the next generation. The three operators are briefly reviewed as follows (Storn and Price 1997).

Mutation For each target vector X^i that belongs to the current population, the DE randomly samples three other different individuals from the same generation to create a mutant vector $U^i = (u_1^i, u_2^i, \dots, u_D^i)$. The d -th component of the mutant vector is generated according to

$$u_d^i = x_d^{r_1} + F(x_d^{r_2} - x_d^{r_3}) \quad (5)$$

where $r_1 \neq r_2 \neq r_3 \neq i$, and the scalar factor F (usually $\in [0, 1]$) controls the amplification of the difference $(x_d^{r_2} - x_d^{r_3})$.

Crossover The target vector is mixed with the mutated vector to yield a trail vector Y^i . The d -th component of the

trail vector is generated as follows:

$$y_d^i = \begin{cases} u_d^i & \text{if } (\text{rand}() < \text{CR}) \text{ or } (d = d_{\text{rand}}) \\ x_d^i & \text{otherwise} \end{cases} \quad (6)$$

where $\text{rand}()$ is a uniformly distributed random number within the range $[0, 1]$; $\text{CR} \in [0, 1]$ is called crossover rate; and d_{rand} is a randomly chosen integer in the range $[1, D]$.

Selection A one-to-one greedy selection scheme is used. That is, if the trial vector Y^i is better than its parent X^i , it will replace its parent in the next generation; otherwise, the parent will be retained in the population.

3 MOEA/D-SHA for MO-FAP

The MOEA/D framework (Zhang and Li 2007) is chosen for solving the MO-FAP defined by Eq. (4). The reasons are (Peng et al. 2009; Li et al. 2010; Zhang and Li 2007; Ke et al. 2013): (1) the research literature shows that the MOEA/D seems to be more suitable for tackling multiobjective combinatorial optimization problems because it can directly use problem-specific (local search) techniques to intensify the exploration of promising regions in search space, and (2) the MOEA/D framework provides a very natural framework for using single-objective search techniques. The general framework of the MOEA/D is reviewed as follows.

Instead of solving a multiobjective problem directly, the MOEA/D explicitly decomposes it into NP scalar optimization subproblems. It solves these subproblems simultaneously by evolving a population of solutions. At each generation, the population is composed of the best solution found so far for each subproblem. The neighborhood relations among these subproblems are defined based on the distances between their aggregation weight vectors. Each subproblem is optimized using information from its neighboring subproblems.

Although MOEA/D framework provides a very natural framework for using single-objective search techniques, it is crucial when and how a single-objective search technique can be used in MOEA/D (Ke et al. 2013). This study combines the advanced features from both the DE for single-objective FAP (Salman et al. 2010) and MOEA/D, and proposes a hybrid MOEA/D with subproblem-dependent heuristic assignment, called MOEA/D-SHA, for the MO-FAP.

3.1 MOEA/D-SHA framework

Instead of solving the MO-FAP defined by Eq. (4) directly, the MOEA/D-SHA explicitly decomposes it into NP scalar

Algorithm 1 MOEA/D framework

Input:

- 1) a multiobjective problem instance;
- 2) a stopping criterion;
- 3) the number of decomposed subproblems, NP , also called population size;
- 4) a number of uniformly distributed weight vectors $\lambda^1, \dots, \lambda^{NP}$;
- 5) the size of neighborhood of each subproblem denoted as T .

Output: A set of solutions NDS in both decision and objective space.

Step 1: Initialization.

- 1) Set $NDS = \emptyset$.
- 2) Decompose the multiobjective problem into NP scalar optimization subproblems with $\lambda^1, \dots, \lambda^{NP}$.
- 3) Compute the Euclidean distance between each pair of weight vectors and get the T closest weight vectors to each weight vector. For each subproblem $i = 1, \dots, NP$, set the neighborhood $B(i) = \{i_1, \dots, i_T\}$, where $\lambda^{i_1}, \dots, \lambda^{i_T}$ are the T closest weight vectors to λ^i .
- 4) Initialize a population $POP = \{X^1, \dots, X^{NP}\}$.

Step 2: Reproduction.

for $i = 1, \dots, NP$, **do**

- 1) Randomly select two different neighborhood indexes from $B(i)$.
- 2) Generate a new solution Y^i from the selected individuals and X^i by using genetic operators.
- 3) Apply a problem-specific repair/improvement heuristic on Y^i to improve it.

Step 3: Update.

for $i = 1, \dots, NP$, **do**

- 1) Remove all the solutions dominated by $F(Y^i)$ from NDS, and add $F(Y^i)$ to NDS if no solutions in NDS dominates $F(Y^i)$.
- 2) Replace X^i by Y if $g(Y|\lambda^i) < g(X^i|\lambda^i)$, where Y is the solution with the smallest $g(\cdot|\lambda^i)$ value among all the new solutions that produced by subproblem i 's neighbors and have not been used for updating any other old solutions.

Step 3: Stopping criterion: If stopping criterion is met, stop the algorithm. Otherwise, go to **Step 2**.

optimization subproblems. The framework of the MOEA/D-SHA for the MO-FAP is described as follows.

In response to the particularities of each problem, different multiobjective evolutionary algorithms differ in the coding scheme (responsible for the characterization of the search space), objective function, and also operators that depend on the kind of coding scheme adopted. As a consequence, the MOEA/D-SHA is not the same as the MOEA/D of previous papers (Mei et al. 2011; Konstantinidis and Yang 2011a,b; Li and Landa-Silva 2011; Ke et al. 2013; Shim et al. 2012). It employs the same framework, but is composed of dedicated modules, for example, SHA procedure, to solve the MO-FAP. The adaptation of the MOEA/D framework to the MO-FAP is described in detail in the following subsection. The main components of the MOEA/D-SHA include the decom-

position, solution representation, reproduction operator and SHA.

3.2 Decomposition

MOEA/D-SHA decomposes the MO-FAP into NP single-objective subproblems using weighted-sum approach, thus the i th subproblem is defined as:

$$\text{ming}(X^i|\lambda^i) = \lambda_1^i f_L(X^i) + \lambda_2^i f_T(X^i) \quad (7)$$

where $\lambda^i = (\lambda_1^i, \lambda_2^i)$ is the weight vector of subproblem $i = 1, \dots, NP$ and $\lambda_1^i + \lambda_2^i = 1$. $g(X^i|\lambda^i)$ is used to emphasize that λ^i is a coefficient in this objective function while X^i is the variable to be optimized. The weighted-sum approach described in Eq. (7) considers a convex combination of the

Algorithm 2 MOEA/D-SHA for MO-FAP

Input:

1) a MO-FAP instance, and other parameters as in MOEA/D framework.

Output: A set of solutions NDS in both decision and objective space.

Step 1: Initialization.

- 1) Decompose the MO-FAP into NP scalar optimization subproblems with $\lambda^1, \dots, \lambda^{NP}$ by using the weighted sum approach.
- 2) Initialize a population $POP = \{X^1, \dots, X^{NP}\}$ randomly or by problem-specific methods.
- 3) Other steps are as in MOEA/D.

Step 2: Reproduction based on mutation and crossover of DE.

for $i = 1, \dots, NP$, **do**

- 1) Randomly select three different neighborhood indexes from $B(i)$.
- 2) Generate a new solution Y^i from three selected individuals and X^i by using DE mutation and crossover operators defined by Eqs.(5)-(6).
- 3) Apply **subproblem-dependent heuristic assignment (SHA)** procedure (Section 3.4) to Y^i to construct an assignment matrix. If the assignment matrix is infeasible, apply **simple assignment procedure (Fig.5)** to Y^i to construct a feasible assignment matrix. Then, evaluate the objective vector $F(Y^i)$ from the assignment matrix.

Step 3: Update as in MOEA/D.

Step 3: Stopping criterion: If stopping criterion is met, stop the algorithm. Otherwise, go to **Step 2**.

different objectives. In this study, we consider a uniform spread of the weights which is remain fixed for each subproblem i during the whole evolution. These weights are determined as follows:

$$\lambda^i = \left(\frac{i-1}{NP-1}, \frac{NP-i}{NP-1} \right).$$

For example, given 5 subproblems, the weights for each subproblems are (0,1), (1/4,3/4), (2/4,2/4), (3/4,1/4), (1,0), respectively. In Fig. 4, some weights including (0,1), (2/4,2/4), and (1,0) are shown to be corresponding to some search directions in the objective space. How to set these weights can also be found in (Zhang and Li 2007).

In the above definition, two objectives f_L and f_T are of different scales, and direct use of them will make the algorithm bias more to objective f_T . Therefore, normalization is required. The maximal and minimal values among the largest (total) interference of all feasible solutions that have been found so far are used for normalization as in (Mei et al. 2011; Konstantinidis and Yang 2011a,b). It is argued that MOEA/D faces difficulty when the objective functions have not the same scale for the problems with many objectives (Deb and Jain 2014). However, for the MO-FAP with only

two objectives, the normalization method considered in this study can get good results as shown in previous research (Mei et al. 2011; Konstantinidis and Yang 2011a,b).

In general, the weight vector λ^i is mainly used for decomposing a multiobjective problem into single-objective subproblems by adding different weights to the objectives. This parameter can also be given a problem-specific meaning as follows (Konstantinidis and Yang 2011a,b): the weight vector λ^i of a subproblem i , can be used to predict the objective preference of particular design X^i and therefore its position and search direction in the objective space. For example, the extreme solutions X^A and X^B in Fig. 4 focus on optimizing one objective each, respectively. That is, X^A aims at minimizing the largest interference objective ignoring total interference, while X^B aims at minimizing the total interference objective ignoring largest interference. The goal of the MO-FAP is to obtain a trade-off set of solutions between the extreme solutions, providing the interested users with a diverse set design choices (for example, X^a , X^b , X^c in Fig. 4) to facilitate decision making. Thus, single-objective heuristics by extracting knowledge from the problem domain and the objectives properties can be designed and strategically controlled by MOEA/D-SHA to optimize differ-

Fig. 5 Simple assignment procedure **a** Representation for carrier's priorities and **b** decoding to FAP solution (assignment)

C21	C22	C23	C24
0	3	2	1

(a)

Sorting		S11	S12	S13	S14	S15	S16
C22	3						
C23	2						
C24	1						
C21	0						

(b)

ent areas of the objective space (e.g., a , b and c in Fig. 4) (Konstantinidis and Yang 2011a,b). The problem-specific heuristics will be described later in details, and the objective preference of each subproblem indicated by the weight vector λ^i will be used to design subproblem-dependent heuristics. This beneficial procedure cannot be utilized by any nondecompositional MOEA framework (Konstantinidis and Yang 2011a,b).

3.3 Solution representation and reproduction

In this study, the priority-based representation is adopted (Salman et al. 2010). In this representation, each carrier is associated with a value that represents its priority, which is described by Fig. 5a. Thus, each individual X is represented by an array of size N (number of carriers), i.e., $X = (x_1, \dots, x_N)$, where the value of j th gene of the individual specifies the priority associated with j th carrier in system 2.

Each gene is initialized with a value between $[0, N]$ representing the carrier's priority. The initial population is generated randomly with the aim of covering the entire search space. Each individual is constructed as follows: First, a carrier is selected randomly and assigned the number of carriers N as its priority value. Then, a different carrier is chosen at random and assigned the value of $N - 1$ as its priority value. This process is repeated until all the carriers have been assigned different priority values.

For each individual, all the carriers are ranked or sorted by descending order according to their priority values. Then, according to the carrier's order, segments from system 1 are assigned to carriers in system 2 in a consecutive manner using a simple assignment procedure illustrated by Fig. 5b. Detailed implementation of simple assignment procedure can also be found in (Salman et al. 2010). The simple assignment procedure can always produce feasible solution.

The MOEA/D-SHA generates new solutions using the DE mutation and crossover operators defined by Eqs. (5)–(6).

3.4 SHA

The SHA procedure consists of two main components: modification of priority value and selection of heuristic strategy. The SHA introduces more problem-specific heuristic information and thus can be seen as an advanced or sophisticated version of simple assignment procedure.

In the MO-FAP, large carriers with many segments should be assigned as early as possible, otherwise, it would be difficult to assign them after many carriers have been already assigned (Funabiki and Nishikawa 1997; Wang et al. 2008, 2011). Thus, the priority value in the individual representation should consider the carrier length as problem-specific heuristic information. Let c_j represents the number of segments or the length of carrier j , and priority value of j th gene, x_j , is modified according to the following rule:

$$x_j = \begin{cases} (x_j/N) \cdot \max_{1 \leq k \leq N} (c_k) + c_j & \text{if } r < h_M, \\ c_j \cdot \log_{M/N}(x_j) & \text{otherwise.} \end{cases} \quad (8)$$

where r ($\in(0, 1)$) is a random number. The rule Eq. (8) includes two ways to modify the priority value, where the probability of choosing one of the ways is controlled by a parameter h_M . h_M is initialized as 0.5, and then adjusted adaptively during the search. Specifically, $h_M = \text{Suc1}/(\text{Suc1} + \text{Suc2})$, where Suc1 and Suc2 represent the success rate of constructing a feasible assignment matrix using the first and second ways within the previous learning period (LP) generations, respectively.

Based on the priority-based representation, the segments from system 1 are assigned to each carrier using one of the following three heuristics (H1, H2 and H3) (Liu et al. 2007):

H1: From the available consecutive segments in system 1 (i.e., not been assigned yet), select the ones which produce the lowest largest interference with current carrier in system 2.

	S11	S12	S13	S14	S15	S16
C21	20	20	40	0	25	25
C22	50	30	30	15	55	*
C23	30	30	45	0	35	35
C24	45	25	25	10	50	*

(a)

	S11	S12	S13	S14	S15	S16
C21	20	20	40	0	25	25
C22	100	40	30	15	110	*
C23	30	20	45	0	15	35
C24	90	30	25	10	100	*

(b)

Fig. 6 **a** Largest interference cost and **b** total interference cost of each carrier using interference matrix in Fig. 3b

H2: From the available consecutive segments in system 1, select the ones which produce the lowest total interference with current carrier in system 2.

H3: From the available consecutive segments in system 1, select the ones which produce both the lowest largest interference and the lowest total interference with current carrier in system 2 (i.e., this is calculated by giving a weight for each objective).

For these heuristics, two new matrices, namely, the largest interference cost matrix (LI) and the total interference cost matrix (TI), should be firstly generated from the main interference matrix (i.e., Fig. 3b). The element l_{ij} in the largest interference cost matrix is given by the largest element among c_i elements for assigning carrier i to segments from j to $(j + c_i - 1)$. That is, $l_{ij} = \max\{e_{kj}, \dots, e_{k+c_i-1, j+c_i-1}\}$, where k is the first segment number of carrier i in the $M \times M$ interference matrix F (Wang et al. 2008, 2011). Figure 6a shows the largest interference cost matrix derived from the interference matrix described by Fig. 3b. If the carrier length for carrier i is 1, i.e., $c_i = 1$, then row i for carrier i in the cost matrix is the same as the row in the interference matrix for carrier i . For example, $c_1 = 1$ and $c_3 = 1$, therefore, 1-st and 3-rd rows of the cost matrix is the same as the C_{21} and C_{23} rows of the interference matrix. If $c_i > 1$, then we choose the largest value in the diagonal line (dashed line in the interference matrix described

by Fig. 3b) for each j . For example, $c_2 = 2$ and $c_4 = 2$, then the maximum value of the crossed elements on the diagonal line in the C_{22} and C_{24} rows is selected as the corresponding value in the largest interference cost matrix (Wang et al. 2008, 2011). The total interference cost matrix can be generated in a similar way. The only difference is that we should sum up all the values in the diagonal line (dashed line in the interference matrix described by Fig. 3b) as the total interference cost element for each j in the case of $c_i > 1$. Figure 6b shows the total interference cost matrix derived from the interference matrix described by Fig. 3b.

Then, one of the subproblem-dependent heuristics (H1, H2 or H3) is adaptively selected according to the objective preference of each subproblem indicated by the weight vector as mentioned in Sect. 3.2. The complete SHA procedure can be described as follows.

Algorithm 3 SHA

Input:

- 1) a solution $X = (x_1, \dots, x_N)$.
- 2) the weight vector of the corresponding subproblem $\lambda = (\lambda_1, \lambda_2)$.
- 3) the carriers length $C = (c_1, \dots, c_N)$.

Output: An assignment matrix.

Step 1: Modification of priority value.

- 1) Generate a random number $r \in (0, 1)$.

2) **for** $j = 1, \dots, N$, **do**

Modify the priority value of carrier j , x_j , according to Eq.(8).

Step 2: Sorting all the carriers in decreasing order according to their priority values.

Step 3: Selection of heuristic strategy according to the objective preference of each subproblem indicated by the weight coefficient.

if $(\lambda_1 > 0.5)$

$$H = \begin{cases} H3 & \text{if } \text{rand}() < (1 - \lambda_1) / 0.5, \\ H1 & \text{otherwise.} \end{cases}$$

else

$$H = \begin{cases} H3 & \text{if } \text{rand}() < (1 - \lambda_2) / 0.5, \\ H2 & \text{otherwise.} \end{cases}$$

Step 4: Heuristic assignment according to the selected heuristic to construct an assignment matrix.

In the SHA, the selection of heuristic for assignment takes the objective preference of each subproblem (i.e., the weighted vector λ) into consideration. If $\lambda_1 > 0.5$, it means that the subproblem focuses more on (or biases towards) optimizing the first objective (i.e., f_L), and thus H1 or H3 will be selected for assignment. This case mainly favors the solutions of area a and b depicted by Fig. 4. Or else, it means that the subproblem focuses more on optimizing the second objective (i.e., f_T), and thus H2 or H3 will be chosen for assignment. This case mainly favors the solutions of area b and c depicted by Fig. 4. The detailed implementation of heuristic assignment is given as follows.

Algorithm 4 Heuristic assignment

Input:

- 1) the carriers in decreasing order according to their priority values, denoted as carrier π_1, \dots , carrier π_N .
- 2) the largest interference cost matrix $LI = [l_{ij}]$.
- 3) the total interference cost matrix $TI = [t_{ij}]$.
- 4) the weight vector of the corresponding subproblem $\lambda = (\lambda_1, \lambda_2)$.

Output: An assignment matrix.**for** $j = 1, \dots, N$, **do****Step 1: Find available segments.**

Find free consecutive segments whose length is no fewer than the length of carrier π_j , or else return *fail*.

Step 2: Assignment.1) **for** $s = 1, \dots, M$, **do**

if segment s is the first segment of the available consecutive segments,

if s belongs to the first available consecutive segments, set $flag = s$;

else

if (H = H1) //focus on the first objective (largest interference)

if $l_{j,s} < l_{j,flag}$, set $flag = s$;

else if ($t_{j,s} = t_{j,flag}$) and ($t_{j,s} < t_{j,flag}$) and ($\text{rand}() < 1 - p_H$), set $flag = s$.

if (H = H2) // focus on the second objective (total interference)

if $t_{j,s} < t_{j,flag}$, set $flag = s$;

else if ($t_{j,s} = t_{j,flag}$) and ($l_{j,s} < l_{j,flag}$) and ($\text{rand}() < 1 - p_H$), set $flag = s$.

if (H = H3) //focus on the trade-off objective space

if $(\lambda_1 l_{j,s} + \lambda_2 t_{j,s}) < (\lambda_1 l_{j,flag} + \lambda_2 t_{j,flag})$ and ($\text{rand}() < p_H$), set $flag = s$.

2) Carrier π_j is assigned to the consecutive segments from segment $flag$ to $(flag + c_{\pi_j} - 1)$ in the assignment matrix.

In Step 1, current carrier locates free consecutive segments whose length is no fewer than the length of current carrier. In Step 2, current carrier checks every first segment of the available consecutive segments whether or not to be assigned using the specific condition according to the selected heuristic assignment. Note that a parameter p_H is used to control the probability of assignment in Algorithm 4. If p_H is set to 1, H3 will be executed in a completely greedy way, while H1 and H2 will focus on optimizing one objective ignoring the other. This may lead to local convergence. To avoid trapping into local minima, p_H is set to be less than 1, for example 0.9 in this study, for introducing some randomness.

3.5 Comparisons between MOEA/D-SHA and DE

Although both the single-objective DE (Salman et al. 2010) and the MOEA/D-SHA use the similar weighted aggregation of two objectives defined by Eqs. (1) and (7), respectively, they evolve in totally different way in solving the FAP. The MOEA/D-SHA optimizes multiple single-objective optimization subproblems with multiple fixed but uniform weight

vectors in parallel at each generation. The diversity of the solutions is maintained through these preselected weight vectors. Further, it exploits the neighborhood relationship among the subproblems for making its search effectively and efficiently. Thus, the MOEA/D-SHA maintains a set of solutions (subproblems) which is expected to provide a good approximation to the Pareto optimal front or Pareto optimal set of problem in each run. In contrast, the single-objective DE only optimizes a single-objective optimization problem and finally returns a unique solution in each run.

Furthermore, only one fixed problem-specific heuristic is used in DE. However, in the MOEA/D-SHA, one of the subproblem-dependent heuristics (H1, H2 or H3) is adaptively selected according to the objective preference of each subproblem indicated by the weight coefficients.

4 Simulation results

Experimental studies have been carried out to evaluate the efficacy of our proposed approach. All simulations were implemented in C on a PC (Intel Core2 2.4 GHz, 2 GB

Table 1 Specifications of the eight benchmark instances (BM1–BM8) and the case parameters to generate random instances with large size

Group	Benchmark or case	Number of carriers N	Number of M segments	Range of carrier length	Range of interference
Group 1	BM 1	4	6	1–2	5–55
	BM 2	4	6	1–2	1–9
	BM 3	10	32	1–8	1–10
	BM 4	10	32	1–8	1–100
	BM 5	10	32	1–8	1–1,000
	BM 6	18	60	1–8	1–50
	BM 7	30	100	1–8	1–100
	BM 8	15	50	1–7	1–1,000
Group 2	Case 9	50	200	1–10	1–10
	Case 10	50	200	1–10	1–100
	Case 11	50	200	1–10	1–1,000
Group 3	Case 12	80	200	1–5	1–100
	Case 13	80	300	1–8	1–100
	Case 14	80	400	1–10	1–100
	Case 15	80	450	1–12	1–100
	Case 16	80	500	1–14	1–100
	Case 17	80	600	1–16	1–100
	Group 4	Case 18	100	500	1–10
Case 19		150	400	1–5	1–100
Case 20		200	300	1–2	1–100

RAM). The goals of our experimental studies are to (1) study the effect of the proposed SHA on the performance of the MOEA/D-SHA, (2) test the strength of the MOEA/D-SHA against the state-of-the-art in Pareto dominance multiobjective evolutionary algorithms, i.e., NSGA-II and (3) compare with several state-of-the-art single-objective approaches to evaluate whether the proposed multiobjective evolutionary algorithm can provide any advantage over the single-objective approaches.

4.1 Experimental setup

We tested the MOEA/D-SHA on four groups with a total of 20 cases as those in (Wang et al. 2008, 2011). The details of the cases or groups are shown in Table 1 and described as follows (Wang et al. 2008, 2011).

The first group includes 8 benchmark instances (named BM1–BM8) which are widely used in literature (Kurokawa and Kozuka 1993; Funabiki and Nishikawa 1997; Liu et al. 2007; Wang et al. 2008, 2011; Salcedo-Sanz et al. 2004; Salcedo-Sanz and B-Calz'ón 2005; Salman et al. 2010). To evaluate the performance of the MOEA/D-SHA in large size MO-FAPs, Groups 2–4 are randomly generated. According to the method and case parameters in (Wang et al. 2008,

2011), these instances are generated in the following steps: (1) Choose the number of carriers N and the number of segments M for the instance. (2) Select the values of the range of carrier length and the range of interference between the two systems. To test the scalability of the MOEA/D-SHA, these two parameters are gradually increased. (3) Generate a set of carrier lengths c_i ($i = 1, \dots, N$) and interference matrix E ($M \times M$) using uniformly distributed random values in the range defined in the step above.

According to the steps above, Group 2 (Case 9–11) is designed to observe the influence of the magnitude of the interference by varying the interference from 1–10 to 1–1,000. The number of carriers N , the number of segments M , and the range of carrier length are fixed to 50, 200 and 1–10, respectively. Group 3 (Case 12–17) is designed to show the effects of the carrier length. In this group, the range of carrier length changes from 1–5 to 1–16. Group 4 (Case 18–20) is designed to show the ability of the multiobjective evolutionary algorithm to deal with a large number of carriers up to 200. These instances are the largest size problems tested by the existing algorithms so far (Wang et al. 2011).

The parameters of the MOEA/D-SHA are empirically set as: population size $NP = 10N$ for BM1–8 instances and $NP = 500$ for Case 9–20; neighborhood size $T = 20$; the maximum generation $G = 500$; $F = 0.2$ and $CR = 0.9$ as in DE for the

FAP (Salman et al. 2010); LP = 10 and $p_H = 0.9$ in the SHA. These settings may be not optimal parameter settings, but the MOEA/D-SHA with these parameters performs fairly well over the test instances. From the empirical results of preliminary experiments, we found that the MOEA/D-SHA performed equally well with small changes of parameter values, for example, LP = 20 or 30; $p_H = 0.7$ or 0.8.

4.2 Performance measures

The performance of a multiobjective evolutionary algorithm is usually evaluated from two aspects, i.e., proximity (convergence) and diversity (distribution). Proximity means that the obtained NDS should be as close to the true Pareto optimal front as possible. Diversity means that the solutions in the obtained NDS should be distributed as diversely and uniformly as possible. The two aspects can hardly be reflected by a single metric, thus a number of metrics have been suggested in the literature (Deb et al. 2002; Czyzszak and Jaskiewicz 1998; Zitzler et al. 2002; Zitzler and Thiele 1999). In this study, the following five metrics are used.

1. Distance from reference set (I_D): This metric is proposed by Czyzszak and Jaskiewicz (1998). It is defined as follows:

$$I_D(A) = \frac{\sum_{Y \in R} \{\min_{X \in A} \{d(X, Y)\}\}}{|R|}.$$

Given a set A , $I_D(A)$ provides information about the average distance from a solution in the reference set R to the closest solution in A . In our experiment, the NDS obtained by all the algorithms in 30 runs on a test instance are combined, and those solutions remained nondominated in this set are used as the reference set. A smaller value of $I_D(A)$ indicates that A is closer to R and thus is preferable.

2. Spread (I_Δ): This metric is proposed by Deb et al. (2002). It can be stated as follows:

$$I_\Delta(A) = \frac{d_f + d_l + \sum_{i=1}^{n-1} |d_i - \bar{d}|}{d_f + d_l + (n-1) \times \bar{d}},$$

where d_f and d_l are the Euclidean distances between the leftmost and rightmost solutions of the Pareto optimal front and the extreme solutions in A . n is the number of solutions in A and d_i is the Euclidean distance between the i th left and the $(i+1)$ th left solutions in A . \bar{d} stands for the average over all d_i 's. I_Δ is an indicator of the distribution of solutions. A smaller I_Δ indicates that the solutions are distributed more uniformly and have a better extent. The leftmost and rightmost solutions among

the NDS obtained by all the 30 runs of the compared algorithms are used to compute I_Δ in our experiments.

3. Hypervolume (I_H): This metric, proposed by Zitzler et al. (2002), indicates the area in the objective space that is dominated by at least one solution of the NDS. I_H reflects the closeness of the NDS to the Pareto optimal front. The larger the I_H , the closer to the Pareto optimal front the corresponding NDS is. Also, the NDS obtained by all the algorithms are approximately seen as Pareto optimal front.
4. Coverage metric (I_C): it is a commonly used metric for comparing two sets of NDS A and B , originally proposed by Zitzler and Thiele (1999). The I_C , also denoted as $C(A, B)$ metric calculates the ratio of the NDS in B dominated by the NDS in A , divided by the total number of NDS in B . It is defined as follows:

$$C(A, B) = \frac{|\{X \in B | \exists Y \in A : Y \succ X\}|}{|B|}.$$

Therefore, $C(A, B) = 1$ means that all NDS in B are dominated by the NDS in A . Conversely, $C(A, B) = 0$ means that all NDS in A are dominated by the NDS in B . Note that the sum of $C(A, B)$ and $C(B, A)$ is not always equal to 1, because there may exist some solutions in A and B that are not dominated by each other.

5. Nondominated solutions (NDS(A)): it is the cardinality or the number of NDS in set A , which is adopted in (Konstantinidis and Yang 2011a,b). It is more desirable to obtain a high number of NDS(A) to provide an adequate number of Pareto optimal choices in cases of real-life discrete optimization problems such as the FAP. In contrast, in cases of continuous optimization (Konstantinidis and Yang 2011a,b), a high number of NDS is not desirable, since the decision-making procedure becomes more complicated and more time-consuming. Note that the NDS(A) should be considered in combination with other metrics (e.g., I_Δ and I_C), since it is usually desirable to have a high number of NDS when the solutions is of high quality (i.e., low I_C) and spread (i.e., low I_Δ) in the objective space.

In this study, all the metrics are computed based on the normalized objective vectors of the NDS. These metrics are used to measure the performances of multiobjective optimization algorithms from different aspects. I_Δ is typically intended to measuring diversity property. Metrics I_D and I_H take into account both criteria, i.e., proximity and diversity. I_C shows the dominance relationship between two compared solutions sets. More details about these metrics can be referred to (Zhou et al. 2011; Shim et al. 2012; Czyzszak and Jaskiewicz 1998; Zitzler et al. 2002; Zitzler and Thiele 1999).

Table 2 Average value of I_D , I_Δ , and I_H of the MOEA/D (denoted as M), MOEA/D-H (denoted as M-H) and MOEA/D-SHA (denoted as M-S)

Instance	I_D			I_Δ			I_H			Time (s)		
	M	M-H	M-S	M	M-H	M-S	M	M-H	M-S	M	M-H	M-S
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	1.0000	1.0000	0.09	0.14	0.15
2	0.0333	0.0000	0.0000	0.0000	0.0000	0.0000	0.9666	1.0000	1.0000	0.09	0.14	0.15
3	0.4003	0.0000	0.0000	0.3199	0.0000	0.0000	0.6116	1.0000	1.0000	0.38	1.28	1.35
4	0.3606	0.0642	0.0625	0.6578	0.4721	0.4704	0.6213	0.8930	0.8939	0.38	1.28	1.37
5	0.2090	0.0859	0.0843	0.6556	0.2584	0.2416	0.6743	0.8976	0.9033	0.38	1.34	1.39
6	0.4890	0.2268	0.0678	0.7977	0.6187	0.5402	0.2813	0.6209	0.8970	1.05	6.09	6.04
7	0.6841	0.6671	0.0538	0.7896	0.8000	0.3416	0.2007	0.2180	0.9554	2.76	24.94	22.47
8	0.5147	0.1847	0.1511	0.6895	0.7602	0.8019	0.3453	0.8250	0.9224	0.76	3.69	3.76
9	0.7974	0.9671	0.0677	0.0299	0.0000	0.1463	0.0822	0.0332	0.9301	8.05	98.58	98.95
10	0.8706	0.8971	0.0720	0.9103	0.9433	0.6418	0.0408	0.0355	0.9307	8.16	98.78	99.08
11	0.8032	0.8307	0.0695	0.8811	0.9155	0.8065	0.0594	0.0559	0.9278	8.10	96.59	94.34
12	1.0416	1.0458	0.0334	0.8957	0.9603	0.5772	0.0217	0.0206	0.9839	13.53	163.12	172.19
13	1.0314	1.0325	0.0552	0.8952	0.9520	0.5819	0.0143	0.0150	0.9060	14.91	225.99	234.28
14	0.9536	0.9525	0.0804	0.9592	0.9527	0.6975	0.0181	0.0175	0.8128	16.33	296.36	276.91
15	0.9880	0.9869	0.1049	0.9510	0.9490	0.6174	0.0097	0.0102	0.8279	17.14	323.68	322.37
16	0.9161	0.9132	0.1306	0.9502	0.9136	0.6733	0.0134	0.0134	0.7616	17.94	367.22	351.61
17	0.8911	0.8901	0.1282	0.9164	0.9446	0.6380	0.0188	0.0194	0.7384	19.82	449.60	427.80
18	0.9887	0.9891	0.0856	0.9549	0.9511	0.5957	0.0068	0.0073	0.8504	22.18	447.82	449.24
19	1.1935	1.1913	0.0407	0.9749	0.9435	0.5843	0.0045	0.0047	0.9404	36.49	574.79	614.47
20	1.2705	1.2685	0.0125	0.9806	0.9775	0.5796	0.0047	0.0049	0.9791	50.60	631.56	626.19

Furthermore, to show the significant differences between the MOEA/D-SHA and competitors, we adopt several nonparametric statistical tests, i.e., Friedman test, Iman–Davenport test, Holm test and Wilcoxon test (Derrac et al. 2011). All of the nonparametric statistical tests are carried out by the KEEL software (Alcalá-Fdez et al. 2008).

4.3 Effectiveness of main components in MOEA/D-SHA

To confirm the effectiveness of the main components in the MOEA/D-SHA, and thus gain a better understanding of how and why the algorithm works, we conduct our first experiment to compare the MOEA/D-SHA with its two variants. In the first variant of the MOEA/D-SHA, the SHA procedure described in Algorithm 3 is totally removed, thus in the Step 2(3) of Algorithm 2, only simple assignment is used. This variant is denoted as MOEA/D. In the second variant, only the modification of priority value strategy (i.e., Step 1 in Algorithm 3) is removed. The heuristic strategies are still reserved and thus this variant is denoted as MOEA/D-H.

Tables 2, 3 show the average values of metrics over the 30 independent runs of the compared algorithms on the test sets. For each instance and each performance metric, the single-problem Wilcoxon rank sum test on paired algorithms (Der-

rac et al. 2011; Alcalá-Fdez et al. 2008) was further carried out on the results obtained by 30 runs of the compared algorithms, and the one that is significantly better than that of the other two (with the significance level of 5 %) is in boldface. To comprehensively evaluate the performance of the compared algorithms, the NDS obtained on selected instances by them in all 30 runs are also plotted in the objective space and shown in Fig. 7. From the quantitative results in Tables 2, 3 and graphical representation of the NDS in Fig. 7, we point out the following observations.

Firstly, we can find that the MOEA/D-SHA outperforms the MOEA/D on all instances except for small size instances (BM1–BM3) in terms of metrics I_D and I_H , on all instances except for small size instances (BM1–BM3, BM8) in terms of metric I_Δ , and on all instances except for small size instances (BM1 and BM2) in terms of metric I_C , respectively. Especially, for the large size instances, the average value of metric I_C in column C(M-S, M) is equal to 1.0, while that in column C(M, M-S) is equal to 0. This means that for these instances, all the solutions yielded by the MOEA/D are dominated by those of the MOEA/D-SHA, which also can be easily observed from Fig. 7. Moreover, the MOEA/D-SHA obtains a higher number of the NDS. These results lead to the conclusion that the MOEA/D-SHA can produce much more NDS with good proximity and diversity performances than

Table 3 Average value of I_C and NDS of the MOEA/D (denoted as M), MOEA/D-H (denoted as M-H) and MOEA/D-SHA (denoted as M-S)

Instance	NDS			I_C					
	M	M-H	M-S	C(M,M-H)	C(M-H,M)	C(M,M-S)	C(M-S,M)	C(M-H,M-S)	C(M-S,M-H)
1	1	1	1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2	1	1	1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3	1	1	1	0.0000	1.0000	0.0000	1.0000	0.0000	0.0000
4	3	4	4	0.0000	0.3333	0.0000	0.3333	0.0000	0.0000
5	4	4	4	0.0000	0.2500	0.0000	0.2500	0.0000	0.0000
6	3	8	5	0.0000	1.0000	0.0000	1.0000	0.0000	0.6250
7	5	5	4	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000
8	5	5	5	0.0000	1.0000	0.0000	1.0000	0.0000	0.0000
9	1	1	4	1.0000	0.0000	0.0000	1.0000	0.0000	1.0000
10	5	4	9	0.5000	0.4000	0.0000	1.0000	0.0000	1.0000
11	6	3	10	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000
12	6	6	4	0.8333	0.1667	0.0000	1.0000	0.0000	1.0000
13	5	5	8	0.2000	0.6000	0.0000	1.0000	0.0000	1.0000
14	4	4	12	0.7500	0.2500	0.0000	1.0000	0.0000	1.0000
15	4	3	10	0.6667	0.2500	0.0000	1.0000	0.0000	1.0000
16	3	4	14	0.5000	0.3333	0.0000	1.0000	0.0000	1.0000
17	3	3	10	0.3333	0.6667	0.0000	1.0000	0.0000	1.0000
18	3	4	15	0.2500	0.6667	0.0000	1.0000	0.0000	1.0000
19	4	3	6	0.3333	0.5000	0.0000	1.0000	0.0000	1.0000
20	5	5	5	0.2000	0.8000	0.0000	1.0000	0.0000	1.0000

the MOEA/D, which shows the positive contribution of the SHA procedure.

Secondly, the MOEA/D-SHA outperforms the MOEA/D-H on all instances except for small size instances (BM1–BM3) in terms of metrics I_D and I_H , on all instances except for small size instances (BM1–BM3, BM9) in terms of metric I_Δ , and on all instances except for small size instances (BM1–BM2, BM8) in terms of metric I_C , respectively. For the large size instances, the average value of metric I_C in column C(M-S, M-H) is equal to 1.0, while that in column C(M-H, M-S) is equal to 0. At the same time, the MOEA/D-SHA obtains a higher number of the NDS, and all the solutions yielded by the MOEA/D-H are dominated by those of the MOEA/D-SHA for these instances as shown in Fig. 7. Figure 8 shows the success rate of constructing feasible solution by the SHA with and without modification of priority value during the whole evolution process of the MOEA/D-SHA and MOEA/D-H, respectively. From Fig. 8, we can find the SHA without modification of priority value in the MOEA/D-H is very difficult to produce feasible solution or assignment on larger size instances, which severely deteriorates the performance the MOEA/D-H. These results indicate that the MOEA/D-SHA can produce much more NDS with good performances than the MOEA/D-H, which suggests the positive contribu-

tion of the modification of priority value in the SHA procedure.

Finally, the MOEA/D-H outperforms the MOEA/D on small size instances BM4–BM8 in terms of most of metrics, which indicates the positive contribution of the heuristic strategies. But the MOEA/D-H cannot consistently achieve better performance on the larger size instances, i.e., Case 9–20. The reason is that the heuristic strategies alone are very difficult to produce feasible assignment on larger size instances as shown in Fig. 8. This result suggests that combining the modification of priority value and selection of heuristic strategy is necessary for performance improvement.

It is natural that the MOEA/D-SHA and MOEA/D-H spend more time because they include the problem-specific heuristics.

To show the significant differences between the MOEA/D-SHA and its two variants, MOEA/D and MOEA/D-H, on all problems, nonparametric statistical tests (Derrac et al. 2011; Alcalá-Fdez et al. 2008), i.e., Iman–Davenport (Friedman), Holm and Wilcoxon test are adopted. First, we detect whether there are significant differences between all the algorithms for all the problems. Table 4 shows the average rankings for all the algorithms considered, and the results of Iman–Davenport (Friedman) test. The results in Table 4 demonstrate that the MOEA/D-SHA is the best performing algo-

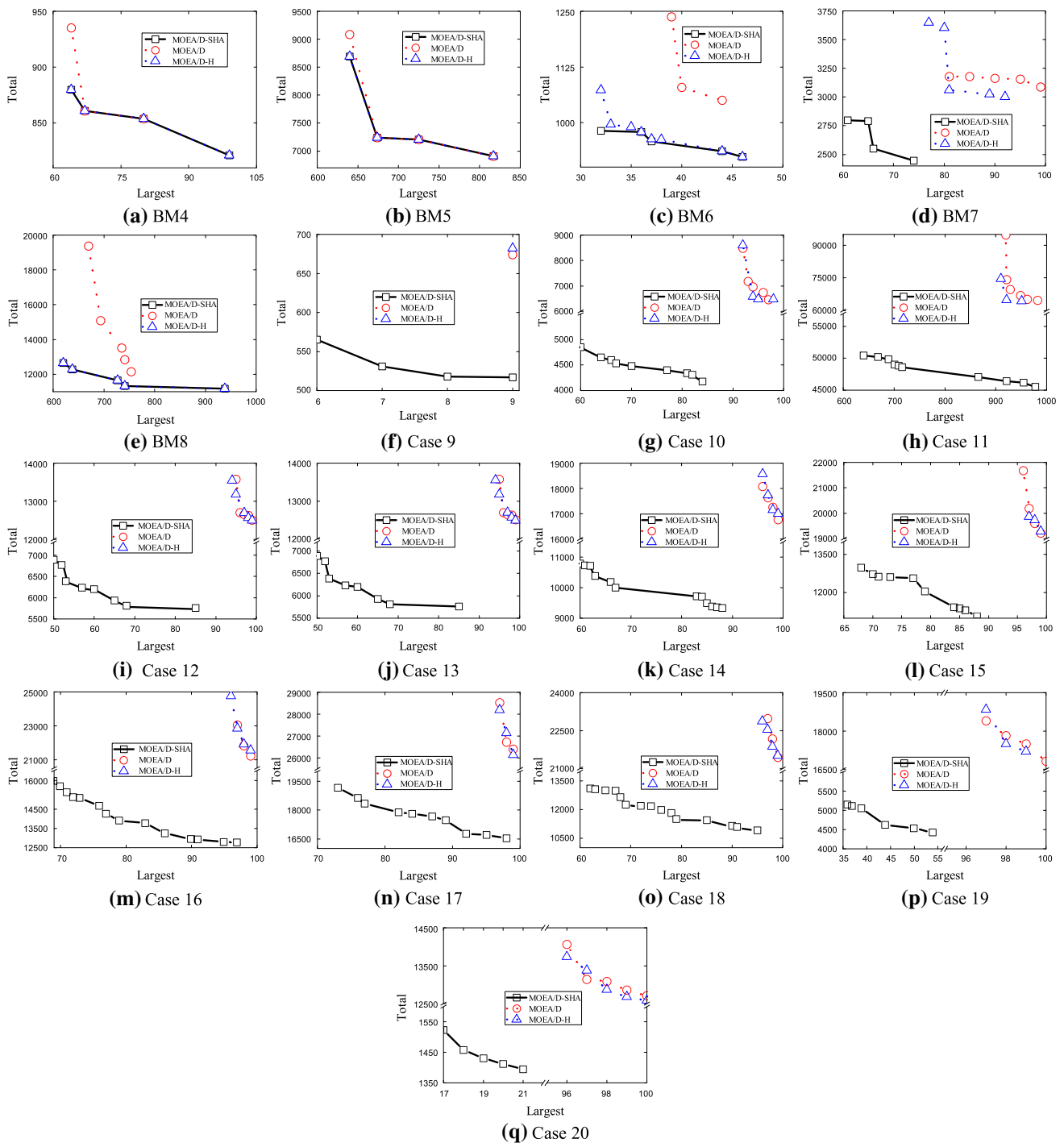


Fig. 7 Nondominated solutions obtained on the selected instances by MOEA/D-SHA and its two variants

rithm, with an average ranking of 1.10, 1.33, and 1.10 for metrics I_D , I_Δ , and I_H , respectively. Furthermore, according to the results of the Iman–Davenport (Friedman) test, we can observe that there are significant differences between the algorithms in all the cases. Second, to analyze the differences between pairs of algorithms for all the problems,

we perform the Holm test considering the MOEA/D-SHA as the control algorithm (because the MOEA/D-SHA has the best average ranking). The results of Holm test are presented in Table 5. For all the cases, the MOEA/D-SHA is statistically better than the MOEA/D and MOEA/D-H. Third, to further show the significant differences between the MOEA/D-

Fig. 8 Success rate of constructing feasible solution by the SHA with and without modification of priority value in the MOEA/D-SHA and MOEA/D-H, respectively

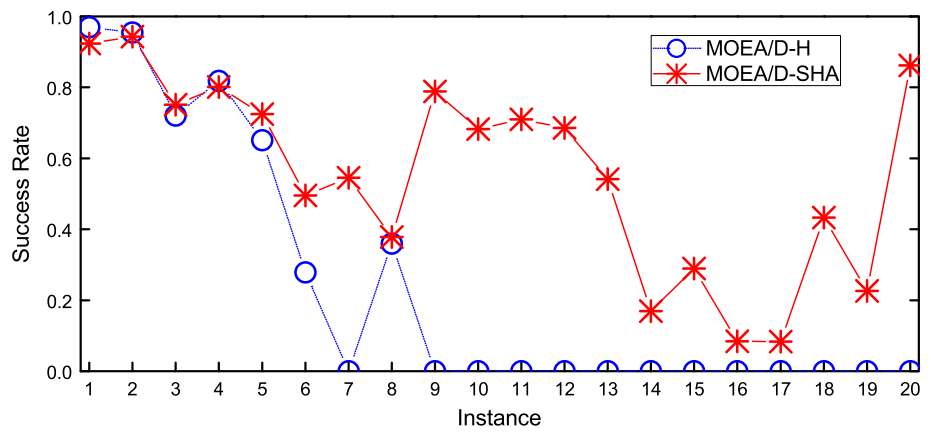


Table 4 Average rankings of MOEA/D, MOEA/D-H and MOEA/D-SHA by Friedman and Iman and Davenport test for all problems

	I_D	I_Δ	I_H
MOEA/D	2.65	2.45	2.65
MOEA/D-H	2.25	2.23	2.25
MOEA/D-SHA	1.10	1.33	1.10
p value (Friedman test)	2.00E-06	8.35E-04	2.00E-06
p value (Iman and Davenport test)	2.49E-09	2.45E-04	2.49E-09
Significant	Yes	Yes	Yes

Table 6 Results of multi-problem Wilcoxon signed-rank test for MOEA/D, MOEA/D-H and MOEA/D-SHA at $\alpha = 0.05$

Metric	Algorithm	R^+	R^-	p value	$\alpha = 0.05$
I_D	MOEA/D	190	0	1.21E-04	Yes
	MOEA/D-H	188.5	1.5	1.55E-04	Yes
I_Δ	MOEA/D	190.5	10.5	3.90E-04	Yes
	MOEA/D-H	175.5	14.5	1.12E-03	Yes
I_H	MOEA/D	190	0	1.21E-04	Yes
	MOEA/D-H	188.5	1.5	1.55E-04	Yes

Overall, these results highlight the usefulness of the individual components and their combined contribution to an overall improvement in the algorithm performance.

Table 5 Results of Holm test for MOEA/D, MOEA/D-H and MOEA/D-SHA (control algorithm)

Metric	Algorithm	z	p value	Significant
I_D	MOEA/D	4.9015	1.000E-06	Yes
	MOEA/D-H	3.6366	2.760E-04	Yes
I_Δ	MOEA/D	3.5576	3.740E-04	Yes
	MOEA/D-H	2.8461	4.427E-03	Yes
I_H	MOEA/D	4.9015	1.000E-06	Yes
	MOEA/D-H	3.6366	2.760E-04	Yes

SHA and its variants, we carry out the pairwise Wilcoxon signed-rank test, which is more powerful to identify differences between two algorithms (Derrac et al. 2011). The multi-problem Wilcoxon signed-rank test is conducted for comparing pairs of algorithms on all the problems for each metric. The results are presented in Table 6. From Table 6, it is clear that the MOEA/D-SHA obtains higher R^+ values than R^- values in all the cases. Furthermore, the p values of all the cases are all less than 0.05, which indicates that the MOEA/D-SHA is significantly better than its variants.

4.4 Comparing MOEA/D-SHA with NSGA-II

The NSGA-II serves as a representative of the traditional multiobjective evolutionary algorithms without problem-specific heuristic or local search, and represents the way of directly using an existing or off-the-shelf multiobjective evolutionary algorithm for the MO-FAP. For fair comparison, the NSGA-II also adopts the DE operators for offspring reproduction and simple assignment procedure for assignment as the MOEA/D-SHA. Two algorithms also share the same key parameters, such as population size NP , F and CR . Since the NSGA-II does not employ a problem-specific heuristic or local search process, it is assigned a larger generation number, and thus the computational time of the two algorithms is the same for fair comparison.

Table 7 shows the average values of metrics over the 30 independent runs of the compared algorithms on the test sets. The NDS obtained on the selected instances by MOEA/D-SHA and NSGA-II in all 30 runs are shown in Fig. 9. From Table 7, we can find that the MOEA/D-SHA outperforms the NSGA-II on all instances except for small size instances (BM1 and BM2) in terms of metrics I_D , I_H

Table 7 Average value of I_D , I_Δ , I_H , I_C and NDS of the MOEA/D-SHA (denoted as M-S) and NSGA-II (denoted as N)

Instance	I_D		I_Δ		I_H		I_C		NDS	
	M-S	N	M-S	N	M-S	N	C(M-S,N)	C(N,M-S)	M-S	N
1	0.0000	0.0000	0.0000	0.0000	1.0000	1.0000	0.0000	0.0000	1	1
2	0.0000	0.0000	0.0000	0.0000	1.0000	1.0000	0.0000	0.0000	1	1
3	0.0000	0.4645	0.0000	0.2642	1.0000	0.5194	1.0000	0.0000	1	1
4	0.0347	0.3479	0.4846	0.6095	0.9363	0.5644	0.6667	0.0000	4	3
5	0.0854	0.3426	0.2365	0.5853	0.8991	0.4568	1.0000	0.0000	4	4
6	0.0611	0.4857	0.5425	0.7643	0.9059	0.2743	1.0000	0.0000	5	6
7	0.0615	0.6724	0.3392	0.8344	0.9481	0.2048	1.0000	0.0000	4	4
8	0.1407	0.4750	0.8170	0.7676	0.9360	0.3597	1.0000	0.0000	5	4
9	0.1073	0.9751	0.1549	0.0280	0.8615	0.0006	1.0000	0.0000	4	2
10	0.0769	0.6667	0.6387	0.8665	0.9232	0.1262	1.0000	0.0000	9	4
11	0.0779	0.6625	0.8038	0.9294	0.9136	0.1163	1.0000	0.0000	10	7
12	0.0802	1.0229	0.5769	0.8885	0.9735	0.0490	1.0000	0.0000	4	5
13	0.0618	0.8500	0.5535	0.9210	0.8811	0.0272	1.0000	0.0000	8	6
14	0.0884	0.7842	0.6921	0.9255	0.7987	0.0446	1.0000	0.0000	12	7
15	0.1089	0.7688	0.6153	0.8909	0.8228	0.0477	1.0000	0.0000	10	4
16	0.1365	0.7384	0.6696	0.8458	0.7542	0.0445	1.0000	0.0000	14	5
17	0.1467	0.7623	0.6297	0.8778	0.7074	0.0291	1.0000	0.0000	10	5
18	0.0905	0.7809	0.5925	0.8936	0.8437	0.0441	1.0000	0.0000	15	7
19	0.0422	0.9786	0.5832	0.9736	0.9391	0.0423	1.0000	0.0000	6	7
20	0.0127	1.0235	0.5783	0.8101	0.9789	0.0563	1.0000	0.0000	5	3

and I_C , and on all instances expect for small size instances (BM1, BM2, BM8, and BM9) in terms of metric I_Δ , respectively. For the large size instances, the average value of metric I_C in column C(M-S, N) is equal to 1.0, while that in column C(N, M-S) is equal to 0. Further, the MOEA/D-SHA obtains a higher number of the NDS on the most of instances except for small size instances, and all the solutions produced by the NSGA-II are dominated by those of the MOEA/D-SHA for these instances as shown in Fig. 9.

Since the pairwise Wilcoxon signed-rank test is more powerful to identify differences between two algorithms on all problems (Derrac et al. 2011). The multi-problem Wilcoxon signed-rank test is conducted for comparing the MOEA/D-SHA and NSGA-II on all the problems for each metric. The results are presented in Table 8. From Table 8, it is clear that the MOEA/D-SHA obtains higher R^+ values than R^- values in all the cases. Furthermore, the p values of all the cases are all less than 0.05, which indicates that the MOEA/D-SHA is significantly better than the NSGA-II.

The quantitative results in Tables 7, 8 and graphical representation of the NDS in Fig. 9 lead to the conclusion that the MOEA/D-SHA is more efficient and effective than the NSGA-II. The reason is that the NSGA-II, as most multiob-

jective evolutionary algorithms based on Pareto dominance, tries to deal with the multiobjective problem as a whole and without any problem-specific heuristic as a black box method (Konstantinidis and Yang 2011a). This results in difficulties to explore the search space efficiently. The decomposition nature of the MOEA/D-SHA, on the other hand, alleviates this difficulty by decomposing the multiobjective problem into a population of single-objective subproblems, allowing the incorporation of single-objective problem-specific heuristic (e.g., the SHA) in a simple manner. This directs the search into good feasible regions of the objective space, improving the convergence and diversity of the MOEA/D-SHA as shown in Fig. 9.

4.5 Comparing MOEA/D-SHA with single-objective approaches

Previous results have shown that the proposed multiobjective optimization algorithm can find simultaneously a set of alternative trade-off solutions. Some researchers (Jaszkiewicz 2003; Lacomme et al. 2006) coming from the combinatorial optimization community have claimed that the authors of multiobjective algorithms should prove to be competitive to single-objective algorithms in terms of quality of solutions and computational efficiency. Hence, the MS-

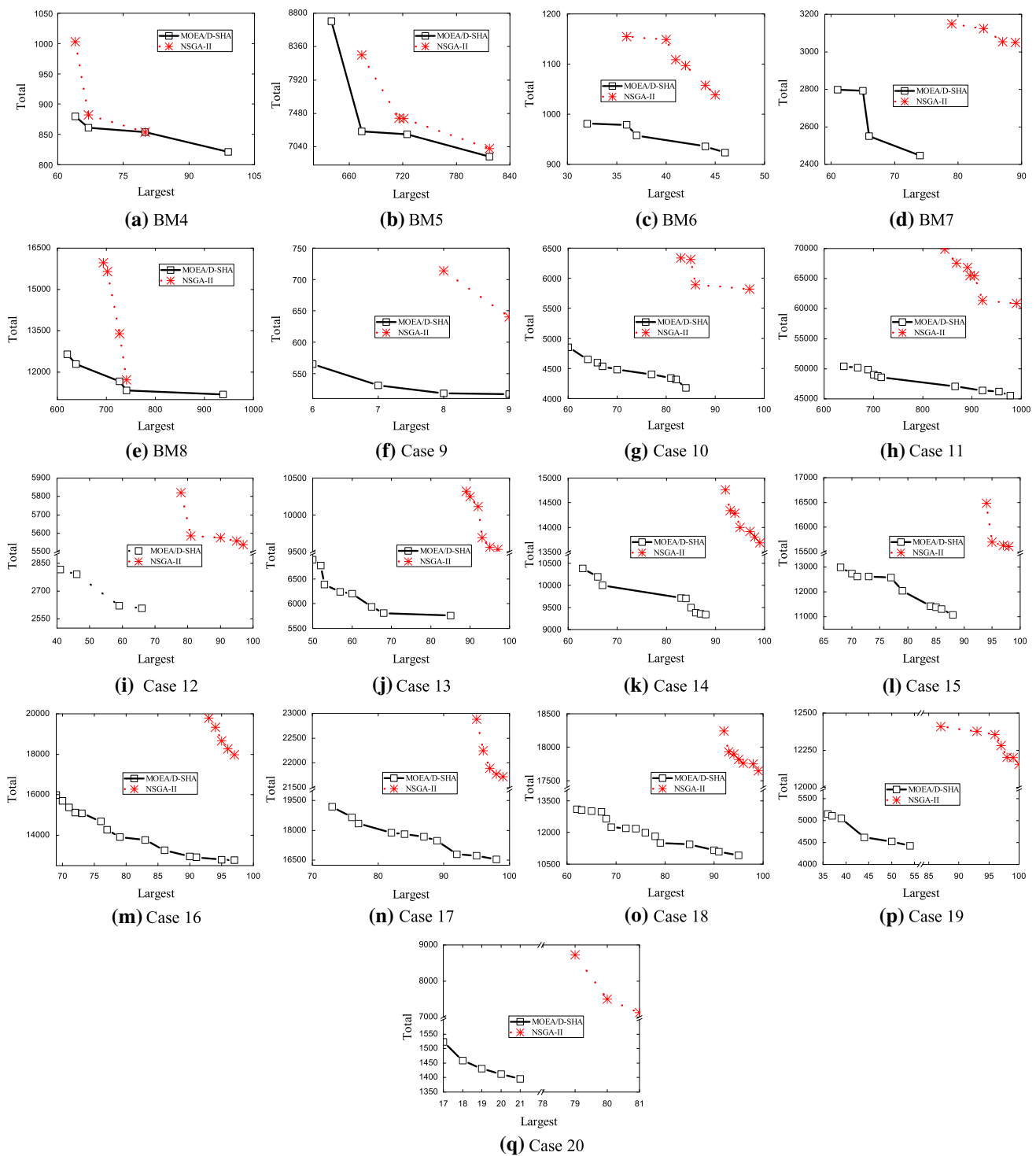


Fig. 9 Nondominated solutions obtained on the selected instances by MOEA/D-SHA and NSGA-II

SCHNN (Wang et al. 2011) and DE (Salman et al. 2010), as the best state-of-the-art single-objective approaches for the FAP, are selected for comparison. As in (Lacomme et al. 2006), the leftmost or rightmost (extreme solutions) produced by the MOEA/D-SHA are chosen to compare with the best solutions produced by the single-objective algo-

rithms. The aim of this comparison is to show that the proposed multiobjective algorithm not only finds simultaneously a set of alternative trade-off solutions, but also manages to achieve almost the same or better extreme solutions than those obtained by single-objective algorithms.

Firstly, we focus on comparing the MOEA/D-SHA with the single-objective algorithms concentrating on achieving minimum largest interference. In the MS-SCHNN, only the largest interference is encoded into the energy function to be minimized. The configurations of the MS-SCHNN follow the original paper (Wang et al. 2011) to guarantee its best performance. The DE with H1 (called DEH1), and DE with H3 (called DEH3/L) obtaining solution with low largest interference are selected since they show the best performance on the benchmark instances among all the six DE versions in (Salman et al. 2010). The DEH1, DEH3/L, and MOEA/D-SHA share the same key parameters, such as population size NP, maximum generation number G , F and CR, and thus the computational time of these algorithms is the same for fair comparison.

Table 8 Results of multi-problem Wilcoxon signed-rank test for NSGA-II and MOEA/D-SHA at $\alpha = 0.05$

Metric	Algorithm	R^+	R^-	p value	$\alpha = 0.05$
I_D	NSGA-II	208.5	1.5	1.03E-04	Yes
I_Δ	NSGA-II	199.5	10.5	3.90E-04	Yes
I_H	NSGA-II	208.5	1.5	1.03E-04	Yes

Table 9 shows the nondominated solutions with the minimum largest interference obtained by MOEA/D-SHA and the best solutions obtained by the MS-SCHNN, DEH1 and DEH3/L, respectively. The computational time of the algorithms of the MS-SCHNN and MOEA/D-SHA is also given. Note that the computational time of the DE-based algorithm is the same as that of the MOEA/D-SHA, since they share the same key parameters for fair comparison. For each instance, the solution dominating the others is in boldface. From Table 9, it can be observed that the MOEA/D-SHA obtains solutions dominating those obtained by the MS-SCHNN on 13 out of 20 instances. The opposite case occurred on only one instance. Both algorithms obtain the same solutions on 6 small size instances, i.e., BM1–BM6. The MS-SCHNN is more time-consuming for larger size instances, i.e., case 14–20. The DEH1 fails to obtain any feasible solution on 4 large size instances, i.e., case 14–17. On the remaining 16 instances, the MOEA/D-SHA obtains solutions dominating those obtained by the DEH1 on 10 instances. Both algorithms obtain the same solutions on 6 small size instances, i.e., BM1–BM5, and BM8. The DEH3/L even fails to obtain any feasible solution on 6 large size instances, i.e., case 14–19. On the remaining 14 instances, the MOEA/D-SHA obtains solutions dominating those obtained by the DEH3/L on 8

Table 9 Nondominated solutions with the minimum largest interference obtained by MOEA/D-SHA and the best solutions obtained by single-objective optimization algorithms (MS-SCHNN, DEH1 and DEH3/L)

Instance	MOEA/D-SHA			MS-SCHNN			DEH1		DEH3/L	
	Largest	Total	Time (s)	Largest	Total	Time (s)	Largest	Total	Largest	Total
1	30	100	0.16	30	100	0.06	30	100	30	100
2	4	13	0.15	4	13	0.03	4	13	4	13
3	7	85	1.36	7	85	0.63	7	85	7	85
4	64	880	1.38	64	880	0.60	64	880	64	880
5	640	8,693	1.40	640	8,693	0.57	640	8,693	640	8,693
6	32	981	6.05	32	981	4.53	32	1,032	32	1,017
7	61	2,799	22.47	61	2,892	19.90	61	2,824	61	3,178
8	620	12,653	3.77	620	15,937	2.50	620	12,653	620	12,653
9	6	565	102.30	7	589	124.40	6	591	7	589
10	60	4,852	104.60	67	5,280	127.57	64	5,444	77	6,179
11	640	50,425	102.90	678	60,666	126.43	640	50,894	676	61,813
12	41	2,795	171.20	41	3,082	210.23	43	3,260	59	4,259
13	50	6,891	242.90	61	7,047	475.17	85	7,411	89	8,959
14	60	10,791	330.60	80	12,201	774.70	–	–	–	–
15	68	12,982	368.90	83	15,106	4,407.2	–	–	–	–
16	69	15,979	418.40	85	17,748	4,369.3	–	–	–	–
17	73	19,185	512.90	92	23,531	1,696.43	–	–	–	–
18	62	13,100	490.10	77	15,070	1,537.07	81	14,049	–	–
19	36	5,148	620.60	44	5,887	1,508.43	87	6,298	–	–
20	17	1,523	639.20	16	1,434	1,000.33	21	1,622	31	2,544

Table 10 Nondominated solutions with the minimum total interference obtained by MOEA/D-SHA and the best solutions obtained by single-objective optimization algorithms (DEH2 and DEH3/T)

Instance	MOEA/D-SHA		DEH2		DEH3/T	
	Total	Largest	Total	Largest	Total	Largest
1	30	100	100	30	100	30
2	13	4	13	4	13	4
3	85	7	86	7	85	7
4	821	99	821	99	821	99
5	6,910	817	6,910	817	6,910	817
6	923	46	923	46	923	46
7	2,447	74	2,409	80	2,560	93
8	11,182	938	11,182	938	11,182	938
9	517	9	507	9	551	9
10	4,175	84	4,362	90	5,008	90
11	45,530	978	4,4903	978	4,9145	963
12	2,598	66	2,592	66	3,382	95
13	5,759	85	6,628	94	7,614	99
14	9,340	88	–	–	–	–
15	11,068	88	–	–	–	–
16	12,777	97	–	–	–	–
17	16,550	98	–	–	–	–
18	10,914	95	11,863	98	13,615	99
19	4,421	54	–	–	–	–
20	1,395	21	1,324	20	2,034	38

instances. Both algorithms obtain the same solutions on 6 small size instances, i.e., BM1–BM5, and BM8.

Secondly, we focus on comparing the MOEA/D-SHA with the single-objective algorithms concentrating on achieving minimum total interference. Hence, the DE with H2 (called DEH2), and DE with H3 (called DEH3/T) obtaining solution with low total interference, are selected from (Salman et al. 2010). Table 10 shows the nondominated solutions with the minimum total interference obtained by MOEA/D-SHA and the best solutions obtained by DEH2 and DEH3/T, respectively. The computational time of the algorithms is almost the same as that in Table 9. The DEH2 fails to obtain any feasible solution on 5 large size instances, i.e., Case 14–17 and Case 19. On the remaining 15 instances, the MOEA/D-SHA obtains solutions dominating those obtained by the DEH2 on 4 instances. The opposite case occurred on 4 instances. Both algorithms obtain the same solutions and nondominated solution on 6 and 1 (i.e., BM7) small size instances, respectively. The DEH3/T also fails to obtain any feasible solution on 5 large size instances, i.e., Case 14–17 and Case 19. On the remaining 15 instances, the MOEA/D-SHA obtains solutions dominating those obtained by the DEH3/T on 7 instances. Both algorithms obtain the same solutions on 7 small size instances, and obtain nondominated solutions on one small size instances.

In the DEH1, DEH2, and DEH3, due to order of carriers and the heuristic method used when assigning segments, some carriers may not find free consecutive segments equal to their length. This means that population individuals of the DEH1, DEH2, and DEH3 always stratify Constraint 2 and Constraint 3 but not necessarily satisfy Constraint 1, especially for larger size instances. Hence, they fail to obtain any feasible solution on large size instances. In contrast, the SHA with modification of priority value in the MOEA/D-SHA can alleviate this problem as shown in Fig. 8. On few large size instances, the MOEA/D-SHA is outperformed slightly by the single-optimization algorithms. The reason is that MOEA/D-SHA uniformly allocates computational resources to all subproblems to search for the whole Pareto optimal front, while single-objective optimization algorithms solely focuses on one problem whose size is equivalent to that of a subproblem in the MOEA/D-SHA (Mei et al. 2011). Thus, for some large size instances, the computational resources assigned to each subproblem might not be sufficient for the MOEA/D-SHA. Given limited computational resources, this problem may be mitigated by adopting the dynamical resource allocation strategy for subproblem because the different subproblems have different difficulty and thus require different amounts of computational resources (Li and Zhang 2009).

4.6 Discussion

The most important issue is whether the FAP really does need to be treated as a multiobjective problem. To answer this question, we should check whether the multiobjective evolutionary algorithm finds more than one nondominated solutions in the Pareto approximation sets (Garcia-Najera and Bullinaria 2011; Tan et al. 2006; Castro-Gutierrez et al. 2011). Results shown in Figs. 7 and 9 clearly confirm the multiobjective nature of the FAP. That is, more than one nondominated solutions shown in Figs. 7 and 9 demonstrate the conflict between the two objectives. Since the FAP is intrinsically a multiobjective problem in nature, the MOEA/D-SHA recognizes these alternative or trade-off solutions. Further, the MOEA/D-SHA obtains better extreme results than the single-optimization algorithms in many problem instances. Hence, there are indeed some benefits of the multiobjective problem formulation of the FAP.

In summary, the MOEA/D-SHA not only manages to achieve better or competitive extreme solutions than single-objective algorithms, but also finds simultaneously a set of alternative solutions. These solutions show different trade-offs between two objectives, which can be used to yield insight into the choices available to the decision maker and hence facilitate final decision making.

The MOEA/D-SHA focuses on generating Pareto optimal solutions. From a decision maker's point of view, the goal of multiobjective optimization is to find the single solution giving the best compromise between multiple objectives. This can be done using one of the multi-criteria decision-making approaches. Recently, several studies (Bechikh et al. 2011) have addressed the decision-making task to assist the decision maker in choosing the final alternative. In fact, the best compromise solution among Pareto optimal solutions can be easily found in the case of only two objectives because the decision maker can utilize a graphical representation of the Pareto optimal solutions like Figs. 7 and 9 or a simple multi-criteria decision-making approach. That is, some problem-specific expert's knowledge or decision maker's preference can be incorporated then for selecting the best solution from the Pareto optimal set produced by a multiobjective optimizer. In the absence of explicit decision maker's preferences, research in (Bechikh et al. 2011) supposes that knee regions represent the decision maker's preferences themselves. Knee regions are potential parts of the Pareto front presenting the maximal trade-offs between objectives. For solutions residing in knee regions, a small improvement in either objective will cause a large deterioration in at least another one which makes moving in either direction not attractive. Thus, solution in the knee regions can be chosen as the final alternative in multi-criteria decision making.

5 Conclusions

This study explicitly formulates FAP in satellite communication as a bi-objective optimization problem and thus presents a hybrid multiobjective evolutionary algorithm, named MOEA/D-SHA, for the MO-FAP. The MOEA/D-SHA combines the advanced features from both the DE for single-objective FAP and MOEA/D. Simulation results show that the MOEA/D-SHA outperforms significantly general-purpose MOEA/D and an off-the-shelf multiobjective evolutionary algorithm, i.e., NSGA-II. The advantages of MOEA/D-SHA over the state-of-the-art single-objective approaches are also shown through simulation results.

The most significant contribution of this study is our interpretation of the FAP as a multiobjective problem, which represents a philosophically different view of the problem as the whole. Our simple translation of the FAP into a multiobjective problem is surprisingly effective since the MOEA/D-SHA obtains promising results for the MO-FAP. Along this research line, other advanced multiobjective evolutionary algorithms (Zhou et al. 2011; Talbi et al. 2012) can also be adopted to solve this problem. Especially, modified MOEA/D methods (Li and Landa-Silva 2011; Ke et al. 2013; Shim et al. 2012) with the proposed SHA can be used to obtain Pareto optimal solutions for the MO-FAP in future.

As mentioned in Sect. 1, there are different FAP models and many different types of instances within the models (Aardal et al. 2007). Different FAP models have different features of the problem, such as the handling of interference among radio signals, the availability of frequencies, optimization criterion or objective, and constraints (Aardal et al. 2007). The general idea of the MOEA/D-SHA for the FAP in satellite communication can be extended to solve other FAPs, for example, the FAP in digital cellular phone standard GSM (Maximiano et al. 2013; Luna et al. 2011; Maximiano et al. 2009a,b). But the problem-specific components should be carefully redesigned. The main problem-specific components include solution representation, reproduction operators and SHA operators.

At the aspect of fundamental research perspective, the interplay between problem-specific subproblem-dependent heuristic for local search and MOEA/D framework for global search should be further studied. As stated in (Ishibuchi et al. 2003), it is important to maintain good balance between local search (exploitation) and global search (exploration) in multiobjective optimization algorithms. On the other hand, though the MOEA/D provides a very natural framework for using single-objective search techniques, how to utilize more effectively single-objective local search for optimizing each objective still should be further studied (Ishibuchi et al. 2008; Peng and Zhang 2012).

Acknowledgments The authors would like to thank Salman et al. (2010) for sending the original source code of the DE for FAP. This work was supported in part by the National Natural Science Foundation of China (60805026, 61070076, 61305085), and the Zhujiang New Star of Science and Technology in Guangzhou City (2011J2200093).

References

- Aardal KI, van Hoesel SPM, Koster AMCA, Mannino C, Sassano A (2007) Models and solution techniques for frequency assignment problems. *Ann Oper Res* 153(1):79–129
- Alcalá-Fdez J, Sánchez L, García S (2008) KEEL: a software tool to assess evolutionary algorithms to data mining problems. *Soft Comput* 13(3):307–318
- Bechikh S, Said LB, Ghédira K (2011) Searching for knee regions of the Pareto front using mobile reference points. *Soft Comput* 15(9):1807–1823
- Castro-Gutierrez J, Landa-Silva D, Pérez JM (2011) Nature of real world multi-objective vehicle routing with evolutionary algorithms. In: *Proceedings of IEEE International Conference on SMC*, pp 257–264
- Czyzszak P, Jaszkiwicz A (1998) Pareto simulated annealing: a metaheuristic technique for multiple-objective combinatorial optimization. *J Multi-Criteria Decis Anal* 7(1):34–47
- Das S, Suganthan PN (2011) Differential evolution: a survey of the state-of-the-art. *IEEE Trans Evolut Comput* 15(1):4–31
- Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: solving problems with box constraints. In: *Proceedings of IEEE Transactions on Evolutionary Computation*, 2014 (in press)
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans Evolut Comput* 6(2):182–197
- Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evolut Comput* 1(1):3–18
- Funabiki N, Nishikawa S (1997) A gradual neural-network approach for frequency assignment in satellite communication systems. *IEEE Trans Neural Netw* 8(6):1359–1370
- García-Najera A, Bullinaria J (2011) An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows. *Comput Oper Res* 38(1):287–300
- Hopfield JJ, Tank DW (1985) Neural computation of decisions in optimization problems. *Biol Cybern* 52:141–152
- Ishibuchi H, Hitotsuyanagi Y, Tsukamoto N, Nojima Y (2008) Use of heuristic local search for single-objective optimization in multi-objective memetic algorithms. In: *Proceedings of parallel problem solving from nature-PPSN X*, pp 743–752
- Ishibuchi H, Yoshida T, Murata T (2003) Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Trans Evolut Comput* 7(2):204–223
- Jaszkiwicz A (2003) Do multiple objective metaheuristics deliver on their promises? A computational experiment on the set covering problem. *IEEE Trans Evolut Comput* 7(2):133–143
- Ke L, Zhang Q, Battiti R (2013) MOEA/D-ACO: a multiobjective evolutionary algorithm using decomposition and ant colony. *IEEE Trans Cybern* 43(6):1845–1859
- Konstantinidis A, Yang K (2011a) Multi-objective energy-efficient dense deployment in wireless sensor networks using a hybrid problem-specific MOEA/D. *Appl Soft Comput* 11(6):4117–4134
- Konstantinidis A, Yang K (2011b) Multi-objective k-connected deployment and power assignment in WSNs using a problem-specific constrained evolutionary algorithm based on decomposition. *Comput Commun* 34(1):83–98
- Kurokawa T, Kozuka S (1993) A proposal of neural networks for the optimum frequency assignment problem. *Trans IEICE J76-B-II(10):811–819* (in Japanese)
- Lacomme P, Prins C, Sevaux M (2006) A genetic algorithm for a biobjective capacitated arc routing problem. *Comput Oper Res* 33(12):3473–3493
- Li H, Landa-silva D, Gandibleux X (2010) Evolutionary multi-objective optimization algorithms with probabilistic representation based on pheromone trails. In: *Proceedings of the IEEE congress on evolutionary computation*, pp 2307–2314
- Li H, Zhang Q (2009) Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Trans Evolut Comput* 12(2):284–302
- Li H, Landa-Silva D (2011) An adaptive evolutionary multi-objective approach based on simulated annealing. *Evolut Comput* 19(4):561–595
- Liu W, Shi H, Wang L (2007) Minimizing interference in satellite communications using chaotic neural networks. In: *Proceeding of the third international conference on natural computation*, pp 441–444
- Luna F, Estébanez C et al (2011) Optimization algorithms for large-scale real-world instances of the frequency assignment problem. *Soft Comput* 15(5):975–990
- Maximiano MS, Vega-Rodríguez MA et al (2009) Multiobjective frequency assignment problem using the MO-VNS and MO-SVNS algorithms. In: *Proceedings of the world congress on nature and biologically inspired computing*, pp 221–226
- Maximiano MS, Vega-Rodríguez MA et al (2009) Parameter analysis for differential evolution with Pareto tournaments in a multiobjective frequency assignment problem. In: *Proceedings of the 10th international conference on intelligent data engineering and automated learning*, pp 799–806
- Maximiano MS, Vega-Rodríguez MA et al (2013) A new multiobjective artificial bee colony algorithm to solve a real-world frequency assignment problem. *Neural Comput Appl* 22(7–8):1447–1459
- Mei Y, Tang K, Yao X (2011) Decomposition-based memetic algorithm for multi-objective capacitated arc routing problem. *IEEE Trans Evolut Comput* 15(2):151–165
- Mizuike T, Ito Y (1989) Optimization of frequency assignment. *IEEE Trans Commun* 37(10):1031–1041
- Mizuike T, Ito Y (1986) Optimum frequency assignment for reduction of cochannel interference. *Trans IEICE J69-B(9):921–932* (in Japanese)
- Pelton JN, Oslund RJ, Marshall P (2004) *Communications satellites: global change agents*. Lawrence Erlbaum, Mahwah
- Peng W, Zhang Q, Li H (2009) Comparison between MOEA/D and NSGA-II on the multi-objective travelling salesman problem. In: *Proceedings of studies in computational intelligence, multi-objective memetic algorithms*, vol 171, pp 309–324
- Peng W, Zhang Q (2012) Network topology planning using MOEA/D with objective-guided operators. In: *Proceedings of parallel problem solving from nature-PPSN II*, pp 62–71
- Praditwong K, Harman M, Yao X (2011) Software module clustering as a multi-objective search problem. *IEEE Trans Softw Eng* 37(2):264–282
- Qu B-Y, Li H, Zhao S-Z, Suganthan PN, Zhang Q (2011) Multiobjective evolutionary algorithms: a survey of the state-of-the-art. *Swarm Evolut Comput* 1(1):32–49
- Saha S, Sur-Kolay S, Dasgupta P, Bandyopadhyay S (2009) MAKE: multiobjective algorithm for k-way equipartitioning of a point set. *Appl Soft Comput* 9(2):711–724
- Salcedo-Sanz S, Santiago-Mozos R, B-Calz'on C (2004) A hybrid Hopfield network-simulated annealing approach for frequency assign-

- ment in satellite communications systems. *IEEE Trans Syst Man Cybern-B Cybern* 34(2):1108–1116
- Salcedo-Sanz S, B-Calz'ón C (2005) A hybrid neural-genetic algorithm for the frequency assignment problem in satellite communications. *Appl Intell* 22(3):207–217
- Salman AA, Ahmad I, Omran MGH, Mohammad MG (2010) Frequency assignment problem in satellite communications using differential evolution. *Comput Oper Res* 37(12):2152–2163
- Shim VA, Tan KC, Cheong CY (2012) A hybrid estimation of distribution algorithm with decomposition for solving the multiobjective multiple traveling salesman problem. *IEEE Trans Syst Man Cybern Part C Rev Appl* 42(5):682–691
- Storn RM, Price KV (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11(4):341–359
- Talbi E-G, Basseur M, Nebro AJ, Alba E (2012) Multi-objective optimization using metaheuristics: non-standard algorithms. *Int Trans Oper Res* 19(1—2):283–305
- Tan KC, Chew YH, Lee LH (2006) A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows. *Comput Optim Appl* 34(1):115–151
- Wang J, Cai Y, Yin J (2011) Multi-start stochastic competitive Hopfield neural network for frequency assignment problem in satellite communications. *Expert Syst Appl* 38(1):131–145
- Wang L, Liu W, Shi H (2008) Noisy chaotic neural networks with variable thresholds for the frequency assignment problem in satellite communications. *IEEE Trans Syst Man Cybern Part C Rev Appl* 38(2):209–217
- Wang Z, Tang K, Yao X (2010) Multi-objective approaches to optimal testing resource allocation in modular software systems. *IEEE Trans Reliab* 59(3):563–575
- Whalen DJ (2008) Communication satellites: making the global village possible. Available at: <http://www.hq.nasa.gov/office/pao/History/satcomhistory.html>
- Zhang Q, Liu W, Li H (2009) The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In: Proceedings of IEEE congress on evolutionary computation, pp 1–6
- Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evolut Comput* 11(6):712–731
- Zitzler E, Laumanns M, Thiele L (2002) SPEA2: improving the strength Pareto evolutionary algorithm. In: Proceedings of evolution methods for design, optimization and control applications to industrial problems, pp 95–100
- Zitzler E, Thiele L (1999) Multi-objective evolutionary algorithms: a comparative study and the strength Pareto approach. *IEEE Trans Evolut Comput* 3(4):257–271