Neural Comput & Applic (2012) 21:1917–1929
DOI 10.1007/s00521-011-0588-4

ICIC2010

# Hypervolume-based multi-objective local search

**Matthieu Basseur · Rong-Qiang Zeng · Jin-Kao Hao**

Received: 30 August 2010 / Accepted: 30 March 2011 / Published online: 19 April 2011


**Abstract** This paper presents a multi-objective local search, where the selection is realized according to the hypervolume contribution of solutions. The HBMOLS algorithm proposed is inspired from the IBEA algorithm, an indicator-based multi-objective evolutionary algorithm proposed by Zitzler and Künzli in 2004, where the optimization goal is defined in terms of a binary indicator defining the selection operator. In this paper, we use the indicator optimization principle, and we apply it to an iterated local search algorithm, using hypervolume contribution indicator as selection mechanism. The methodology proposed here has been defined in order to be easily adaptable and to be as parameter-independent as possible. We carry out a range of experiments on the multi-objective flow shop problem and the multi-objective quadratic assignment problem, using the hypervolume contribution selection as well as two different binary indicators which were initially proposed in the IBEA algorithm. Experimental results indicate that the HBMOLS algorithm is highly effective in comparison with the algorithms based on binary indicators.

**Keywords** Hypervolume contribution · Multi-objective · Local search · Flow shop problem · Quadratic assignment problem

M. Basseur · R.-Q. Zeng (✉) · J.-K. Hao
Department of Computer Science, University of Angers, 2, Boulevard Lavoisier, 49045 Angers Cedex 01, France
e-mail: zeng@info.univ-angers.fr

M. Basseur
e-mail: matthieu.basseur@info.univ-angers.fr

J.-K. Hao
e-mail: Jin-Kao.Hao@info.univ-angers.fr

## 1 Introduction

Problems with multiple objectives arise in a natural fashion in many areas, such as computer science, engineering, economics, physics, chemistry, and ecology. Since most of these problems are known to be NP-complete [10], heuristic approaches are often proposed in order to find good compromise solutions. These methods are usually derived from metaheuristics proposed in single-objective optimization, such as Genetic Algorithms, Evolution Strategies, Simulated Annealing, and Tabu Search. In order to adapt a metaheuristic to multi-objective optimization, a major step to consider is how we can assign a fitness value to a solution. In single-objective optimization, a total order of the relation can be easily used to rank the solutions. In Multi-Objective Optimization (MOO), such a natural total order relation does not exist. The dominance relation allows us to define a partial order to rank the solutions in some special cases.

In multi-objective optimization, we are interested in finding the set of Pareto solutions, which keeps the best compromise solutions among all the objectives. Since in most cases, it is not possible to compute the Pareto optimal set in a reasonable time, we are interested in computing a set of non-dominated solutions which is as close as possible to the Pareto optimal set, a non-dominated solution being an explored solution which has never been dominated by another explored solution.

Most of the solution methods for MOO can be classified into two categories: (1) scalar approaches, i.e. the fitness value of a solution is defined as a weighted sum of each objective function, (2) Pareto approaches, i.e., the fitness value of a solution is defined according to its dominance relation with the other solutions of the population. In [26], the authors extend the idea of flexible integration of preference information by Fonseca and Fleming, and propose a general Indicator-Based Evolutionary Algorithm (IBEA) that can be combined with arbitrary indicators. In their initial study, they show that IBEA can significantly improve the quality of the generated Pareto set approximation with respect to the considered optimization goal. Since this study, we can distinguish a third class of MOO solution approaches: The *indicator-based optimization*.

The actual studies about indicator-based optimization algorithms are mainly evolutionary algorithms. In this paper, we propose a population based multi-objective local search using the indicator-based optimization principle. The main principle of indicator-based search consists in using a comparison metric to assign the fitness values to the solutions. The metrics are used to compare the outputs at the end of the optimization process, possibly including the decision-maker preferences. A popular metric used to compare these fronts is the hypervolume measure (also known as S-metric or the Lebesgue measure).

Based on the indicator proposed in [3], we define a hypervolume contribution indicator to assign to each solution a fitness value that can be used in the selection process of multi-objective evolution algorithms. We aim to show the interest of the use of a hypervolume indicator for the fitness assignment in a local search context. In order to evaluate the quality of the hypervolume-based approach, we compare hypervolume contribution indicator with those binary indicators. Our aim is to show the effectiveness of the hypervolume search concept, and to demonstrate by experiments that our proposed algorithm obtains good results and outperforms Indicator-Based Multi-Objective Local Search (IBMOLS) using binary indicators proposed in [26].

The paper is organized as follows. In Sect. 2, we briefly present the indicator-based optimization principle and the IBMOLS algorithm associated to this principle, which is presented in a previous study [3]. In Sect. 3, we present the hypervolume-based optimization principle, and compare hypervolume contribution indicator with binary indicators. In this section, the algorithm computing the *hypervolume contribution* is also presented, then IBMOLS algorithm is modified into the Hypervolume-Based Multi-Objective Local Search algorithm (HBMOLS). In Sect. 4, we present the experimental results, which are obtained by the application of HBMOLS to a multi-objective permutation flow shop problem and a multi-objective quadratic assignment problem. Then, the conclusions and perspectives are discussed in Sect. 5.

## 2 Multi-objective selection using hypervolume

Our research is strongly inspired from the work of Zitzler and Künzli [26]. In this section, we first give some basic notations and definitions related to multi-objective optimization. Then, we briefly discuss about two binary indicators which are strongly related to our study. Finally, we present the indicator-based multi-objective local search algorithm proposed in [3].

### 2.1 Multi-objective optimization

First, we recall some useful notations and definitions of multi-objective optimization problems (MOPs), which are taken from [26]. Let $X$ denote the search space of the optimization problem under consideration and $Z$ the corresponding objective space. Without loss of generality, we assume that $Z = \Re^n$ and that all $n$ objectives are to be minimized or maximized. Each $x \in X$ is assigned exactly one objective vector $z \in Z$ on the basis of a vector function $f : X \rightarrow Z$ with $z = f(x)$. The mapping $f$ defines the evaluation of a solution $x \in X$, and often one is interested in those solutions that are Pareto optimal with respect to $f$. The relation $x_1 \succ x_2$ means that the solution $x_1$ is *preferable* to $x_2$. Let $f_1, \ldots, f_n$ be the $n$ objective function to minimize. The dominance relation between two solutions $x_1$ and $x_2$ is usually defined as follows:

**Definition 1** A decision vector $x_1$ is said to dominate another decision vector $x_2$ (written as $x_1 \succ x_2$), if $f_i(x_1) \leq f_i(x_2)$ for all $i \in \{1, \ldots, n\}$ and $f_j(x_1) < f_j(x_2)$ for at least one $j \in \{1, \ldots, n\}$.

**Definition 2** $x \in X$ is said to be Pareto optimal if and only if a solution $x_i \in X$ which *dominates* $x$ does not exist.

As we know, identifying a good approximation of the Pareto optimal set usually depends on the decision maker and the optimization scenario. In [26], the authors first define the optimization goal in terms of a binary quality indicator and then to directly use this measure in the selection process. In fact, a binary quality indicator can be regarded as a continuous extension of the concept of Pareto dominance on sets of objective vectors [28].

### 2.2 Binary indicator

A binary indicator can be used to compare two single solutions, or a single solution against an entire population. With such a comparison, every solution will be given a fitness value. In their study, Zitzler and Künzli proposed the Indicator-Based Evolutionary Algorithm (IBEA), which used this principle to evolve a population of solutions during the search. During the selection process, the solution with the smallest fitness value, in terms of the quality indicator used, is deleted from the population. In IBEA, two indicators are tested: The epsilon $I_\epsilon$ and the hypervolume indicator $I_{\mathrm{Hyp}}$, defined as follow[1]:

[1] We assume throughout the paper that all the objective functions are normalized.

$$I_{\epsilon}(x_1,x_2) = \max_{i\in\{1,\ldots,n\}} (f_i(x_1) - f_i(x_2)) \quad (1)$$

$I_{\epsilon}(x_1,x_2)$ (where $x_1 \in X$ and $x_2 \in X$) represents the minimal translation (in the objective space) on which to execute $x_1$ so that it dominates $x_2$. Let us note that the translation could take negative values.

$$I_{\text{Hyp}}(x_1,x_2) = \begin{cases} H(x_2) - H(x_1) & \text{if } x_2 \succ x_1 \\ H(\{x_1,x_2\}) - H(x_1) & \text{otherwise} \end{cases} \quad (2)$$

$H(x)$ represents the volume of space dominated by $x$, $I_{\text{Hyp}}(x_1, x_2)$ represents the volume of the space that is dominated by $x_2$, but not by $x_1$.

In order to compute the fitness values of solutions, different methods are proposed in [26]. We will use an intuitive and simple method to evaluate a solution $x$ against the whole population: The indicator values of $x$ with respect to the rest of the population is computed, and $(P/\{\ x\}, x)$ corresponds to the minimal indicator value computed (Eq. 3). In our experiments, we will use this formulation for the $I_{\epsilon}$ and $I_{\text{Hyp}}$ indicators.

$$I(P/\{x\},x) = \min_{z\in P/\{x\}} I(z,x) \quad (3)$$

### 2.3 Indicator-based multi-objective local search

Generally, local search methods are known to be very efficient in many real-world applications, and especially on large scale problems. However, define local search for MOO is not an easy task. Yet, the use of such an algorithm is often necessary since evolutionary algorithm convergence is usually very slow. Since several years, some researchers have proposed local search for MOO.

In [15], a multi-objective local search is based on the dominance relation between the considered solution and an archive of compromise solutions and is incorporated into an evolution strategy method; this algorithm is known as the Pareto Archived Evolution Strategy. In [1, 19], multi-objective local searches are proposed to solve MOO flow shop problem.

In [3], the authors propose a simple and generic indicator-based multi-objective local search (IBMOLS), where selection is realized according to binary indicators. The outlines of the IBMOLS algorithm are described in algorithm 1. The HBMOLS algorithm proposed in this paper is greatly inspired from this study.

In order to run experiments in a fixed amount of time, IBMOLS was iterated until the running time is reached (see algorithm 2). According to the results obtained in [3], we choose to initialize populations by applying noise on randomly chosen $PO$ solutions.

In [3], experimental results show that IBMOLS algorithm was highly efficient in comparison with similar approaches using classical ranking methods. However, the best results were achieved by $I_{\epsilon}$ indicator which outperforms $I_{\text{Hyp}}$ indicator. The main reason is that $I_{\text{Hyp}}(x, P)$ does not compute the hypervolume contribution of a solution $x$ regarding to a population $P$, but something slightly different. In the next section, we will discuss this issue, then we propose the HBMOLS algorithm, which is similar to IBMOLS, but in HBMOLS, selection is realized according to the hypervolume contribution of solutions.

**Algorithm 1** Indicator-Based Multi-Objective Local Search (IBMOLS / HBMOLS)

**Input**: $N$ (population size)
 $I$ (binary indicator)
**Output**: $A$: (Pareto approximation set)
**Step 1 - initialization**: Generate an initial population $P$ of size $N$
**Step 2**: $A \leftarrow$ Non-dominated solutions of $P$
**Step 3 - fitness assignment**: Calculate fitness values of individual $x$ in $P$
**Step 4 - local search step**: For all $x \in P$ do:
update P upper and lower bounds
repeat
 1) $x^* \leftarrow$ one neighbors of $x$
 2) compute $x^*$ fitness: $I(P, x^*)$ / $HC(P, P\backslash\{x^*\})$
 3) update $z \in P$ fitness value, which is the neighbor of $x^*$
 4) $\omega \leftarrow$ worst individual in $P$
 5) remove $\omega$ from $P$
 6) update $z \in P$ fitness value, which is the neighbor of $\omega$
until all neighbors are explored or $\omega \neq x^*$
**Step 5 - termination**: $A \leftarrow$ Non-dominated solutions of $A \bigcup P$. If $A$ does not change, then
return $A$; else perform another local search step.

**Algorithm 2** Iterated IBMOLS / HBMOLS Algorithm

**Input**: $N$ (population size)
 $I$ (binary indicator)
**Output**: $PO$: (Pareto approximation set)
**Step 1**: $PO \leftarrow \Phi$
**Step 2**:
**while** Running time is not reached **do**
 $P \leftarrow$ Generate a new population (random noise on $PO$ solutions)
 $A \leftarrow$ IBMOLS (initialized with $P$)
 $PO \leftarrow$ Non-dominated solutions of $PO \bigcup A$
**end while**
**Step 3**: Return $PO$

## 3 Hypervolume-based multi-objective local search

This section presents the main contribution of the paper, which is the application of hypervolume contribution as the selection mechanism in HBMOLS. Hypervolume is the

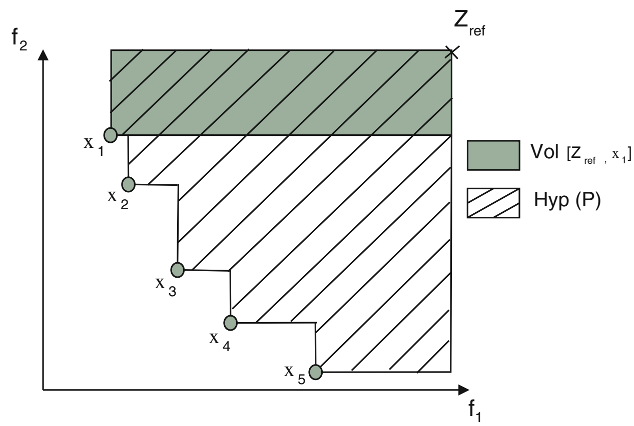


**Fig. 1** The hypervolume Hyp($P$) is the area which is enclosed by a population $P$ (Including the solutions $x_i, i \in \{1,\ldots,5\}$), according to a reference point $Z_{\text{ref}}$; The hypervolume of the solution $x_1$ is denoted as [$Z_{\text{ref}}$, $x_1$], which is colored in *gray*

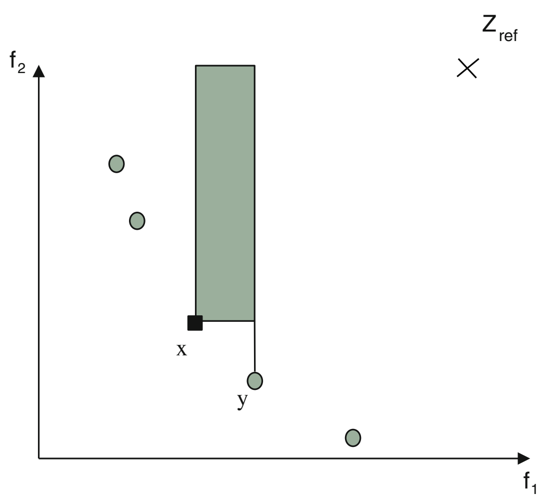


**Fig. 2** $I_{\text{Hyp}}(P \setminus \{X\}, P)$: minimal hypervolume difference $I_{\text{Hyp}}(x, y)$ between $x$ and $y, y \in P$

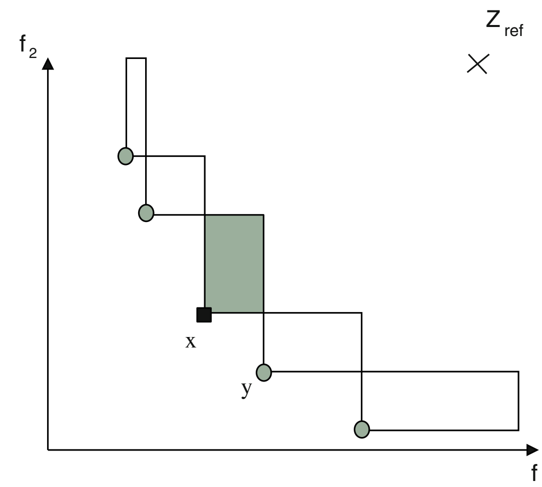


**Fig. 3** Hyp$C$($x$, $P$): hypervolume contribution of a solution $x$ to a population $P$

n-dimensional space that is dominated by the points (solutions) in a front. A front with a larger hypervolume is likely to present a set of better trade-off to a user than a front with a smaller hypervolume.

A few papers relating to hypervolume indicator optimization are already published in the literature. In [2], the authors propose HypE, a hypervolume estimation algorithm for multi-objective optimization. A fast incremental hypervolume algorithm IHSO is presented by Bradstreet, While and Barone [7], which is used as part of the selection in a multi-objective evolutionary algorithm. Besides, in [6], the authors devise $\mathcal{S}$ metric selection EMOA (SMS-EMOA), which is a multi-objective selection based on dominated hypervolume. The experimental results indicate that HypE, IHSO, and SMS-EMOA are highly effective for multi-objective problems in comparison with the state-of-the-art multi-objective evolutionary algorithms.

### 3.1 Hypervolume computation

Let us consider the following definition of the hypervolume Hyp enclosed by a population $P$ according to a reference point $Z_{\text{ref}}$ [9] (see Fig. 1):

$$\text{Hyp}(P) = \text{VOL}([Z_{\text{ref}}, x_1] \bigcup \cdots \bigcup [Z_{\text{ref}}, x_n]) \quad (4)$$

with VOL(.) being the usual Lebesgue measure. The contribution can be seen as the measure of the space that is dominated by $x$, but no other point in $P$. With the definition 4, let us now define the hypervolume contribution Hyp$C$ of a solution $x$ to a population $P$:

$HypC(x, P) = \text{Hyp}(P) - \text{Hyp}(P \setminus \{x\})$.

As introduced in the previous section, $I_{\text{Hyp}}$ used in IBEA and IBMOLS algorithms does not compute the hypervolume contribution, but something quite different, since solutions are compared by pairs. According to the definition of hypervolume contribution, we need to evaluate the considered solution with the whole population. The main difference between $I_{\text{Hyp}}$ and Hyp$C$ is illustrated in Figs. 2 and 3.

The main drawback of the above definition is the fact that if $P$ contains some dominated solutions, their corresponding hypervolume contribution will be 0 for all of them. In order to refine this measure, we will define HC measure, where we distinguish two cases:

$$\text{HC}(x, P) = \begin{cases} \text{Hyp}C(x, P) & \text{if } x \text{ is non-dominated} \\ -\max_{y \in P, y \succ x}(\text{VOL}([y, x])) & \text{otherwise} \end{cases}$$

HC measure includes a new case applied when $x$ is dominated by at least one solution of $P$. In this special case, the main difference between $I_{\text{Hyp}}$ and Hyp$C$ is illustrated in Figs. 4 and 5.

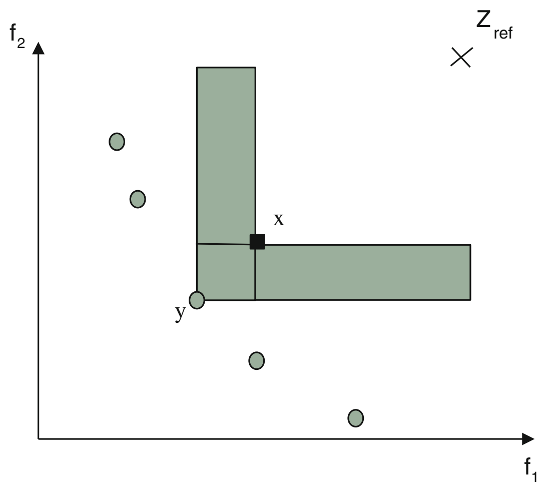


**Fig. 4** $I_{\mathrm{Hyp}}(x, P)$ of a dominated solution $x$ to a population $P$: the dominance area $I_{\mathrm{Hyp}}(x, y)$ (*gray box*) between $x$ and $y(y \succ x, y \in P)$

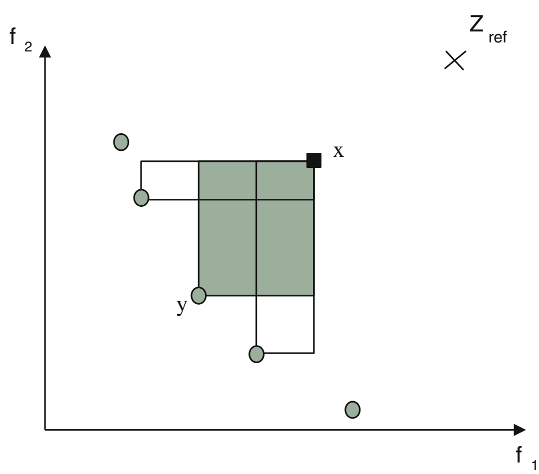


**Fig. 5** HC($x$, $P$) of a dominated solution $x$ to a population $P$: maximum dominance area (*gray box*) computed between $x$ and $y(y \succ x, y \in P)$

It is well known that the computation of hypervolume contribution, as well as the general hypervolume computation, is NP-hard, which cannot be solved exactly in time polynomial in the number of dimensions unless $P = \mathrm{NP}$ [8]. Then HC computation is also NP-hard.

### 3.2 Bi-objective hypervolume computation algorithm

In this section, we will focus on the hypervolume computation in the bi-objective case. The extension of HBMOLS to multi-objective case will be discussed in the next section.

Here, we consider the local search as described in algorithm 1, where selection is based on HC values. The HC selection computation in the bi-objective case consists in (1) evaluating a fitness value (HC) of a new solution $x$,

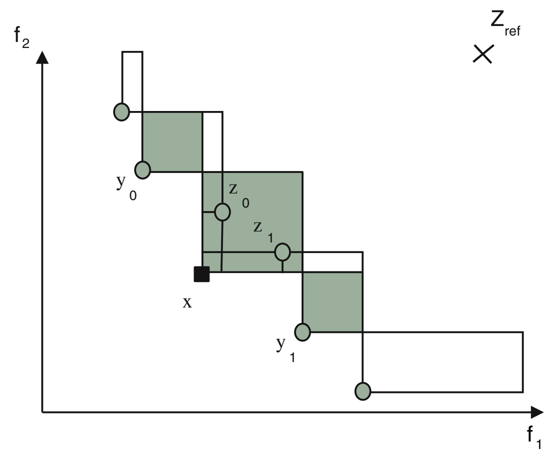


**Fig. 6** HC fitness update: new non-dominated solution found (*gray boxes*: new dominance areas to compute)

and updates the fitness values of the whole population $P$ according to $x$, (2) determining and deleting the worst solution $w$, (3) updating the fitness value of each solution in $P$ according to $w$. These steps are detailed bellow[2]:

(1) If $x$ is dominated then

$$\mathrm{HC}(x, P) \leftarrow -\max_{y \in P, y \succ x} (\mathrm{VOL}([y, x]))$$

Other fitness values in $P$ do not need to be updated. If $x$ is non-dominated, then first update the set of dominated solutions (solutions $z_0$ and $z_1$ in Fig. 6), and compute their new negative fitness values according to $x$. Then compute the fitness value for $x$, using its non-dominated neighbors (solutions $y_0$ and $y_1$ in Fig. 6):

$$\mathrm{HC}(x, P) \leftarrow (f_1(y_1) - f_1(x)) \times (f_2(y_0) - f_2(x))$$

Finally, compute new fitness values for the neighbors of $x$ ($y_0$ and $y_1$). As illustrated in Fig. 6:

$$\mathrm{HC}(y_0, P) \leftarrow \mathrm{HC}(y_0, P \backslash \{x\}) \times \frac{f_1(x) - f_1(y_0)}{f_1(z_0) - f_1(y_0)}$$

$z_0$ being the non-dominated or newly dominated solution with the smallest $f_1$ value greater than $f_1(x)$ (this solution can be $y_1$). A similar equation can be defined to compute $HC(y_1, P)$.

(2) $w$ is selected from $P$ with respect to: $\mathrm{HC}(w) = \min_{y \in P}(\mathrm{HC}(y, P))$. Delete $w$ from the population.

(3) If $w$ is dominated, the fitness values of the remaining solutions do not need to be updated. If $w$ is non-dominated, then the fitness values of the neighbors of $w$ need to be updated (see Fig. 7):

[2] If a neighbor does not exist, its objective value is replaced by the objective value of the reference point $Z_{\mathrm{ref}}$.

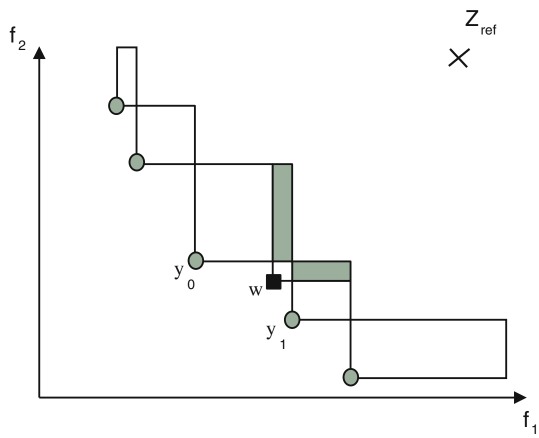


**Fig. 7** HC fitness update: deletion of a non-dominated solution: the fitness values of solution $y_0$ and $y_1$ are computed by adding corresponding grey area, respectively

$$\mathrm{HC}(y_0, P) \leftarrow \mathrm{HC}(y_0, P \backslash \{x\}) \times \frac{f_1(y_1) - f_1(y_0)}{f_1(y_2) - f_1(y_0)}$$
$$\mathrm{HC}(y_1, P) \leftarrow \mathrm{HC}(y_1, P \backslash \{x\}) \times \frac{f_2(y_0) - f_2(y_1)}{f_2(y_2) - f_2(y_1)}$$

One interest of the HBMOLS algorithm is that its complexity is linear with respect to the population size. However, let us note that the updating process in HC computation is described only for the bi-objective case, and it can not easily be extended to the multi-objective case in its present form.

### 3.3 Three-objective hypervolume computation algorithm

In the three-objective case, it is more complex to compute the hypervolume contribution directly for each solution in the non-dominated set. As is shown in Fig. 8, areas 2 and 4 are the only two different non-dominated areas of solution $x$ in the bi-objective case.

However, in Fig. 9, we can see that there are six different non-dominated areas of solution $x$ in the three-objective case. Therefore, HC computation in the three-objective algorithm is a little different from that in the bi-objective algorithm.

The HC selection computation in the three-objective case consists in (1) evaluating the fitness value (HC) of a new solution $x$ which is dominated, (2) if the new solution $x$ is non-dominated, computing the hypervolume contribution for the solution $x$ and each solution in the non-dominated set, (3) determining and deleting the worst solution $w$. These steps are described as follows:

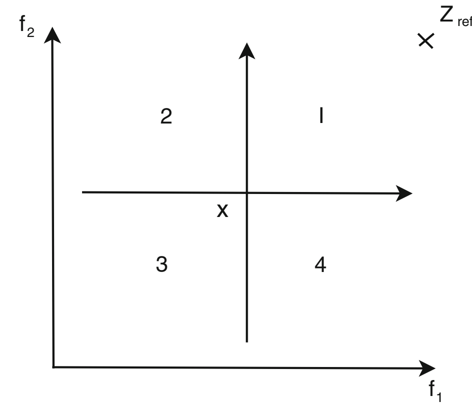


**Fig. 8** The non-dominated areas in bi-objective case (areas 2 and 4)

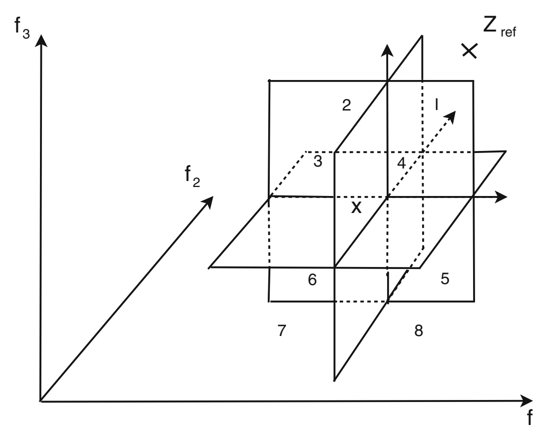


**Fig. 9** The non-dominated areas in three-objective case (areas 2, 3, 4, 5, 6, and 8)

(1) If $x$ is dominated then

$$\begin{aligned}\mathrm{HC}(x, P) &\leftarrow -\max_{y \in P, y \succ x}(\mathrm{VOL}([y, x])) \\ &= -\max_{y \in P, y \succ x} \mid (f_1(y) - f_1(x)) \\ &\times (f_2(y) - f_2(x)) \times (f_3(y) - f_3(x)) \mid\end{aligned}$$

Other fitness values in $P$ only need to be kept.

(2) If $x$ is non-dominated, then first update the set of the dominated solutions, and compute their new negative fitness values according to $x$. Then compute the fitness values for $x$ and each solution in the non-dominated set, using their non-dominated neighbors. For example, all the neighbors of solution $x$ are illustrated in Fig. 10, which are denoted as $y_1$, $y_2$, $y_3$, $y_4$, $y_5$, $y_6$, and $y_7$. The point $x_0$ in Fig. 10 is the projection of solution $x$ on the plane $C$. The relation between solution $x$ and its neighbors is summarized in Table 1.

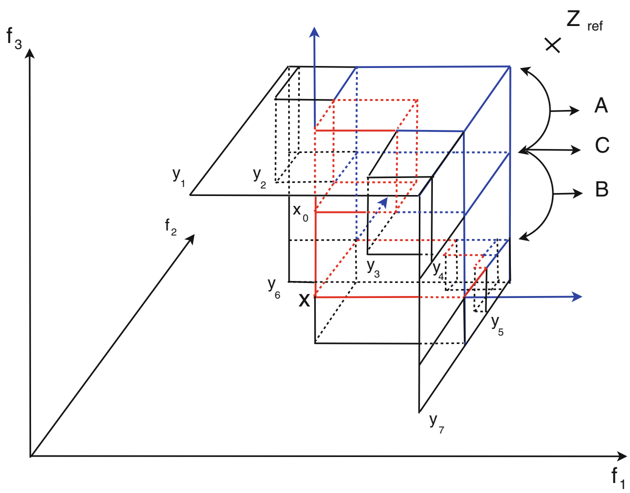


**Fig. 10** HC fitness computation in the three-objective case, the two parts *A* and *B* are described in Figs. 9 and 10, respectively

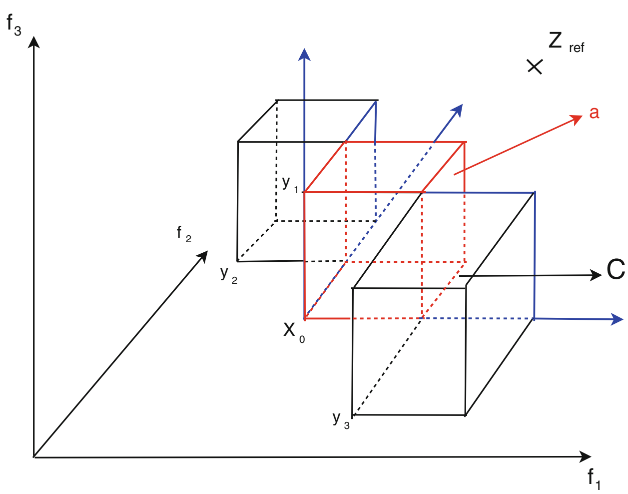


**Fig. 11** HC fitness computation in the three-objective case: part *A*, the hypervolume contribution of solution *x* in this part is computed by equation *a*

The hypervolume contribution of solution $x$ is computed as follows:

$$\begin{aligned}\mathrm{HC}(x,P) \leftarrow\ & (f_1(y_3)-f_1(x))\times(f_2(y_2)-f_2(x))\\ & \times(f_3(y_1)-f_3(y_2))\cdots\cdots\cdots(a)\\ & +(f_1(y_4)-f_1(x))\times(f_2(y_6)-f_2(x))\\ & \times(f_3(y_2)-f_3(x))\cdots\cdots\cdots\cdots\cdots(b)\\ & +(f_1(y_5)-f_1(y_4))\times(f_2(y_4)-f_2(x))\\ & \times(f_3(y_2)-f_3(x))\cdots\cdots\cdots\cdots\cdots(c)\\ & +(f_1(y_7)-f_1(y_5))\times(f_2(y_5)-f_2(x))\\ & \times(f_3(y_2)-f_3(x))\cdots\cdots\cdots\cdots\cdots(d)\end{aligned}$$

(3) $w$ is selected from $P$ with respect to: $\mathrm{HC}(w) = \min_{y\in P}(\mathrm{HC}(y,P))$. Delete $w$ from the population. Other fitness values in $P$ do not need to be updated (Figs. 11 and 12).

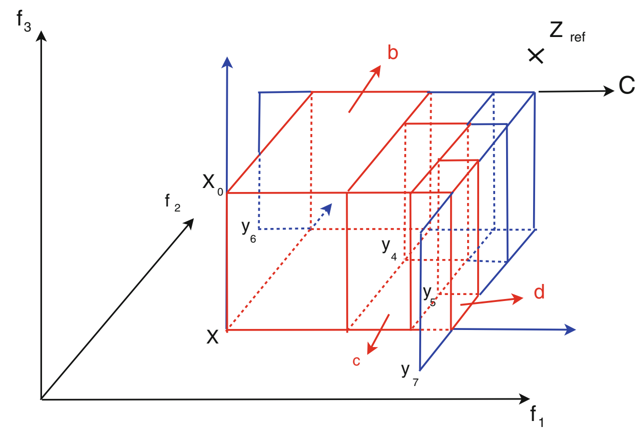


**Fig. 12** HC fitness computation in the three-objective case: part *B*, the hypervolume contribution of solution *x* in this part has three components, which are computed by equation *b*, equation *c*, and equation *d*, respectively

**Table 1** The *neighbors* of solution $x$ in Fig. 10 (solutions $y_4$ and $y_5$ are non-dominated)

| Area and Objective | The non-dominated neighbors of solution $x$ | | | | | | |
|---|---|---|---|---|---|---|---|
| | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ | $y_6$ | $y_7$ |
| Area | 3 | 2 | 4 | 5 | 5 | 6 | 8 |
| $f_1(x)$ | $>f_1(y_1)$ | $>f_1(y_2)$ | $<f_1(y_3)$ | $<f_1(y_4)$ | $<f_1(y_5)$ | $>f_1(y_6)$ | $<f_1(y_7)$ |
| $f_2(x)$ | $>f_2(y_1)$ | $<f_2(y_2)$ | $>f_2(y_3)$ | $<f_2(y_4)$ | $<f_2(y_5)$ | $<f_2(y_6)$ | $>f_2(y_7)$ |
| $f_3(x)$ | $<f_3(y_1)$ | $<f_3(y_2)$ | $<f_3(y_3)$ | $>f_3(y_4)$ | $>f_3(y_5)$ | $>f_3(y_6)$ | $>f_3(y_7)$ |

Computing the hypervolume contribution of each solution in the non-dominated set plays an important role in the HBMOLS algorithm. Nevertheless, it is not easy to extend *HC* computation to more than three objectives.

We take the case of four objectives as example. Let $f_1, f_2, f_3$, and $f_4$ be the four objectives to minimize, the hypervolume contribution of each solution in the population can be computed as follows: (1) If the solution $x$ is dominated, a negative value is assigned to this solution. The value corresponds to the greatest area between a non-dominated solution and the solution $x$; (2) If the solution $x$ is non-dominated, firstly we compute the hypervolume contribution $HC_3$ in the space composed by the three objectives $f_2$, $f_3$, and $f_4$. Then, we multiply the corresponding value of the solution $x$ in the objective $f_1$ with $HC_3$, that is, the hypervolume contribution of the solution $x$ in the four objectives.

Generally, we can iteratively compute the hypervolume contribution for each solution in $N$ ($N \geq 4$) objectives. However, the number of the non-dominated areas, where the neighbor solutions could be located in, equals to $2^N - 2$ ($N$ is the number of objectives). Bringmann and Friedrich [8] have proven that the problem of computing the hypervolume is *NP*-hard. Therefore, with the number of objectives increasing, it will usually take too much time to compute *HC* and the effectiveness of the HBMOLS algorithm will be affected.

## 4 Experiments

In this section, experiments with the HBMOLS and IBMOLS algorithms are reported. Comparison and statistical testing are realized in order to compare the different outputs obtained. First, we present the problem tested: multi-objective flow shop problem and multi-objective quadratic assignment problem. Then, we discuss about the parameter setting and performance analysis protocol before detailing results.

### 4.1 Flow shop problem

#### 4.1.1 A bi-objective flow shop problem

The flow shop problem (FSP) represents one category of the numerous scheduling problems. Overviews of multi-objective methods applied to scheduling problems are given in [17] and [20]. The flow shop problem can be presented as a set of $N$ jobs $J_1, J_2, \ldots, J_N$ to be scheduled on $M$ machines. Machines are critical resources, i.e., one machine cannot be assigned to two jobs simultaneously. Each job $J_i$ is composed of $M$ consecutive tasks $t_{i1}, t_{i2}, \ldots, t_{iM}$, where $t_{ij}$ represents the $j$th task of the job $J_i$ requiring the machine $m_j$. To each task $t_{ij}$ is associated a processing time $p_{ij}$. Each job $J_i$ must be achieved before the due date $d_j$. Then, we have to minimize two objectives [4]:

$C_{\max}$ : Makespan (Total completion time),

$T$ : Total tardiness.

The task $t_{ij}$ is scheduled at the time $s_{ij}$. The two objectives can be computed as follows [4]:

$$f_1 = C_{\max} = \mathrm{Max}\{s_{iM} + p_{iM} | i \in [1\ldots N]\} \tag{5}$$

$$f_2 = T = \sum_{i=1}^{N} [\max(0, s_{iM} + p_{iM} - d_i)] \tag{6}$$

$C_{\max}$ minimization has been proved to be NP-hard for more than two machines [18] and total tardiness minimization $T$ has been proved to be NP-hard even with one machine [13].

#### 4.1.2 Multi-objective flow shop problem instances

There are no established benchmarks for the flow shop problem with more than two objectives. For our purpose, we use two bi-objective instances with the same dimension to generate one three-objective or four-objective instance each time. For example, $C_{\max}$ and $T$ of the first instance are the first two objectives, $C_{\max}$ and $T$ of the second instance are the third objective and the fourth objective, respectively.

The instances used were proposed by Liefooghe et al. [14]. Some instances are bi-objective extensions of the famous Taillard benchmarks [24]; these examples are suffixed by *_ta* in their name (see Table 2).

#### 4.1.3 Parameters setting

One interest of indicator-based algorithms is that it requires to set only a few parameters. In this section, we will discuss about the running time, the population size, the initialization function and the reference point used for HC computation.

– **Running time:** The running time $T$ is a key parameter in the experiment. We compute the time $T$ for each instance by Eq. 7, in which $N_{\mathrm{Job}}$, $N_{\mathrm{Mac}}$, and $N_{\mathrm{Obj}}$ represent, respectively, the number of jobs, the number of machines, and the number of objectives in an instance (see Table 2).

$$T = \frac{{N_{\mathrm{Job}}}^2 \times N_{\mathrm{Mac}} \times N_{\mathrm{Obj}}}{100} \tag{7}$$

– **Population size:** Experiments realized on IBMOLS show that best results are achieved with a small

**Table 2** The instances of multi-objective flow shop problem (*i_j_k* represents the *k*th multi-objective instance with *i* jobs and *j* machines. Parameters: population size *N*, running time for two objectives $T_2$, running time for three objectives $T_3$)

| Instance 1 | Instance 2 | Dim | $N$ | $T_2$ | $T_3$ |
|---|---|---|---|---|---|
| 20_05_01_ta001 | 20_05_02_ta002 | 20 × 5 | 10 | 40″ | 1′ |
| 20_10_01_ta011 | 20_10_02_ta012 | 20 × 10 | 10 | 1′20″ | 2′ |
| 20_15_01 | 20_15_02 | 20 × 15 | 10 | 2′ | 3′ |
| 20_20_01_ta021 | 20_20_02_ta022 | 20 × 20 | 10 | 2′40″ | 4′ |
| 30_05_01 | 30_05_02 | 30 × 5 | 10 | 1′30″ | 2′15″ |
| 30_10_01 | 30_10_02 | 30 × 10 | 10 | 3′ | 4′30″ |
| 30_15_01 | 30_15_02 | 30 × 15 | 10 | 4′30″ | 6′45″ |
| 30_20_01 | 30_20_02 | 30 × 20 | 20 | 6′ | 9′ |
| 50_05_01_ta031 | 50_05_02_ta032 | 50 × 5 | 10 | 4′10″ | 6′15″ |
| 50_10_01_ta041 | 50_10_02_ta042 | 50 × 10 | 20 | 8′20″ | 12′30″ |
| 50_15_01 | 50_15_02 | 50 × 15 | 20 | 12′30″ | 18′45″ |
| 50_20_01_ta051 | 50_20_02_ta052 | 50 × 20 | 30 | 16′40″ | 25′ |
| 70_05_01 | 70_05_02 | 70 × 5 | 10 | 8′10″ | 12′15″ |
| 70_10_01 | 70_10_02 | 70 × 10 | 20 | 16′20″ | 24′30″ |
| 70_15_01 | 70_15_02 | 70 × 15 | 30 | 24′30″ | 36′45″ |
| 70_20_01 | 70_20_02 | 70 × 20 | 30 | 32′40″ | 49′ |
| 100_05_01_ta061 | 100_05_02_ta062 | 100 × 5 | 20 | 16′40″ | 25′ |

population size. Here, we set this size from 10 to 40 individuals according to Eq. 8, relatively to the size of the instance tested (see Table 2).

$$|N| = \begin{cases} 10 & : 0 < |N_{Job} \times N_{Mac}| < 500 \\ 20 & : 500 \leq |N_{Job} \times N_{Mac}| < 1000 \\ 30 & : 1000 \leq |N_{Job} \times N_{Mac}| < 2000 \\ 40 & : 2000 \leq |N_{Job} \times N_{Mac}| < 3000 \end{cases} \quad (8)$$

– **Initialization function:** Random mutations are applied on the archived solutions, completed with random solutions if the archive size is smaller than the population size. The number of mutations applied to the original solutions is 0.3 $t$, where $t$ is the permutation (decision vector) size.
– **Reference point:** Hypervolume and *HC* indicators need a reference point $Z_{\mathrm{ref}}$ which has to be set. We define its coordinates to $[+\infty, +\infty]$, since it allows us to be sure to keep the extreme non-dominated solutions during the search—it has no influence on other non-dominated solutions.

Of course, there are also some other parameters directly related to the problem treated: individual coding (permutation of jobs on the first machine) and the neighborhood operator (insertion operator) [3].

#### 4.1.4 Performance assessment protocol

With this set of experiments, we are aiming to show the efficiency of HBMOLS in comparison with IBMOLS, which was known to be highly efficient on different problems [3].

The quality assessment protocol works as follows: We first create a set of 20 runs with different initial populations for each algorithm and each benchmark instance.

To evaluate the quality of $k$ different sets $A_0 \ldots A_{k-1}$ of non-dominated solutions obtained on a problem instance, we first compute the set PO*, which corresponds to the set of non-dominated solutions extracted from the union of all solutions obtained with the different executions. Moreover, we define a reference point $z = [w_1, \ldots, w_i, \ldots, w_n]$, where $w_i$ corresponds to the worst value for each objective function in $A_0 \cup \cdots \cup A_{k-1}$. Then, to evaluate a set $A_i$ of solutions, we compute the difference between $A_i$ and PO* in terms of hypervolume [27]. This hypervolume difference has to be as close as possible to zero.

For each algorithm, we compute the 20 hypervolume differences corresponding to the 20 runs. In Tables 3, 4, 5, 7, 8 and 9, the average hypervolume differences are given. As suggested in [16], we also perform statistical tests on the sets of hypervolume differences computed. Values which are given in **bold** style means that the corresponding algorithm is **not** statistically outperformed by the algorithm which obtains the best average result (with a confidence level greater than 95%).

**Table 3** Hypervolume differences of FSP (two objectives): average values and statistical best values (in bold)

| Instance | Indicator | | |
|---|---|---|---|
| | $I_\varepsilon$ | $I_{\mathrm{Hyp}}$ | $I_{\mathrm{HC}}$ |
| 20_05_01_ta001 | 0.001887 | 0.001850 | **0.000582** |
| 20_10_01_ta011 | 0.002400 | 0.001419 | **0.000507** |
| 20_15_01 | 0.008451 | 0.009373 | **0.002340** |
| 20_20_01_ta021 | 0.001032 | 0.001101 | **0.000123** |
| 30_05_01 | 0.055391 | 0.053436 | **0.016533** |
| 30_10_01 | 0.064436 | 0.072351 | **0.035049** |
| 30_15_01 | 0.039613 | 0.039592 | **0.020245** |
| 30_20_01 | 0.035141 | 0.038451 | **0.018831** |
| 50_05_01_ta031 | 0.055578 | 0.060626 | **0.036728** |
| 50_10_01_ta041 | 0.094280 | 0.090979 | **0.056749** |
| 50_15_01 | 0.094024 | 0.098653 | **0.069960** |
| 50_20_01_ta051 | 0.108520 | 0.110138 | **0.091586** |
| 70_05_01 | 0.157068 | 0.161081 | **0.075577** |
| 70_10_01 | 0.088532 | 0.121396 | **0.060121** |
| 70_15_01 | 0.102436 | 0.157277 | **0.080052** |
| 70_20_01 | 0.104864 | 0.151036 | **0.073617** |
| 100_05_01_ta061 | 0.168508 | 0.181243 | **0.082768** |

#### 4.1.5 Bi-objective results

The results of bi-objective are summarized in Table 3. HBMOLS statistically outperforms IBMOLS on all instances with a confidence level greater than 95%. The best result is achieved on 20_20_01_*ta*021 instance, where the average hypervolume difference is around 10 times smaller for HBMOLS in comparison with IBMOLS using $I_\epsilon$ or $I_{\mathrm{Hyp}}$ indicator. On many other instances, the average hypervolume difference obtained by HBMOLS is half of those obtained by the two IBMOLS approaches, except on instances 50_20_01_*ta*051 and 70_15_01, where the difference is smaller. For these two instances, it is not easy to improve the quality of the entire population by hypervolume contribution indicator. The test procedure has been undertaken with the performance assessment package provided by Zitzler et al.[3]

#### 4.1.6 Multi-objective results

We use the same test procedure as in bi-objective case to assess the performance of multi-objective case. The results of three-objective are summarized in Table 4. HBMOLS statistically outperforms IBMOLS on all instances with a

[3] http://www.tik.ee.ethz.ch/pisa/assessment.html.

confidence level greater than 95% except on one group of instances 50_10_01_*ta*041 and 50_10_02_*ta*042. For this group, the difference of average hypervolume values obtained by indicator $I_\epsilon$ and $I_{HC}$ is small. Considering another two groups of instances 50_20_01_*ta*051 and 50_20_02_*ta*052, 70_15_01 and 70_15_02, their values are almost the biggest among all the instances, which implies that the quality of the entire population is difficult to improve. The average hypervolume difference for the other groups is around 40% smaller for HBMOLS in comparison with IBMOLS using $I_\epsilon$ or $I_{Hyp}$ indicator.

We carry out experiments on nine groups of instances in four-objective case. The running time of these groups of instances are computed by Eq. 7. The results of four-objective are summarized in Table 5. IBMOLS statistically outperforms HBMOLS on the first 7 groups of instances with a confidence level greater than 95% except on two groups of instances 20_15_01, 20_15_02 20_20_01_*ta*021, and 20_20_02_*ta*022. The average hypervolume differences of these two groups obtained by HBMOLS are very close to those obtained by indicator $I_\epsilon$. Indeed, the solution evaluation of HBMOLS is very expensive in time because of *HC* computation. Besides, the results achieved on last two groups by indicator $I_{HC}$, are the best, since the running time of these big instances is sufficient for HC computation (See Table 5).

## 4.2 Quadratic assignment problem

### 4.2.1 *Single-objective quadratic assignment problem*

The quadratic assignment problem (QAP) is an important problem in theory and practice, which can be described as the problem of assigning a set of facilities to a set of

**Table 4** Hypervolume differences of FSP (three objectives): average values and statistical best values (in bold)

| Instance | Indicator | | |
|---|---|---|---|
| | $I_\varepsilon$ | $I_{Hyp}$ | $I_{HC}$ |
| 20_05_01_ta001 and 20_05_02_ta002 | 0.008820 | 0.017081 | **0.006391** |
| 20_10_01_ta011 and 20_10_02_ta012 | 0.044904 | 0.072363 | **0.013242** |
| 20_15_01 and 20_15_02 | 0.046100 | 0.052205 | **0.017604** |
| 20_20_01_ta021 and 20_20_01_ta022 | 0.045666 | 0.046985 | **0.014360** |
| 30_05_01 and 30_05_02 | 0.049223 | 0.060183 | **0.040111** |
| 30_10_01 and 30_10_02 | 0.119965 | 0.133386 | **0.073294** |
| 30_15_01 and 30_15_02 | 0.109033 | 0.114696 | **0.048600** |
| 30_20_01 and 30_20_02 | 0.113254 | 0.139326 | **0.076801** |
| 50_05_01_ta031 and 50_05_02_ta032 | 0.133615 | 0.123253 | **0.040184** |
| 50_10_01_ta041 and 50_10_02_ta042 | **0.119970** | 0.142391 | **0.114228** |
| 50_15_01 and 50_15_02 | 0.124154 | 0.147594 | **0.092283** |
| 50_20_01_ta051 and 50_20_02_ta052 | 0.144222 | 0.139727 | **0.121222** |
| 70_5_01 and 70_5_02 | 0.082378 | 0.061161 | **0.017095** |
| 70_10_01 and 70_10_02 | 0.116721 | 0.146748 | **0.080265** |
| 70_15_01 and 70_15_02 | 0.149541 | 0.175881 | **0.140359** |
| 70_20_01 and 70_20_02 | 0.132750 | 0.156302 | **0.112108** |
| 100_05_01_ta061 and 100_05_02_ta062 | 0.108150 | 0.116861 | **0.051062** |

**Table 5** Hypervolume differences of FSP (four objectives): average values and statistical best values (in bold)

| Instance | Indicator | | |
|---|---|---|---|
| | $I_\varepsilon$ | $I_{Hyp}$ | $I_{HC}$ |
| 20_05_01_ta001 and 20_05_02_ta002 | **0.031935** | 0.044096 | 0.043522 |
| 20_10_01_ta011 and 20_10_02_ta012 | **0.072619** | 0.118593 | 0.088238 |
| 20_15_01 and 20_15_02 | **0.070087** | 0.091664 | **0.072850** |
| 20_20_01_ta021 and 20_20_01_ta022 | **0.085270** | 0.104347 | **0.086623** |
| 30_05_01 and 30_05_02 | **0.089932** | 0.106148 | 0.135253 |
| 30_10_01 and 30_10_02 | **0.117082** | 0.158320 | 0.146288 |
| 30_15_01 and 30_15_02 | **0.121316** | 0.135059 | 0.132427 |
| 50_05_01_ta031 and 50_05_02_ta032 | 0.143373 | **0.119908** | **0.115970** |
| 70_05_01 and 70_05_02 | 0.194644 | 0.150409 | **0.127535** |

**Table 6** The instances of multi-objective quadratic assignment problem (the symbol ′ − ′ in columns 3 and 4 refers to no corresponding instance, the symbol ′ − ′ in the last two columns means no corresponding running time; parameters: population size $N$, running time for two objectives $T_2$, running time for three objectives $T_3$, running time for four objectives $T_4$)

| Instance 1 | Instance 2 | Instance 3 | Instance 4 | Dim | N | $T_2$ | $T_3$ | $T_4$ |
|---|---|---|---|---|---|---|---|---|
| chr_12_a | chr_12_b | chr_12_c | – | 12 × 12 | 10 | 4′48″ | 7′12″ | – |
| chr_15_a | chr_15_b | chr_15_c | – | 15 × 15 | 10 | 7′30″ | 11′15″ | – |
| chr_20_a | chr_20_b | chr_20_c | – | 20 × 20 | 10 | 13′20″ | 20′ | – |
| esc_16_a | esc_16_b | esc_16_c | esc_16_d | 16 × 16 | 10 | 8′32″ | 12′48″ | 17′4″ |
| esc_32_a | esc_32_b | esc_32_c | esc_32_d | 32 × 32 | 30 | 16′20″ | 24′30″ | 32′40″ |
| Lipa_30_a | Lipa_30_b | – | – | 30 × 30 | 20 | 15′ | – | – |
| Ste_36_a | Ste_36_b | Ste_36_c | – | 36 × 36 | 30 | 21′36″ | 33′24″ | – |

locations with given distances between the locations and given flows between the facilities [22]. The goal is to minimize the sum of the product between flows and distances. In fact, the QAP is an NP-hard optimization problem [23].

Given $n$ facilities and $n$ locations, two $n \times n$ matrices $D$ and $F$, where $d_{ij}$ is the distance between location $i$ and $j$, and $f_{rs}$ is the flow between facilities $r$ and $s$, the objective in the QAP is to find an assignment of facilities to locations such that every facility is assigned to exactly one location and no location is assigned more than one facility. Since in the QAP, the number of facilities is the same as the number of locations, such an assignment corresponds to a permutation of the integers in $\{1, \ldots, n\}$. The objective for the QAP can then be formulated as

$$\min_{\phi \in \Phi} \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} f_{\phi_i \phi_j} \tag{9}$$

where $\Phi$ is the set of all permutations of $\{1, \ldots, n\}$, and $\phi_i$ gives the location of item $i$ in a solution $\phi \in \Phi$.

### 4.2.2 Multi-objective quadratic assignment problem

In [11, 21], the authors consider the multi-objective QAP introduced by Knowles and Corne, which has different flow matrices but always keeps the distance matrix. Formally, the goal is to minimize the

$$\min_{\phi \in \Phi} \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} f^{k}_{\phi_i \phi_j}, \; k \in \{1, \ldots, m\} \tag{10}$$

where min refers to obtain the Pareto front, and $f^k_{ij}$ is the $k$th flow between facilities $i$ and $j$. Besides, all the instances of the QAP tested in this paper are provided by R.E. Burkard et al.[4] Exactly, a multi-objective QAP instance is generated by keeping the distance matrix of the first instance and using different flow matrices (see Table 6).

[4] http://www.seas.upenn.edu/qaplib/inst.html.

**Table 7** Hypervolume differences of QAP (two objectives): average values and statistical best values (in bold)

| Instance | Indicator | | |
|---|---|---|---|
| | $I_\varepsilon$ | $I_{\mathrm{Hyp}}$ | $I_{\mathrm{HC}}$ |
| chr_12_a and chr_12_b | 0.002851 | 0.002981 | **0.002185** |
| chr_15_a and chr_15_b | 0.014420 | **0.012719** | **0.010036** |
| chr_20_a and chr_20_b | 0.069174 | 0.064390 | **0.050232** |
| esc_16_a and esc_16_b | **0.000000** | **0.000000** | **0.000000** |
| esc_32_a and esc_32_b | 0.119026 | 0.119634 | **0.087324** |
| Lipa_30_a and Lipa_30_b | 0.220230 | 0.246207 | **0.167518** |
| Ste_36_a and Ste_36_b | 0.816516 | 0.725983 | **0.323517** |

### 4.2.3 Parameters setting

As is for the FSP, there are also four parameters required to set for the QAP. The parameters "initialization function" and "reference point" are set like for the FSP, we only consider how to set the other two parameters "running time" and "population size".

- **Running time:** We compute the time $T$ for each instance by Eq. 11, in which $N_{\mathrm{Dis}}$, $N_{\mathrm{Flow}}$, and $N_{\mathrm{Obj}}$, represent, respectively, the size of distance matrix, the size of flow matrix and the number of objectives in an instance (see Table 6).

$$T = N_{\mathrm{Dis}} \times N_{\mathrm{Flow}} \times N_{\mathrm{Obj}} \tag{11}$$

- **Population size:** Here, we set this size from 10 to 30 individuals according to Eq. 12, relatively to the size of the tested instance (see Table 6).

$$|N| = \begin{cases} 10 & : 0 < |N_{\mathrm{Dis}} \times N_{\mathrm{Flow}}| < 500 \\ 20 & : 500 \le |N_{\mathrm{Dis}} \times N_{\mathrm{Flow}}| < 1000 \\ 30 & : 1000 \le |N_{\mathrm{Dis}} \times N_{\mathrm{Flow}}| < 2000 \end{cases} \tag{12}$$

In order to evaluate the performance of HBMOLS applied to the QAP, we use the same quality assessment protocol, which is given in detail in Sect. 4.1.

**Table 8** Hypervolume differences of QAP (three objectives): average values and statistical best values (in bold)

| Instance | Indicator | | |
|---|---|---|---|
| | $I_\varepsilon$ | $I_{Hyp}$ | $I_{HC}$ |
| chr_12_a and chr_12_b and chr_12_c | **0.000932** | **0.001520** | 0.001573 |
| chr_15_a and chr_15_b and chr_15_c | **0.036520** | 0.042716 | **0.036127** |
| chr_20_a and chr_20_b and chr_20_c | 0.114485 | 0.117231 | **0.103788** |
| esc_16_a and esc_16_b and esc_16_c | **0.000083** | 0.000293 | **0.000051** |
| esc_32_a and esc_32_b and esc_32_c | **0.148244** | **0.135921** | 0.160581 |
| Ste_36_a and Ste_36_b and Ste_36_c | 0.930265 | 1.168162 | **0.618190** |

**Table 9** Hypervolume differences of QAP (four objectives): average values and statistical best values (in bold)

| Instance | Indicator | | |
|---|---|---|---|
| | $I_\varepsilon$ | $I_{Hyp}$ | $I_{HC}$ |
| esc_16_a and esc_16_b and esc_16_c and esc_16_d | **0.004785** | 0.007641 | 0.009373 |
| esc_32_a and esc_32_b and esc_32_c and esc_32_d | **0.159861** | 0.200011 | 0.324591 |

#### *4.2.4 Bi-objective results*

The results of bi-objective QAP are summarized in Table 7. HBMOLS statistically outperforms IBMOLS on all the groups of instances except two groups (chr_15_a and chr_15_b, esc_16_a and esc_16_b) with a confidence level greater than 95%. The best result is achieved on one group of instances esc_16_a and esc_16_b, where the average hypervolume difference equals to 0 by all the indicators. That is because this group of instances is relatively easy in comparison with other groups, we can get the optimal solution in reasonable time. On the other hand, the average hypervolume difference on all the groups of instances obtained by the indicator $I_{HC}$ is always the smallest among the three indicators. As the size of the instance is becoming bigger, the advantage of the indicator $I_{HC}$ is becoming more obvious.

#### *4.2.5 Multi-objective results*

In three-objective case, we test six groups of instances in total. The results of three-objective QAP are summarized in Table 8. HBMOLS statistically outperforms IBMOLS only on two groups of instances (chr_20_a, chr_20_b and chr_20_c; Ste_36_a, Ste_36_b and Ste_36_c) with a confidence level greater than 95%. The average hypervolume difference on another two groups of instances (chr_15_a, chr_15_b and chr_15_c; esc_16_a, esc_16_b and esc_16_c), which is obtained by the indicator $I_{HC}$, is also the smallest. On the left two groups of instances, the indicator $I_{HC}$ is outperformed by the other two indicators $I_\epsilon$ and $I_{Hyp}$.

Especially, we carry out experiments on two groups of instances in four-objective case. The results are given in Table 9. IBMOLS statistically outperforms HBMOLS with a confidence level greater than 95%. As a matter of fact, it usually takes too much time to compute hypervolume contribution for each solution in four-objective QAP, which leads to poor performance of HBMOLS.

## 5 Conclusions and perspectives

In this paper, we have introduced a hypervolume contribution indicator (*HC*) in order to compare and select the solutions for multi-objective optimization algorithms. Since *HC* is based on the whole population, we have more information to make a good choice. We have applied it in a local search-based algorithm to build the HBMOLS algorithm. We have performed experiments on a multi-objective flow shop problem and a multi-objective quadratic assignment problem.

Experimental results showed most of the time, the superiority of *HC* indicator for these problems. We have obtained good results in two and three objectives cases, where *HC* has been proved to be highly effective. However, the computation cost is too high if the number of objectives is larger than 3. On the other hand, we consider that the results presented here are very interesting, since most real-world multi-objective problems are studied in bi-objective or three-objective case.

Besides, the present work opens some perspectives. First, we can try to improve HBMOLS in different ways, such as by tuning the population size automatically according to the problem size and the number of objectives. We can also improve the way that HC indicator evaluates the dominated solutions.

The main perspective is certainly the improvement of HBMOLS to optimize problems with more than three-objective functions. It would be interesting to study the limitation of the proposed HC indicator in terms of complexity and the possible use of approximation algorithms [2]. Besides, we can also consider using the exact method proposed in [5, 12] or the multi-objective evolutionary algorithm based on decomposition proposed in [25].

**Acknowledgments** We are grateful to the referees for their comments and questions which helped us to improve the paper. The work is partially supported by the "Pays de la Loire" Region (France) within the RaDaPop (2009–2013) and LigeRO (2010–2013) projects.

## References

1. Arroyo J, Armentano V (2005) Genetic local search for multi-objective flowshop scheduling problems. Eur J Oper Res 167(3): 717–738
2. Bader J, Deb K, Zitzler E (2008) Faster hypervolume-based search using monte carlo sampling. In: Conference on multiple criteria decision making (MCDM 2008). Springer, New York, pp 313–326
3. Basseur M, Burke EK (2007) Indicator-based multiobjective local search. In: Proceedings of the IEEE congress on evolutionary computation (CEC 2007). Singapore, September, pp 3100–3107
4. Basseur M, Seynhaeve F, Talbi E-G (2002) Design of multi-objective evolutionary algorithms: application to the flow-shop scheduling problem. In: Congress on evolutionary computation, vol 2. Honolulu, USA, pp 1151–1156
5. Beume N (2009) S-metric calculation by considering dominated hypervolume as klee's measure problem. Evol Comput 17(4): 477–492
6. Beume N, Naujoks B, Emmerich M (2007) SMS-EMOA: multiobjective selection based on dominated hypervolume. Eur J Oper Res 181:1653–1669
7. Bradstreet L, While L, Barone L (2008) A fast incremental hypervolume algorithm. IEEE Trans Evol Comput 12(6):714–723
8. Bringmann K, Friedrich T (2008) Approximating the volume of unions and intersections of high-dimensional geometric objects. In: ISAAC'08: proceedings of the 19th international symposium on algorithms and computation, Berlin, Heidelberg. Springer, pp 436–447
9. Bringmann K, Friedrich T (2009) Don't be greedy when calculating hypervolume contributions. In: FOGA'09: proceedings of the tenth ACM SIGEVO workshop on foundations of genetic algorithms. New York, NY, USA, ACM, pp 103–112
10. Coello Coello CA, Lamont GB, Van Veldhuizen DA (2006) Evolutionary algorithms for solving multi-objective problems (genetic and evolutionary computation). Springer, New York, Inc., Secaucus, NJ, USA
11. Day O, Lamont B (2005) Multiobjective quadratic assignment problem solved by an explicit building block search algorithm—momga-iia. In: EvoCOP, volume 3448 of lecture notes in computer science. Springer, New York, pp 91–100
12. Dhaenens C, Lemesre J, Talbi E (2010) K-ppm: A new exact method to solve multiobjective combinatorial optimization problems. Eur J Oper Res 1:45–53
13. Du J, Leung JY-T (1990) Minimizing total tardiness on one machine is NP-hard. Math Oper Res 15:483–495
14. Figueira J, Liefooghe A, Talbi E-G, Wierzbicki A (2010) A parallel multiple reference point approach for multi-objective optimization. Eur J Oper Res 205(2):390–400
15. Knowles JD, Corne DW (2000) Approximating the nondominated front using the pareto archived evolution strategy. Evol Comput 8(2):149–172
16. Knowles JD, Thiele L, Zitzler E (2005) A tutorial on the performance assessment of stochastive multiobjective optimizers. Technical report TIK-Report No. 214, Computer Engineering and Networks Laboratory, ETH Zurich, July
17. Landa-Silva D, Burke EK, Petrovic S (2004) Metaheuristic for multiobjective optimisation, chapter an introduction to multiobjective metaheuristics for scheduling and timetabling. Springer, New York, pp 91–129
18. Lenstra JK, Rinnooy Kan AHG, Brucker P (1977) Complexity of machine scheduling problems. Ann Discrete Math 1:343–362
19. Liefooghe A, Mesmoudi S, Humeau J, Jourdan L, Talbi E-G (2009) Designing, implementing and analyzing effective heuristics. In: Engineering stochastic local search algorithms, volume 5752 of lecture notes in computer science. Springer, New York, pp 120–124
20. Nagar A, Haddock J, Heragu S (1995) Multiple and bicriteria scheduling: a litterature survey. Eur J Oper Res 81:88–104
21. Paquete L, Stuetzle T (2006) A study of local search algorithms for the biobjective QAP with correlated flow matrices. Eur J Oper Res 169(3):943–959
22. Pardalos P, Rendl F, Wolkowicz H (1994) The quadratic assignment problem: a survey and recent developments. In: Proceedings of the DIMACS workshop on quadratic assignment problems, volume 16 of DIMACS series in discrete mathematics and theoretical computer science, pp 1–42
23. Sahni S, Gonzalez T (1976) P-complete approximation problems. J ACM 23(3):555–565
24. Taillard E (1993) Benchmarks for basic scheduling problems. Eur J Oper Res 64:278–285
25. Zhang Q, Li H (2006) A multiobjective evolutionary algorithm based on decomposition. Technical report TIK-report CSM-450, Department of Computer Science, University of Essex
26. Zitzler E, Künzli S (2004) Indicator-based selection in multiobjective search. In 8th international conference on parallel problem solving from nature (PPSN VIII), pp 832–842, Birmingham, UK, September
27. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. Evol Comput 3:257–271
28. Zitzler E, Thiele L, Laumanns M, Foneseca CM, Grunert da Fonseca V (2003) Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans Evol Comput 7(2):117–132