Indicator-Based Multi-Objective Local Search

M. Basseur and E. K. Burke

Abstract— **This paper presents a simple and generic indicator-based multi-objective local search. This algorithm is a direct extension of the IBEA algorithm, an indicator-based evolutionary algorithm proposed in 2004 by Zitzler and Kuenzli, where the optimization goal is defined in terms of a binary indicator defining the selection operator. The methodology proposed in this paper 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 different binary indicators: Those used in IBEA experiments, and also the indicators derived from classical Pareto ranking methods taken from well-known multi-objective evolutionary algorithms of the literature. Experiments show that the best results are obtained using selection indicators which are not only based on Pareto dominance relation. Moreover, the generic local search algorithm presented in this paper and the proposed indicators obtain promising results which lead to a number of future research directions.**

I. Introduction

The application of metaheuristics to multi-objective combinatorial optimization is mainly represented by the translation of single objective optimization techniques into a multi-objective context. For this purpose, several methods have been designed, such as archive maintaining, ranking methods for fitness assignment, and multi-objective diversity maintaining techniques. Many of the algorithms in the literature are evolutionary algorithms, since such methods are easily adaptable to a multi-objective context. Indeed, evolving a population of solutions is a natural way to find a population of compromise solutions.

However, local search methods are known to be efficient in many real-world applications, and especially on large-scale problems. Several papers propose local search for Pareto multi-objective optimization. In [18], 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 [15], a Multi-Objective Genetic Local Search was proposed. The local search method, which is based on aggregations of the objective functions, is incorporated within a multi-objective genetic algorithm. In [16], the Tabu search principle is applied to multi-objective optimization in the Multi-Objective Tabu Search. Several other papers propose multi-objective local search approaches, such as in [25], [22], [26]. Tutorials on multi-objective techniques are presented in [29], [8], and some applications areas are given in [10], including portfolio optimization, airline operations, railway transportation, radiation therapy planning or computer networks.

M. Basseur and E. K. Burke are with the Automated, Scheduling, Optimisation and Planning (ASAP) research group, University of Nottingham, Jubilee Campus, Wollaton road, Nottingham NG8 1BB, UK; e-mail: {bzm,ekb}@cs.nott.ac.uk.

Most of the multi-objective metaheuristics proposed in the literature are designed to address a particular problem, and only a few of them really propose some generic mechanisms. A system designer who would like to define quickly a multi-objective search method could choose from these generic algorithms, such as NSGA-II [9] or SPEA2 [28], but without any real guarantee of their efficiency for the specific problem in hand. However, a multi-objective hyper-heuristic (heuristics to choose heuristics [5], [23]) approach has been developed which aims to raise the level of generality of decision support systems [7].

In this paper, we aim to adapt several recent ideas from the multi-objective evolutionary algorithms of the literature in order to propose a simple, generic, but efficient, multi-objective local search algorithm. The resulting local search proposed in this paper could be viewed as an elitist evolutionary algorithm since it attempts to evolve, by neighbourhood exploration, a population of compromise solutions.

Our proposed approach is mainly based on an algorithm recently proposed by Zitzler and Kuenzli [27], called the IBEA (*Indicator-Based Evolutionary Algorithm*). As described by the authors, *"IBEA is based on quality indicators where a function I assigns each Pareto set approximation a real value reflecting its quality [31]: Then the optimization goal becomes the identification of a Pareto set approximation that minimizes (or maximises) I"*. As such, they say, *"I induces a total order of the set of approximation sets in the objective space, in contrast to the classical aggregation functions like weighted sum that operate on single solutions only and gives rise to a total order of the corresponding objective vectors"* (see [27]). In [17] and [27], different indicator-based multi-objective optimizers have been proposed. The main advantage of the indicator concept is that no additional diversity preservation mechanisms are required. Zitzler and Kuenzli [27] have demonstrated that indicator-specific search can yield results which are superior to popular algorithms such as SPEA2 and NSGA-II and with respect to the indicator under consideration. Furthermore, since the principle is simple, it could be adapted to other types of problem approaches. For instance, the indicator-based search has been proposed also in [11], and successfully adapted to optimization with uncertainty [3].

In this paper, we will extend the principle of indicator-based selection to multi-objective local search. The decision maker will define a binary indicator which will be used to define which neighbour is good, or not. Then we propose a simple and generic indicator-based multi-objective local search (IBMOLS), which could be easily adapted in respect

1-4244-1340-0/07/$25.00

to the decision-maker preferences.

This paper has two main motivations:

- proposition of a simple and generic multi-objective local search algorithm. The IBMOLS algorithm proposed here has different specificities, which allow us to say that it is simple and generic: (1) The proposed algorithm has only a few parameters; (2) no diversity preservation mechanism is required, the diversity of the population should be contained and optimised by the use of the binary indicator defined by the decision maker; (3) the local search deals with a fixed population size, which enable it to find multiple non-dominated solutions in a single run, without any specific mechanism dedicated to control the number of non-dominated solutions during the multi-objective local search.
- the presentation of alternative Pareto dominance and aggregation based algorithms which compose the major part of the multi-objective local search literature (figure 1): A well-known and reported criticism of aggregation-based algorithms is the non-optimality of the non-supported Pareto solutions [20]; in Pareto dominance-based algorithms, no quantification of the difference between two solutions is provided, but only a possible order relation between these two solutions.

The approach proposed in this paper is slightly different to the approaches usually found in the literature, which use aggregation of the objective functions, or the Pareto dominance relation. The main drawbacks of aggregation and Pareto dominance based methods are schematised in figure 1 (bi-objective case).

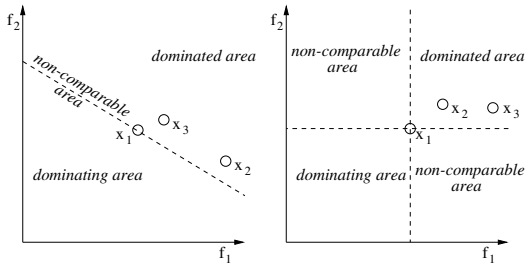

Fig. 1. Aggregation dominance relation (left hand side): using the aggregation method, x_2 and x_3 have a comparable fitness value, but x_2 is more interesting since this solution is non-dominated - in a general way, aggregation methods are not well suited to finding non supported solutions [20]. Pareto dominance relation (right hand side): using the Pareto dominance concept, x_1 dominates x_2 and x_3, but there is no direct way to quantify the difference in terms of dominance.

In this paper, we aim to show the interest of the use of a binary indicator for fitness assignment. In order to evaluate the quality of the indicator based approach, we adapt the classical multi-objective Pareto dominance-based ranking methods of the literature, and we compare them with other indicators, such as those proposed in [27]. Our aim is to show the simplicity of the indicator search concept, and to demonstrate with experiments that IBMOLS obtains good results, and that search methods which are only based on dominance relations are not necessarily the most efficient.

The paper is organised as follows. In section 2, we will briefly present the indicator-based optimization principle, and we define classical multi-objective ranking techniques in terms of binary indicators. In section 3, the IBMOLS algorithm is described, together with its iterative version where the population initialisation is realized in different ways. In section 4, we present the experimental results, obtained with the application of IBMOLS to a bi-objective permutation flow-shop scheduling problem. Then the conclusions and perspectives are discussed in section 5.

II. Indicator-based optimization

Before introducing the concept of indicator-based optimization, let us introduce some useful notations and definitions, partially taken from [27] and [3]. 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 minimised. 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. A Pareto optimal solution is defined as follows:

Definition 1: $x \in X$ is said to be Pareto optimal if and only if a solution $x_i \in X$ which *dominates* x does not exist.

Definition 2: 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 $j \in \{1, \ldots, n\}$.

The relation $x_1 \succ x_2$ means that the solution x_1 is *preferable* to x_2. Since generating the entire set of Pareto optimal solutions is usually infeasible, the overall goal is to identify a good approximation of the Pareto optimal set. Different notions of what a good Pareto set approximation is are possible, and the definition of approximation quality strongly depends on the decision maker and the optimization scenario. As in [27], we here assume that the optimization goal is given in terms of a binary quality indicator I.

A binary quality indicator [31], can be regarded as a continuous extension of the concept of Pareto dominance on sets of objective vectors. The value $I(A, B)$ quantifies the difference in quality between two sets of objective vectors A and B. Now, if R denotes the set of Pareto optimal solutions (or any other reference set), then the overall optimization goal can be formulated as:

$$\text{argmin}_{A \in \mathcal{M}(X)} \; I(A, R) \tag{1}$$

where $\mathcal{M}(X)$ is the space of *objective vector sets*. Since R is fixed, I actually represents a unary function that assigns to each Pareto set approximation a real number; the smaller the number, the more preferable is the approximation.

The indicator could be used to compare two single solutions, or a single solution against an entire population. With such a comparison, the indicator can be used to establish the

selection process of evolutionary algorithms [27]. Indeed, the solutions to delete (respectively select) from the population should be those which have the worst (respectively best) indicator value according to the current population. In other words, during the selection process, the goal is to delete the solution with the smallest degradation of the overall quality of the population, in terms of the quality indicator used.

We first propose to use the two indicators presented in [27]: The epsilon indicator (I_ϵ - equation 2) and the hypervolume indicator (I_{HD} - equation 3).

$$I_\epsilon(x_1, x_2) = max_{i\in\{1,\dots,n\}}(f_i(x_1) - f_i(x_2)) \qquad (2)$$

$I_\epsilon(x_1, x_2)$, $x_1 \in X$ and $x_2 \in X$, represents the minimal translation on which to execute x_1 so that it dominates x_2 (see figure 2). Let us note that the translation could take negative values, and that we assume throughout the paper that all the objective functions are normalised.

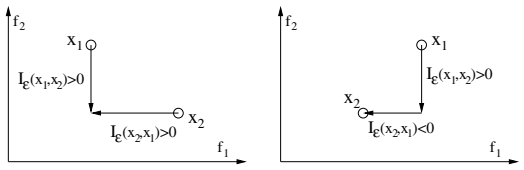

Fig. 2. Illustration of the I_ϵ indicator applied to two solutions x_1 and x_2 (left hand side: no dominance relation between x_1 and x_2; right hand side: $x_2 \succ x_1$).

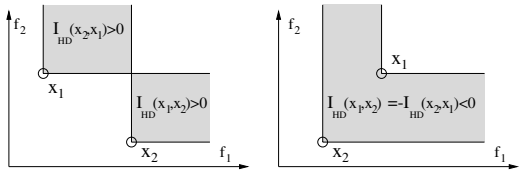

Fig. 3. Illustration of the I_{HD} indicator applied to two solutions x_1 and x_2 (left hand side: no dominance relation between x_1 and x_2; right hand side: $x_2 \succ x_1$).

$$I_{HD}(x_1, x_2) = \begin{cases} H(x_2) - H(x_1) & \text{if } x_2 \succ x_1 \\ H(x_1 + x_2) - H(x_1) & otherwise \end{cases} \qquad (3)$$

$H(x_1)$ represents the volume of the space that is dominated by x_1. $I_{HD}(x_1, x_2)$ represents the volume of the space that is dominated by x_2 but not by x_1 (figure 3).

In order to evaluate the quality of solutions according to a whole population P and a binary indicator I, several different approaches could be defined:

- One possibility is to simply sum up the indicator values for each population member with respect to the rest of the population (equation 4). The value obtained takes into account every solution in the population.

$$I(P \setminus \{x\}, x) = \sum_{z\in P\setminus\{x\}} I(z, x) \qquad (4)$$

- One could be interested in finding the solution which obtains the minimal indicator value against x, i.e. the best solution according to x and I (equation 5). The computed value is not influenced by the dominated solutions of the population.

$$I(P \setminus \{x\}, x) = min_{z\in P\setminus\{x\}}(I(z, x)) \qquad (5)$$

- Lastly, we consider a tradeoff between these two approaches, which is an additive approach that amplifies the influence of dominating population members over dominated ones (equation 6, where $\kappa > 0$ represents the scaling factor). In our experiments, we will use this formulation for the I_ϵ and I_{HD} indicators.

$$I(P \setminus \{x\}, x) = \sum_{z\in P\setminus\{x\}} -e^{I_\epsilon(z,x)/\kappa} \qquad (6)$$

Please note that when $\kappa \to 0$, the same order relation between solutions is obtained with equations 5 and 6, with one difference: when two solutions have the same minimal indicator value with equation 5, equation 6 enables a decision between them according to the second minimal indicator value computed for these solutions. Then, values near to 0 are preferred for κ.

As shown later in the paper, most of the classical ranking techniques could be used to defined binary indicators, and use equation 5 or 6 to combine the indicator values.

In order to be considered as a natural extension of the Pareto dominance concept, the indicator defined has to be compliant with the Pareto dominance relation. As defined in [27], a binary indicator I has to verify the dominance preserving property (definition 3).

Definition 3: A binary indicator I is denoted as dominance preserving if:
(1) for all $x_1, x_2 \in X$,
$x_1 \succ x_2 \Rightarrow I(\{x_1\}, \{x_2\}) < I(\{x_2\}, \{x_1\})$, and
(2) for all $x_1, x_2, x_3 \in X$,
$x_1 \succ x_2 \Rightarrow I(\{x_3\}, \{x_1\}) \geq I(\{x_3\}, \{x_2\})$.

I_ϵ and I_{HD} are proved to be dominance preserving in [27]. Moreover, the majority of the multi-objective ranking techniques in the literature are easily adaptable into binary quality indicators and verify the dominance preserving relation. The expressions obtained are very simple and could be used as a comparison of the different available indicators. Some classical multi-objective ranking techniques from the literature are adapted into binary indicators below, without taking into account the diversity maintaining mechanisms of the corresponding algorithms.

First, the ranking method proposed by Bentley and Wakefield [4], approximatively corresponds to the binary indicator defined in equation 7. In this equation, we keep only the Pareto dominance relation part of the original method proposed by Bentley and Wakefield, since one goal of this paper is to evaluate the quality of I_ϵ and I_{HD} according to Pareto dominance based indicators. By using the additive combination of the indicator values, we obtain equation 8, which corresponds approximatively to the ranking method of Bentley and Wakefield ranking method.

$$I_{Ben}(x_1, x_2) = \sum_{i \in \{1,\ldots,n\}} -\phi(f_i(x_1), f_i(x_2)), \quad (7)$$

$$\text{with } \phi(f_i(x_1), f_i(x_2)) = \begin{cases} 1 & \text{if } f_i(x_1) < f_i(x_2) \\ \frac{1}{2} & \text{if } f_i(x_1) = f_i(x_2) \\ 0 & \text{otherwise} \end{cases}$$

$$I_{Ben}(P, x) = \sum_{z \in P} (I_{Ben}(z, x)) \quad (8)$$

The ranking method of Fonseca and Flemming [12] (equations 9 and 10), can be adapted in the same way, and we obtain a similar formulation which, in this case, exactly corresponds to the original ranking method.

$$I_{Fon}(x_1, x_2) = \begin{cases} -1 & \text{if } x_1 \succ x_2 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

$$I_{Fon}(P, x) = \sum_{z \in P} (I_{Fon}(z, x)) \quad (10)$$

Lastly, the well known ranking method proposed in NSGA-II by Srinivas and Deb [24] is described by equations 11 and 12. Let us note that this indicator uses the *min* combination method, and we consider that the fitness of x_1 is known.

$$I_{Sri}(x_1, x_2) = \begin{cases} I_{Sri}(P, x_1) - 1 & \text{if } x_1 \succ x_2 \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

$$I_{Sri}(P, x) = min_{z \in P}(I_{Sri}(z, x)) \quad (12)$$

With the equations 8, 10, and 12, we have formulated several classical Pareto ranking methods (from respectively [4], [12] and [24]). Particularly, the ranks obtained by 10 and 12 during the selection process exactly corresponds to the ranks obtained by the corresponding Pareto ranking method. Let us note that several ranking methods, such as those used in the SPEA2 algorithm [28], are not adaptable into binary indicators.

We will apply all these indicators as selection operators for the IBMOLS algorithm, which is presented in the next section.

III. Indicator-Based Multi-Objective Local Search

We propose here a general indicator-based multi-objective local search (IBMOLS), which uses the basic principle of local search and is focused upon indicator-based fitness assignment. We first present the IBMOLS baselines, then we discuss the parameter values.

A. Algorithm description

The IBMOLS algorithm maintains a population P. Then, it generates the neighbourhood of an individual in P until a good solution is found, i.e. one which is better than at least one solution of P in terms of the indicator being used. By iterating this principle to all the solutions in P, we obtain a local search step. The entire local search is terminated when the archive A of non-dominated solutions has not received a new solution during a complete local search step. A detailed description of IBMOLS is outlined in algorithm 1.

Algorithm 1 (Baseline Algorithm: IBMOLS):

Input: N (population size), I (binary quality indicator)
Output: A (Pareto approximation set)
Step 1: **Initialisation:** 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, i.e., $Fit(x) = I(P \setminus \{x\}, x)$.
Step 4: **Local search step:** for all $x \in P$ do:
repeat
1) $x^* \leftarrow$ one neighbour of x
2) compute x^* fitness: $I(P, x^*)$
3) update all $z \in P$ fitness values: $Fit(z) += I(x^*, z)$
4) $w \leftarrow$ worst individual in P
5) remove w from P.
6) update all $z \in P$ fitness values: $Fit(z) -= I(w, z)$
until all neighbours are explored or $w \neq x^*$ (i.e. a new solution is found).
Step 5: **Termination:** $A \leftarrow$ non dominated solutions of $A \cup P$. If A does not change, then return A; else perform another local search step.

Algorithm 2 (iterated IBMOLS algorithm):

Input: N (population size), I (binary quality indicator)
Output: PO (Pareto approximation set)
Step 1: $PO \leftarrow \emptyset$
Step 2: while running time not reached do:
1) $P \leftarrow generateNewSolutions(PO, N)$
2) $A \leftarrow IBMOLS'$ output (initialized with P)
3) $PO \leftarrow$ non dominated solutions of $PO \cup A$
Step 3: Return PO.

In this paper, the neighbourhood will be generated in a random order, each neighbour being generated once at maximum. Moreover, we assume that objective values of the solutions are normalised; then, extreme values of the population are computed after the initialisation process and after each new solution is inserted into the population. When an indicator value is computed, the extreme values are used in order to scale the objective functions in the interval $[0, 1]$.

Local search methods are usually used in an iterative way, in order to increase the chance of finding good local optima. In our experiments, we will test an iterative IBMOLS algorithm. Iterated local search algorithms imply the use of a solution(s) initialisation function for each local search. The overall execution process, outlined in algorithm 2 works as follows: a Pareto set approximation PO is maintained and actualised with the solutions found by IBMOLS. After each local search a new initial population is created for the next IBMOLS execution.

B. Parameters

In order to design a generic metaheuristic, the number of parameters which are sensitive to the problem treated has to be reduced as much as possible. IBMOLS algorithm is defined by only three main parameters, that have to be fixed,

defined dynamically during the search or fixed according to the instance considered. These parameters are the population size, the binary indicator and the function which initialises the population. We do not consider the neighbourhood structure, which is a problem-oriented parameter. We discuss these parameters below.

Population size: the most intuitive multi-objective local search consists of maintaining a set A of non-dominated solutions, then generating the neighbourhood of each solution A', then extracting the non-dominated solutions of $A \cup A'$, and repeating the process until improvement is realized [10]. The main problem with this algorithm is the population size which is not fixed, and which greatly depends on the problem and objective functions considered. In extreme cases, one can obtain, during the search, only one non-dominated solution, which implies a great loss of diversity, or a number of non-dominated solutions which grow exponentially and which radically slow down the convergence. The IBMOLS algorithm deals with a fixed population size, which allows us to avoid this problem. In this paper, we do not provide a performance analysis of different population sizes. In our experiments, this size has been fixed to 10, and 20 for large size instances.

Binary indicator: the evaluation of different binary indicators is the main objective of the paper. We will compare the efficiency of two binary indicators (I_ϵ and I_{HD}) previously presented in [27] against other ones, which are based on the dominance relation (I_{Ben}, I_{Fon} and I_{Sri}). The indicator used in the IBMOLS algorithm has a significant influence on determining its efficiency. Experiments are used to determine the most efficient indicators.

Population initialisation: in most metaheuristics, populations are randomly initialised. The first initial population is randomly created here. However, once the first local search is terminated, the initialisation function is very important. Even if the initial population is entirely created randomly, it is crucial to keep information about good solutions when we iterate the local search process.

Different versions of an iterative IBMOLS could be designed, according to the initialisation function being used in the algorithm. The populations can be initialised randomly i.e. without keeping informations from the archived non-dominated set of solutions. The populations can also be initialised by applying crossover on pairs of solutions but the quality of the created solutions depends on the quality of the crossover operator. Here we will consider the following initialisation method: We apply random noise on N randomly selected and different solutions of PO, such as in the basic simulated annealing algorithm [6]. If the size of the archive PO is less than N, then all individuals of PO are selected, and the missing solutions are filled with random individuals.

IV. Experiments

In this section, we present our experiments realized on a typical permutation problem: A bi-objective flow-shop problem. More precisely, the type of the problem is $F/permu, d_i/(C_{max}, \bar{T})$ according to the notation of Graham et al. [13]. An overview of multi-objective methods applied to scheduling and timetabling is given in [21]. Briefly, the problem considered here consists of several jobs which have to be scheduled to a number of machines in a predefined order. Jobs and machines are critical resources, and different processing times are defined for each couple (job, machine). The two objective functions considered are the total completion time, and the sum of tardiness (also called mean tardiness), which are both to be minimised and are NP-hard. For further information about the problem and the benchmark instances used for these experiments, please refer to [2]. As shown in [1], genetic algorithms, and especially NSGA-II, are strongly outperformed by local search based algorithms, on this problem. However, as shown in [1], the application of a basic local search algorithm, keeping all the non-dominated solutions during the neighborhood search, implies a very long computation time on large instances. IBMOLS allows the combination of the local search principle and a population with a fixed size.

Note that we do not evaluate the IBMOLS algorithm against other algorithms from the literature, but we aim to evaluate the quality of the binary indicators derived from these studies (the classical multi-objective evolutionary algorithms, such as NSGA-II or SPEA2 have their own specificities, for instance the diversity maintaining mechanism).

A. Parameter setting

A comparative study has been carried out on the different binary indicators. For the initialisation of the populations, the number of random mutations applied on the archived solutions is $0.3t$, where t is the permutation (decision vector) size. For the I_ϵ and I_{HD} indicators, κ has been set to 10^{-3}. Moreover, I_{HD} needs a reference point, which has been set to $[2, 2]$ (normalised values), as suggested in [27]. The population size N and the running time T are defined according to the size of the instance (see table I). Runs are realized on a P4 - 2.4GHz machine, with 1Go RAM.

The execution times are very short in this study (see table I), in order to be able to provide statistical analysis of the different executions. However, the results obtained are close to the best values found in [2], with very long execution time.

TABLE I

Parameter settings: Population size N and running time T. $ta\_\#i\_\#j\_\#k$ represents the k^{th} bi-objective instance with i jobs and j machines.

Instance	N	T	Instance	N	T
ta_20_5_01	10	20”	ta_20_20_01	10	2’
ta_20_5_02	10	20”	ta_50_5_01	10	5’
ta_20_10_01	10	1’	ta_50_10_01	20	10’
ta_20_10_02	10	1’	ta_50_20_01	20	20’

B. Performance assessment protocol

We created a set a 20 runs with different initial populations for each algorithm and each benchmark instance.

To evaluate the quality of k different sets $A_0 \dots A_{k-1}$ 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 $z = [w_1, w_2]$, where w_1 and w_2 correspond 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 [30]. This hypervolume difference has to be as close as possible to zero (figure 4).

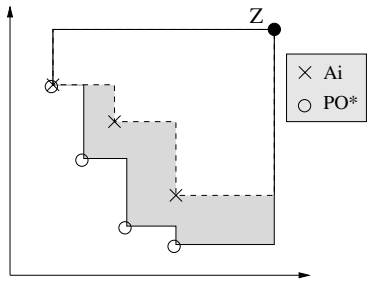

Fig. 4. Illustration of hypervolume difference between a reference set PO^* and a set of non-dominated solutions A_i (shaded area).

For each algorithm, we compute the 20 hypervolume differences corresponding to the 20 runs. Then, we compare each pair of algorithms by statistical testing. We use the Mann-Whitney statistical test, as described in [19]. We obtain P values corresponding to the hypothesis *"the first algorithm performs better than the second one in terms of hypervolume difference"*. This is equal to the lowest significance level for which the null-hypothesis (the medians are drawn from the same distribution) would still be rejected. In Table II, we summarize the P values by giving three main pieces of information:

- $P \leq 0.1\%$: this means that the first algorithm (row) performs better than the second algorithm (column) with a very good confidence level; we will use the symbol "$\succ$" in the corresponding cell.
- $5\% \geq P \geq 0.1\%$: this means that the first algorithm (row) performs better than the second algorithm (column) with a high confidence level which indicates that the first algorithm outperforms the second one; we will use the symbol "$\succeq$" in the corresponding cell.
- $P \geq 5\%$: in this case, the confidence level is too low to conclude with any concrete view; we leave the corresponding cell free.

Note that we obtain similar results using other assessment indicators and/or statistical tests. The test procedure has been undertaken with the useful performance assessment package provided by Zitzler et al. (http://www.tik.ee.ethz.ch/pisa/assessment.html).

C. Results

TABLE II
INDICATOR COMPARISON: I_ϵ, I_{HD}, I_{Ben}, I_{Sri} AND I_{Fon} (HYPERVOLUME DIFFERENCE).

ta_20_5_01

	I_ϵ	I_{HD}	I_{Ben}	I_{Sri}	I_{Fon}
I_ϵ	-	$\succ$	$\succ$		
I_{HD}		-	$\succ$		
I_{Ben}			-		
I_{Sri}		$\succ$	$\succ$	-	
I_{Fon}	$\succeq$	$\succ$	$\succ$	$\succ$	-

ta_20_5_02

	I_ϵ	I_{HD}	I_{Ben}	I_{Sri}	I_{Fon}
I_ϵ	-				
I_{HD}		-	$\succ$		
I_{Ben}			-		
I_{Sri}	$\succeq$	$\succ$	$\succ$	-	
I_{Fon}	$\succ$	$\succ$	$\succ$	$\succ$	-

ta_20_10_01

	I_ϵ	I_{HD}	I_{Ben}	I_{Sri}	I_{Fon}
I_ϵ	-		$\succ$	$\succ$	$\succ$
I_{HD}		-	$\succ$	$\succ$	
I_{Ben}			-		
I_{Sri}			$\succ$	-	
I_{Fon}			$\succ$	$\succ$	-

ta_20_10_02

	I_ϵ	I_{HD}	I_{Ben}	I_{Sri}	I_{Fon}
I_ϵ	-		$\succ$	$\succeq$	$\succeq$
I_{HD}		-	$\succ$		
I_{Ben}			-		
I_{Sri}			$\succ$	-	
I_{Fon}			$\succ$		-

ta_20_20_01

	I_ϵ	I_{HD}	I_{Ben}	I_{Sri}	I_{Fon}
I_ϵ	-	$\succ$	$\succ$	$\succ$	$\succ$
I_{HD}		-	$\succ$		
I_{Ben}			-		
I_{Sri}		$\succeq$	$\succ$	-	
I_{Fon}		$\succ$	$\succ$	$\succeq$	-

ta_50_5_01

	I_ϵ	I_{HD}	I_{Ben}	I_{Sri}	I_{Fon}	
I_ϵ	-	$\succ$	$\succ$	$\succ$	$\succ$	$\succ$
I_{HD}		-	$\succ$	$\succ$		$\succ$
I_{Ben}			-			$\succ$
I_{Sri}			$\succ$	-		$\succ$
I_{Fon}		$\succ$	$\succ$	$\succ$	-	$\succ$

ta_50_10_01

	I_ϵ	I_{HD}	I_{Ben}	I_{Sri}	I_{Fon}
I_ϵ	-	$\succ$	$\succ$	$\succ$	$\succ$
I_{HD}		-	$\succ$	$\succ$	$\succeq$
I_{Ben}			-		
I_{Sri}			$\succ$	-	
I_{Fon}			$\succ$	$\succ$	-

ta_50_20_01

	I_ϵ	I_{HD}	I_{Ben}	I_{Sri}	I_{Fon}
I_ϵ	-	$\succ$	$\succ$	$\succ$	$\succ$
I_{HD}		-	$\succ$	$\succ$	$\succ$
I_{Ben}			-		
I_{Sri}			$\succ$	-	
I_{Fon}			$\succ$	$\succ$	-

We carried out a set of experiments using different binary indicators. The populations are initialised using random mutations on archived solutions. In Table II, the results obtained with five different indicators (I_ϵ, I_{HD}, I_{Ben}, I_{Sri} and I_{Fon}), applied on the iterated $IBMOLS$ algorithm, are outlined. We can extract several pieces of information for each indicator:

- I_ϵ: This indicator obtains the best average results. Indeed, I_ϵ outperforms all the other indicators for the 4 largest instances ($ta\_20\_20\_01$, $ta\_50\_5\_01$, $ta\_50\_10\_01$ and $ta\_50\_20\_01$). I_ϵ outperforms every indicator except I_{HD}, on the $ta\_20\_10\_01$ and $ta\_20\_10\_02$ instances. On the two smallest instances, I_ϵ does not outperform many indicators. However I_ϵ is only shown to be statistically worse than I_{Fon} on the $ta\_20\_5\_01$ and $ta\_20\_5\_02$ instances, and statistically worse than I_{Sri} on the $ta\_20\_5\_02$ instance.
- I_{HD}: This indicator obtains good overall results. Indeed, this is the only one indicator which does not perform statistically worse than I_ϵ on the $ta\_20\_10\_01$ and $ta\_20\_10\_02$ instances. Moreover, even if I_ϵ outperforms I_{HD} on the two largest instances, I_{HD} outperforms the other indicators. On the 4 other instances ($ta\_20\_5\_01$, $ta\_20\_5\_02$, $ta\_20\_20\_01$ and $ta\_50\_5\_01$), I_{HD} obtains average results.
- I_{Fon}: This indicator obtains the best results within the set of indicators which are only based on the dominance relation. The average results of this indicator are comparable to the I_{HD} indicator. However, the good performances are obtained on different instances. I_{Fon} obtains the best results for the 2 smallest instances, and the quality of the algorithm seems to decrease when the problem size increases.
- I_{Sri}: The results obtained are similar, but worse, to those obtained by I_{Fon}.
- I_{Ben}: This indicator performs statistically worse than the other indicators for all the instances.

To summarise, the I_ϵ and I_{HD} indicators tend to outperform the Pareto dominance based indicators, especially when the problem size increases. In order to allow a graphical comparison of stochastic multi-objective optimisers, the concept of an attainment function is described in [14]. The attainment function $\alpha_P(z)$ corresponds to the probability of at least one element of a non-dominated set P dominating a reference point z (refer to [14] for more details). In figure 5, we have represented the minimal values of z where $\alpha_P(z) \geq 0.9$. This figure illustrates the superiority of I_ϵ and I_{HD} against the other indicators on the largest instance considered.

V. Conclusions

In this paper, we drew upon the principle proposed in [27], which defines the selection operator of evolutionary algorithms according to a binary indicator, in order to establish the IBMOLS algorithm. We showed that the classical multi-objective ranking methods could be used as quality indicators. Our experiments provide several interesting pieces of information:

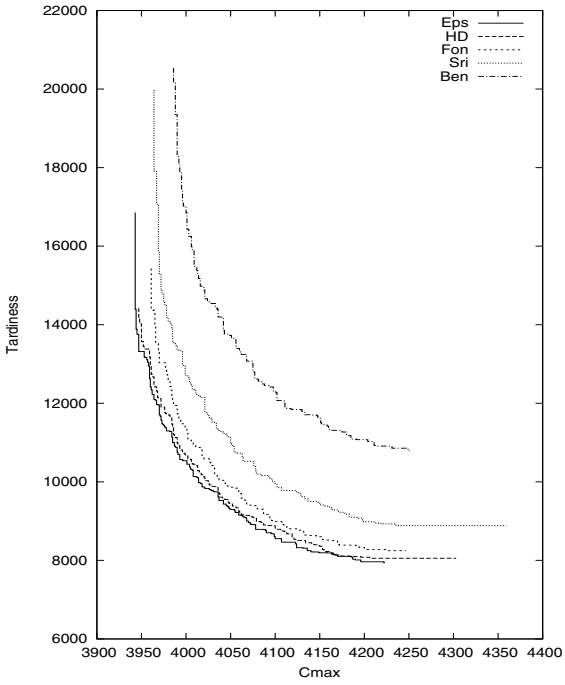

Fig. 5. Empirical attainment functions obtained by the different indicators on the $ta\_50\_20\_01$ instance. The lines correspond to the limit of the objective space which is attained by at least 90% of the runs corried out with a specific binary indicator.

- The quality indicators which are only based on the dominance relation seem to be less efficient than other indicators derived from multi-objective performance assessment research, i.e. the I_ϵ and I_{HD} indicators.
- I_ϵ performs slightly better than I_{HD}. Furthermore, a reference point has to be set for I_{HD} and this indicator needs more computation time, especially with many objective functions.

However, the results obtained are not uniform for each instance, and a lot of future research directions are open. Examples include:

- The indicator seems to depend on instance size: Is the best indicator also dependent on the problem in hand? How do we dynamically select the most useful during the search? We need also to design new and efficient quality indicators and to determine if the quality of the proposed indicators is problem-dependent.
- IBMOLS ability to reach special goals defined by a decision maker has to be evaluated. To achieve the decision-maker preferences, the quality of the different binary indicators has to be re-evaluated.
- A study of the impact of the population size, which is an important parameter is required. Several experiments, which are not detailed in this paper, show that the use of very small or very large population sizes is not useful, and that the optimal size is problem-dependent, and also dependent on the running time available. A study of the impact of the initialisation function could be also useful.

Some of these perspectives could be answered entirely or partially by further experiments on a bi-objective flow-shop problem. The efficiency of the different binary indicators, initialisation functions and population size should be analysed on other multi-objective problems, with two or more objective functions. As for many algorithms, the efficiency of IBMOLS will certainly vary according to the problem treated and its parameter values.

The binary-indicator search principle has been successfully proposed for evolutionary algorithms in [27] and for local search in this paper. This general principle could be adapted to every type of (meta)heuristic search, such as ant colony optimisation and Tabu search [6].

Once we have different indicator-based search strategies, it will be interesting to propose an adaptive version of IBMOLS, which will be efficient on a new problem without preliminary studies. To achieve this goal, the exploration of hyper-heuristics [23] could also lead to significant results. This will help us to search for the adequate population size, binary indicator or initialisation function dynamically during the search, by evolving different indicator-based searches. With such an approach, we could obtain a modified IBMOLS algorithm which will be applicable across a large range of multi-objective problems.

REFERENCES

[1] M. Basseur. *Design of cooperative algorithms for multi-objective optimization: Application to the Flow-shop scheduling problem.* PhD thesis, University of Sciences and Technology of Lille, France, june 2005.

[2] M. Basseur, F. Seynhaeve, and E.-G. Talbi. Design of multi-objective evolutionary algorithms: Application to the flow-shop scheduling problem. In *Congress on Evolutionary Computation*, volume 2, pages 1151–1156, Honolulu, USA, 2002.

[3] M. Basseur and E. Zitzler. Handling uncertainty in indicator-based multiobjective optimization. *International Journal of Computational Intelligence Research (IJCIR)*, 2(3):255–272, 2006.

[4] P. J. Bentley and J. P. Wakefield. Finding acceptable solutions in the pareto-optimal range using multiobjective genetic algorithms. *Soft Computing in Engineering Design and Manufacturing*, 5:231–340, 1997.

[5] E. Burke, G. Kendall, J. Newall, E. Hart, P. Ross, and S. Schulemburg. *Handbook of Metaheuristics*, chapter 16: Hyper-heuristics: an emerging direction in modern search technology, pages 457–474. Kluwer, 2003.

[6] E. K. Burke and G. Kendall, editors. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Springer, 2005. ISBN: 978-0-387-23460-1.

[7] E. K. Burke, D. Landa Silva, and E. Soubeiga. *Meta-heuristics: Progress as Real Problem Solvers*, chapter Multi-objective Hyper-heuristic Approaches for Space Allocation and Timetabling, pages 129–158. Springer, 2005.

[8] K. Deb. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, chapter 10: Multi-objective optimization, pages 273–316. Springer, e. k. burke and g. kendall edition, 2006. ISBN: 978-0-387-23460-1.

[9] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In M. Schoenauer et al., editors, *Parallel Problem Solving from Nature (PPSN VI)*, volume 1917 of *LNCS*, pages 849–858. Springer, 2000.

[10] M. Ehrgott and X. Gandibleux. Approximative solution methods for multiobjective combinatorial optimization. *TOP*, 12(1):1–88, 2004.

[11] M. Emmerich, N. Beume, and B. Naujoks. An EMO algorithm using the hypervolume measure as selection criterion. In *3rd international conference on evolutionary multi-criterion optimization (EMO 2005)*, volume 3410 of *LNCS*, pages 62–76. Springer, 2005.

[12] C. M. Fonseca and P. J. Flemming. Genetic algorithms for multiobjective optimization: Formulation discussion and generalization. In *Fifth International Conference on Genetic Algorithms (ICGA'93)*, pages 416–423, San Mateo, USA, 1993.

[13] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of discrete mathematics*, 5:287–326, 1979.

[14] V. Grunert da Fonseca, C. M. Fonseca, and A. O. Hall. Inferential performance assessment of stochastic optimisers and the attainment function. In *1st international conference on evolutionary multi-criterion optimization (EMO 2001)*, volume 1993 of *LNCS*, pages 213–225. Springer, 2001.

[15] T. H. Ishibuchi. A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Trans. on Systems, Man., and Cybernetics -Part C: Applications and Reviews*, 28(3):1998, Aug 1998.

[16] M. P. Hansen. Tabu search for multiobjective optimization: MOTS. In *MCDM'97 conference*, Cap town, South Africa, Jan 1997.

[17] J. D. Knowles. *Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization*. PhD thesis, University of Reading, 2002.

[18] J. D. Knowles and D. W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149–172, 2000.

[19] J. D. Knowles, L. Thiele, and E. Zitzler. A tutorial on the performance assessment of stochastive multiobjective optimizers. Technical Report TIK-Report No. 214, Computer Engineering and Networks Laboratory, ETH Zurich, July 2005.

[20] J. Koski. Defectiveness of weighting method in multicriterion optimization of structures. *Communications in applied numerical methods*, 1:333–337, 1985.

[21] D. Landa Silva, E. K. Burke, and S. Petrovic. *Metaheuristic for Multiobjective Optimisation*, chapter An Introduction to Multiobjective Metaheuristics for Scheduling and Timetabling, pages 91–129. Springer, 2004.

[22] T. Murata, H. Nozawa, H. Ishibuchi, and M. Gen. Modification of local search directions for non-dominated solutions in cellular multiobjective genetic algorithms for pattern classification problems. In *2nd international conference on evolutionary multi-criterion optimization (EMO 2003)*, volume 2632 of *LNCS*, pages 593–607. Springer, 2003.

[23] P. Ross. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, chapter 17: Hyper-Heuristics, pages 527–556. Springer, e. k. burke and g. kendall edition, 2006. ISBN: 978-0-387-23460-1.

[24] N. Srinivas and K. Deb. Multiobjective optimization using non-dominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.

[25] K. C. Tan, T. H. Lee, and E. F. Khor. Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization. *Evolutionary Computation*, 5(6):565–588, 2001.

[26] A. Viana, J. Pinho de Sousa, and M. A. Matos. Multiobjective constraint oriented neighbourhoods. In *6th Metaheuristics International Conference (MIC 2005)*, pages 896–903, Vienna, Austria, 2005.

[27] E. Zitzler and S. Kuenzli. Indicator-based selection in multiobjective search. In *8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, pages 832–842, Birmingham, UK, September 2004.

[28] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In K. Giannakoglou et al., editors, *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, pages 95–100. International Center for Numerical Methods in Engineering (CIMNE), 2002.

[29] E. Zitzler, M. Laumanns, and L. Thiele. *Metaheuristics for multiobjective optimisation*, volume 535 of *Lecture Notes in Economics and Mathematical Systems*, chapter A tutorial on evolutionary multiobjective optimization, pages 3–37. Springer, 2004.

[30] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *Evolutionary Computation*, 3:257–271, 1999.

[31] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Trans. on Evolutionary Computation*, 7(2):117–132, 2003.