

An approach for the local exploration of discrete many objective optimization problems

Oliver Cuate, Bilel Derbel, Arnaud Liefooghe, El-Ghazali Talbi, Oliver Schütze

▶ **To cite this version:**

Oliver Cuate, Bilel Derbel, Arnaud Liefooghe, El-Ghazali Talbi, Oliver Schütze. An approach for the local exploration of discrete many objective optimization problems. 9th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2017), Mar 2017, Münster, Germany. pp.135-150, 10.1109/TSMCB.2008.926329 . hal-01581433

HAL Id: hal-01581433
https://hal.science/hal-01581433

Submitted on 4 Sep 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An Approach for the Local Exploration of Discrete Many Objective Optimization Problems

Oliver Cuate[1], Bilel Derbel[2], Arnaud Liefooghe[2],
El-Ghazali Talbi[2] and Oliver Schütze[1]

[1] Computer Science Department, CINVESTAV-IPN, Av. IPN 2508,
C.P. 07360, Col. San Pedro Zacatenco, Mexico City, Mexico
ocuate@computacion.cs.cinvestav.mx, schuetze@cinvestav.mx
[2] Univ. Lille, CNRS, Centrale Lille, UMR 9189 – CRIStAL, F-59000 Lille, France
Inria Lille – Nord Europe, F-59650 Villeneuve d'Ascq, France
{bilel.derbel, arnaud.liefooghe, el-ghazali.talbi}@univ-lille1.fr

Abstract. Multi-objective optimization problems with more than three objectives, which are also termed as many objective optimization problems, play an important role in the decision making process. For such problems, it is computationally expensive or even intractable to approximate the entire set of optimal solutions. An alternative is to compute a subset of optimal solutions based on the preferences of the decision maker. Commonly, interactive methods from the literature consider the user preferences at every iteration by means of weight vectors or reference points. Besides the fact that mathematical programming techniques only produce one solution at each iteration, they generally require first or second derivative information, that limits its applicability to certain problems. The approach proposed in this paper allows to steer the search into any direction in the objective space for optimization problems of discrete nature. This provides a more intuitive way to set the preferences, which represents a useful tool to explore the regions of interest of the decision maker. Numerical results on multi-objective multi-dimensional knapsack problem instances show the interest of the proposed approach.

Keywords: Many objective optimization, Multi-Criteria Decision Making, Discrete Optimization, Knapsack, Evolutionary Computation

1 Introduction

In many real-world applications from engineering or finance, one has to face the issue that several objectives have to be optimized concurrently. If more than three objectives are involved, the resulting problem is often termed a *many objective optimization problem* (MaOP) in the literature. Though the treatment of MaOPs is a relatively young research field (so far, mainly problems with two or three objectives have been studied) it is a very important one as the decision making processes are getting more and more important nowadays. One important characteristic of MaOPs is that its solution set, the so-called Pareto set, does typically not consist of one single solution as for 'classical' scalar optimization problems (SOPs, i.e., *one* objective is considered). Instead, the Pareto set

of a continuous MaOP typically forms a $(k-1)$-dimensional object, where k is the number of objectives involved in the problem. For discrete MaOPs, as we will consider here, the magnitude of the solution set typically grows exponentially with k [16]. Specialized evolutionary algorithms have caught the interest of many researchers over the last decades; see, e.g., [12, 11, 32] and references therein. Reasons for this include that these algorithms are applicable to a wide range of problems, are of global nature and hence in principle do not depend on the initial candidate set (i.e., the initial population). Further, due to their set-based approach, they compute a limited-size representation of the entire Pareto set in a single run of the algorithm. Most of these specialized algorithms, called MOEAs (multi-objective evolutionary algorithms) are designed for the treatment of problems with just few objectives (say, 2 to 4). However, more and more algorithms are proposed that deal with many objectives. For instance the use of MAOPs with large population size (e.g. 10,000 individuals) [20], the Dynamical Multi-objective Evolutionary Algorithm (DMOEA) [34] and the Grid-Based Evolutionary Algorithm (GrEA) [31]. However, due to their huge magnitude, the Pareto sets of MaOPs can typically not be computed efficiently, and further on, these sets cannot be visualized adequately. Thus, even with the aid of evolutionary algorithms, if the number of objectives is high, one cannot expect that the solution selected by the *decision maker* (DM) is in fact the preferred solution of the given MaOP with respect to the given setting.

In this paper, we argue that a fine tuning of a selected optimal solution makes sense. More precisely, we propose an approach where, starting from a given initial solution x_0, further solutions x_i, $i = 1, \ldots, N$, are generated such that the sequence of the candidate solutions performs a movement into user-specified directions. Numerical results on different scenarios will show the benefit of the novel approach. We stress that the idea to steer the search along the Pareto set/front into a given user-specified direction is not new. The Directed Search Descent Method [28] and the Pareto Tracer [23] are capable of performing such a search. However, they are both restricted to continuous optimization problems, and they cannot be extended to discrete domains as gradient information is required. Moreover, several specialised MOEAs to solve MaOPs have surged in recent years, for instance in [3] a method which employ the hypervolume for this purpose is proposed and a posteriori method to deal with MaOPs is considered in [7].

The remainder of this paper is organized as follows: in Section 2, we will shortly present the required background and will discuss the related work. In Section 3, we propose a method for the fine tuning of a given solution from the considered MaOP, and we compare possible realizations of this framework. In Section 4, we present some numerical results with a selected approach, and finally, we will conclude and give paths for future research in Section 5.

2 Background

2.1 Definitions

From a mathematical point-of-view, a *multi-objective optimization problem* (MOP) can be defined as follows:

$$\max_{x \in \Omega \subset \mathbb{R}^n} F(x), \text{ s.t. } g(x) \leq 0 \text{ and } h(x) = 0, \tag{1}$$

where $F : D \to \mathbb{R}^k$ is a vector function mapping the *feasible set* $D := \{x \in \Omega \mid g(x) \leq 0 \text{ and } h(x) = 0\}$ to its image $Z := \{F(x) = (f_1(x), \ldots, f_k(x)) \mid x \in D\} \subseteq \mathbb{R}^k$, where $f_i : \Omega \to \mathbb{R}$ stands for the i-th objective. In the combinatorial case, feasible solutions forms a discrete set D. In a maximization context, an objective vector $z \in Z$ is *dominated* by an objective vector $z' \in Z$, denoted by $z \prec z'$, iff $\forall i \in \{1, 2, \ldots, k\}$, $z_i \leqslant z'_i$ and $\exists i \in \{1, 2, \ldots, k\}$ such that $z_i < z'_i$. Similarly, a solution $x \in D$ is dominated by a solution $x' \in D$, denoted by $x \prec x'$, iff $F(x) \prec F(x')$. A solution $x^\star \in D$ is said to be *Pareto optimal*, if there does not exist any other solution $x \in D$ such that $x^\star \prec x$. The set of all Pareto optimal solutions is called the *Pareto set* (PS), and its mapping in the objective space is called the *Pareto front* (PF). One of the most challenging task in multi-objective optimization is to identify a minimal complete Pareto set, i.e., one Pareto optimal solution for each point from the Pareto front. In the combinatorial case, generating a complete Pareto set is often infeasible for two main reasons [16]: (*i*) the number of Pareto optimal solutions is typically exponential in the size of the problem instance, and (*ii*) deciding if a feasible solution belongs to the Pareto set may be NP-complete. Therefore, the overall goal is often to identify a good *Pareto set approximation*. For this purpose, heuristics in general, and evolutionary algorithms in particular, have attracted a lot of attention from the optimization community since the late eighties [10, 12].

One of the most studied NP-hard problem from combinatorial optimization is the *multi-objective (multi-dimensional) 0-1 knapsack problem* (MOKP). Given a collection of n items and a set of k knapsacks, the MOKP seeks a subset of items subject to capacity constraints based on a *weight function* vector $w : \{0,1\}^{k \times n}$, while maximizing a *profit function* vector $p : \{0,1\}^{k \times n}$. More formally, it can be stated as:

$$\begin{array}{rll} \max & \sum_{j=1}^{n} p_{ij} \cdot x_j & i \in \{1, \ldots, k\} \\ \text{s.t.} & \sum_{j=1}^{n} w_{ij} \cdot x_j \leqslant c_i & i \in \{1, \ldots, k\} \\ & x_j \in \{0,1\} & j \in \{1, \ldots, n\} \end{array}$$

where $p_{ij} \in \mathbb{N}$ is the profit of item j on knapsack i, $w_{ij} \in \mathbb{N}$, is the weight of item j on knapsack i, and $c_i \in \mathbb{N}$ is the capacity of knapsack i. For the MOKP, the cardinality of the Pareto set can grow exponentially with the problem size [16], which makes this problem very appealing to investigate. Notice however, that the proposed approach is not specific to the MOKP and could be applied for other combinatorial problems as well.

2.2 Literature overview

The success of MOEAs can be essentially attributed to the fact that they do not require special features or properties from the objective functions, by relying on stochastic search procedures that are able to deal with complex MOPs and a large variety of application domains. MOEAs are in fact inspired by the basic principles of the evolutionary process on a population (a set of individuals or solutions), by means of the so-called evolutionary operators. Considering the wide variety of MOEAs, and the different principles guiding their design, e.g., Pareto dominance [14], indicator-based [5], aggregation-based [32] methods, a lot of progress have been made to better harness the complexity of solving MOPs using an evolutionary process and to better understand the main challenges one has to face. In particular, the dimensionality of the objective space is believed to be one of the main difficult challenges one has to address. In fact, several researchers have pointed out different issues on the use of MOEAs to solve problems having 4 or more objectives [17, 21, 27]. These issues are essentially related to the fact that, as the number of objectives increases, the proportion of non-dominated elements in the population grows. An expression for the portion e in a k-dimensional criteria domain, such that the dominance concept classifies as equivalent solutions is given by $e = \frac{2^k-2}{2^k}$ [17].

In recent years, approaches for improving the behavior and the performance of EMO algorithms in order to deal with so-called MaOPs have received a growing interest. We can classify them in two groups: *(i)* Methods using alternative preference relations, and *(ii)* Methods transforming the original MaOP into a SOP. In the first group, different preference relations were reported such as, to cite a few, the so-called Preference Ordering [15], a generalization of Pareto optimality which uses two more stringent definitions of optimality, or fuzzy relations [17] based on the number of components which are bigger, smaller or equal between two objective vectors. In the second group, different algorithmic approaches can be found such as those based on indicators (e.g., hypervolume [4]), those based on dimensional reduction techniques (e.g., the so-called Pareto Corner Search Evolutionary Algorithm [29]) or those based on space partitioning (e.g., the so-called ϵ Ranking-Evolutionary Multi-objective Optimizer [1]).

It is important to remark that the rationale behind any MOEA is the computation of a *representative* set of solutions from which the DM can eventually pick one or some. However, and apart from difficulties inherent to the optimization/solving process itself, taking the DM requirements into account is a challenging issue when tackling a MaOP. This is essentially because a MOEA might actually fail in providing high-quality solutions in the regions of interest for the DM, or also to entirely cover the Pareto front due to the high dimensionality of MaOPs. For this purpose, taking the DM preferences into account is a hot issue that is being increasingly addressed in the evolutionary multi-objective optimization and multi-criteria decision making communities; see, e.g., [9, 6]. Although reviewing all the literature on the subject is beyond the scope of this paper, we can still comment on three classes that one might encounter [9, 24], namely, (i) *a priori* approaches that aim at guiding the evolutionary process

through pre-defined regions of interest provided by the decision maker, e.g., [22], (ii) *interactive* approaches [26] where the the decision maker preference(s) is iteratively and progressively refined during the evolutionary search procedure until the decision maker is satisfied, and (iii) *a posteriori* approaches, e.g., [18], taking into account the preference of the decision makers solely after the evolutionary process has ended up with a reasonable approximation set.

The work presented in this paper falls at the crossroads of interactive and a priori approaches. In fact, on the one hand, the proposed approach allows the DM to refine a given solution iteratively by providing a direction where to steer the evolutionary search process. On the other hand, we use a reference point approach to transform the direction provided by the decision maker into a reference-point SOP, which is solved using a multi-objective search process. It is to notice that there exist some work in line with our proposal. For instance, Cheng *et al.* [8], and Deb *et al.* [13] proposed different methods that allow the decision maker to attain preferred solutions using reference vectors. However, that methods depart from our proposal in several aspects. First, it falls into the a posteriori class of approaches. More importantly, it aims at providing the decision makers with additional preferred solutions that map different preferred objective regions, whereas we aim at allowing the decision maker to navigate through the Pareto front and to locally explore nearby solutions, hence refining his/her preferences and eventually discovering new preferred solutions. This is similar to the so-called NIMBUS system [25], but our approach allows to include the preferences of the DM modeled using a direction in the objective space more explicitly.

3 Fine tuning method and application to knapsack

3.1 Basic idea and motivations

In the following, we introduce the design principles of the proposed method. Basically, the motivations of our proposal is to allow the DM to navigate along the Pareto front by starting from a given initial solution x_0, possibly coming from the output of any available optimization technique. This initial solution x_0 is to be viewed as a departure point in the objective space from where the DM can refine his/her preferences by discovering on-line the vicinity of x_0, and eventually finding new preferred points in the objective space. Consequently, we shall provide the DM with the necessary tools in order to explore a whole path of solutions being as near as possible to the Pareto front and to locally explore the landscape of Pareto optimal solutions in an iterative manner, i.e., from one solution to a nearby one. For this purpose, the DM is required to provide a direction in the objective space, in which the search process shall be steered. Informally speaking, steering the search along the given direction means providing the DM with a sequence of candidate solutions x_i, $i = 1, \ldots, N$, that can be viewed as forming a path in the objective space and such that every solution x_i is improving the previous solution x_{i-1} with respect to the direction given by the DM in the objective space.

In order to effectively set up this idea for combinatorial MOPs, we need to precisely define the role of the direction provided by the DM and the meaning of improving a given solution according to this direction. Before going further, let us comment that specifying a direction with respect to a starting solution can easily be thought by the DM in many different ways and for different purposes. For the sake of illustration, let us consider a MOP with three objectives (f_1, f_2, f_3) and the following scenario: the DM has an optimal solution for this problem that he/she is not fully satisfied with, e.g., he/she would like to minimize the value of f_2 as much as possible. However, a lot of options can be considered for the above example. For instance, the vector $d_1 = (1, -1, 0)$ can refer to a direction (in the objective space) aiming at reducing the second objective, while increasing the first one. Similarly, the direction $d_2 = (0, -1, 1)$ would imply a reduction of the second objective together with a growth on the third objective. In both cases, we could obtain the minimum value for f_2, but following the direction d_1 or d_2 typically produces different paths that can be associated with different DM preferences. By defining a direction, the DM can actually decides which objectives to improve and which ones to "sacrifice" in order to refine his/her preferences. Performing such local movements in the objective space while following a whole path of Pareto optimal solutions with respect to the preferred direction of the DM is the goal of the proposed framework.

For the target framework to work, it is important to keep in mind the notion of Pareto optimality when performing a movement in the objective space. For instance, assuming that the starting solution is a Pareto optimal solution, then it is obviously not possible to improve all the objectives simultaneously. Consequently, the DM can still define a direction which does not involve an optimal movement, because *no prior knowledge* on the shape of the Pareto front is assumed. In the rest of this section, we provide a step-by-step description of the proposed framework and the necessary algorithmic components for its proper realization. This shall also allow us to better highlight the different issues one has to address in such context.

3.2 Framework for the fine tuning method

As mentioned above, the fine tuning method proposed in this paper is based on the assumption that the DM has a preferred direction d in the *objective space*. More formally, given an initial solution x_0, we assume that the DM is interested in a solution x_1 which is in the vicinity of x_0 such that the following holds:

$$F(x_1) \approx F(x_0) + td, \tag{2}$$

where $t > 0$ is a given (typically small) step size.

As it is very unlikely that such a point x_1 exists where the exact equality in Equation (2) holds (note also that the Pareto front around $F(x_0)$ is not known), we propose to consider a 'best approximation' using an 'approximated' *reference point* Z_1 as follows:

$$Z_1 := F(x_0) + \bar{t}d, \tag{3}$$

Algorithm 1 Fine Tuning Framework

Require: starting point x_0
Ensure: : sequence $\{x_i\}$ of candidate solutions
  for $i = 1, 2, \ldots$ **do**
    Let $d \in \mathbb{R}^m_+$ ▷ Direction for search in objective space
    Let $\delta \in \mathbb{R}_+$ ▷ Step size in objective space
    $Z_i = \text{RefPoint}(x_{i-1}, d)$ ▷ Reference point in step i
    $\text{SOP}_i = \text{G}(Z_i)$ ▷ Next single objective problem to solve
    $x_i = \text{Emo_Optimizer}(\text{SOP}_i)$ ▷ evolutionary search for the next solution
  end for

where $\bar{t} > 0$ is a given, fixed (problem dependent) step size. Then, we propose to compute the 'closest' Pareto optimal solution to the reference point Z_1 in the objective space, which will hence constitute the next point x_1 to be presented to the DM. Notice that we still have to define a metric specifying the closeness of optimal points with respect to the reference point – this will be addressed later. Once x_1 is computed, the DM can consequently change his/her mind or not, by providing a new direction or by keeping the old one. The framework then keeps updating the sequence of reference points and providing the DM with the corresponding closest optimal solutions in an interactive manner. The proposed framework is hence able to provide the DM with a sequence of candidate solutions such that the respective sequence of objective vectors (ideally) performs a movement in the specified direction d.

In Algorithm 1, we summarize the high-level pseudo code of the proposed method. We first remark that the procedure RefPoint implements the idea of transforming the direction d provided by the DM into a reference point, which we simply define as follows:

$$Z_i = F(x_{i-1}) + \delta d, \tag{4}$$

where x_{i-1} is the previous (starting) solution and δ a parameter specifying the magnitude of the movement in the objective space. Actually, δ can be viewed as the preferred Euclidian distance between two consecutive solutions $\|F(x_{i-1}) - F(x_i)\|$ in the objective space. It hence defines the preferred step size of the required movement, which is kept at the discretion of the DM. Notice also that both the direction d and the step size δ can be changed interactively by the DM, which we do not include explicitly in the framework of Algorithm 1 for the sake of simplicity.

Given this reference point, one has to specify more concretely which solution should be sought for the decision maker. This is modeled by function G, which takes the current reference point into account and output a (single-objective) *scalar optimization problem* (SOP) to be solved. At last, the procedure Emo_Optimizer refers to the (evolutionary) algorithm that effectively computes the next solution to be presented to the DM. At this stage of the presentation, it is still not fully clear how to define function G and how to effectively implement the evolutionary solving procedure, which is at the core of this paper. Before going into the technical details of these crucially important issues, let us comment on Fig. 1 showing two hypothetical scenarios in the two-objective case,

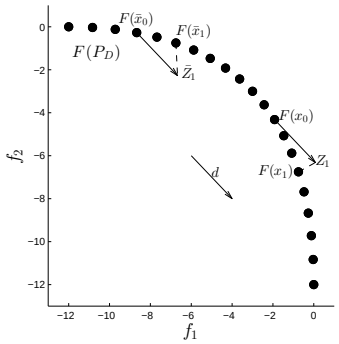


Fig. 1: Illustrative example

chosen for the sake of a better visualization. For $F(\overline{x}_0)$, the reference point $\overline{Z}_1$ is feasible when choosing the direction $d = (1, -1)^T$. That is, there exists a point $\overline{x}$ such that $F(\overline{x}) = \overline{Z}_1$. We want a function $G(\overline{Z}_1)$ that prevents x to be actually chosen. Instead, the solution $\overline{x}_1$ should be a natural candidate since it is a Pareto optimal solution where $F(\overline{x}_1)$ is the closest element to $\overline{Z}_1$ in the Pareto front. The second scenario is for a given point x_0 such that Z_1 is infeasible. Here it is clear that the solution of $G(Z_1)$ must be a Pareto optimal solution whose image $F(x_1)$ is the closest to the given reference point. Notice that in both cases, the Pareto Front is not known when defining function $G(Z)$.

In the following, we propose a possible answer for the definition of G, as well as some alternative (single- and multi-objective) evolutionary procedures for solving the corresponding SOP.

3.3 Framework instantiation

Defining the next single-objective problem to be solved. Since the direction provided by the DM could be arbitrary, and given that we do not assume any prior knowledge neither about the Pareto front, nor on the initial solution from where to steer the search, we propose the following modeling of function $G(x|Z)$, defining the next point to be computed by our framework. We rely on the so-called *Wierzbicki's achievement scalarizing function* (WASF) [30]. More precisely, let $\lambda \in \mathbb{R}^m$ be a weighting coefficient vector (which is different from the direction d provided by the DM). Given a reference point Z, the WASF is defined as follows:

$$g(x|Z, \lambda) := \max_{i=1,\ldots,k} \{\lambda_i(f_i(x) - Z_i)\} + \rho \sum_{i=1}^{k} \lambda_i(f_i(x) - Z_i), \tag{5}$$

where the parameter ρ is the so-called *augmentation coefficient*, that must be set to a small positive value. The motivation of using such a function is that the optimal solution to Problem (5) is a Pareto optimal solution [24], independently of the choice of the reference point. This is an interesting property of the WASF

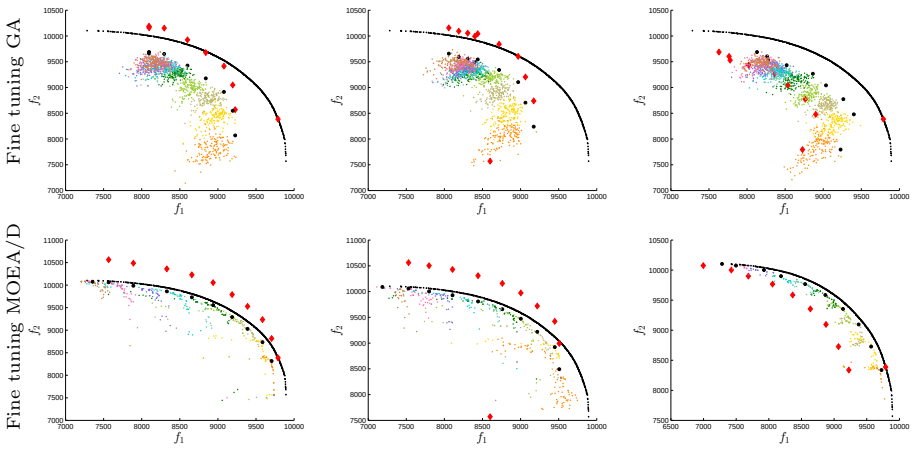
that allows us to deal with reference points that might be defined on the feasible or the infeasible region of the objective space. Notice that this is to contrast to other scalarizing functions, such as the widely-used Chebychev function, that constraint the reference point to be defined beyond the Pareto front.

In our framework, the WASF is intended to capture the DM preferences, expressed by the reference point computed with respect to the DM preferred direction. However, the weighting coefficient vector still has to be specified. It is known that for a given reference point, the solutions generated using different weight vectors are intended to produce a diverse set of solution in the objective space. We here choose to set the weight vector λ as $(1/k, 1/k, \ldots, 1/k)$, which can be viewed as one empirical choice implying a relative fairness among the objectives while approaching the reference point.

The evolutionary solving process. In order to solve the previously defined SOP, we investigate two alternative evolutionary approaches.

The first one consists in using a standard *Genetic Algorithm* (GA). More precisely, and with respect to the experimented knapsack problem, we use the same evolutionary mechanisms and parameters than [2], i.e., a parent selection via a random binary tournament with probability 0.7, an elitist replacement strategy that keeps the best individual, a binary crossover operator with probability 0.5, a single point mutation, and an improve and repair procedure [2] for handling the capacity constraints. However, the initial population of the GA is adapted with respect to the iterations of our proposed framework as follows. Each time the SOP defined by the WASF and the corresponding reference point is updated, we initialize the population with 1/4 of the best individuals from the previous iteration, that we complement with randomly generated individuals. In the first iteration of our framework, the initial population is generated randomly. In our preliminary experiments, this was important in order to obtain a good trade-off between quality and diversity within the evolutionary process. Actually, this observation leads us to consider the following alternative evolutionary algorithm, where population diversity is maintained in a more explicit manner, by using an MOEA for solving the target SOP.

More precisely, our second alternative solving procedure is based on an adaptation of the MOEA/D framework [32]. We recall that MOEA/D is based on the decomposition of a given MOP into multiple subproblems using different weight vectors, which are then solved cooperatively. In contrast to the original algorithm, where the entire Pareto front is approximated using an ideal reference point and a diverse set of weight vectors, typically generated in a uniform way in the objective space; we are here interested in a single solution with respect to the target reference point. Hence, we still consider a set of uniformly-distributed weight vectors, but we use the WASF where the reference point is fixed in order to focus the search process on the region of interest for the DM. At the end of the MOEA/D search process, we output the best-found solution for the weight vector $\lambda = (1/k, 1/k, \ldots, 1/k)$, which precisely corresponds to the target SOP defined with respect to the DM preferred direction. Similarly to the GA, our preliminary experiments revealed that the choice of the initial population for



$F(x_0) = (9789, 8387)$ $d = (0, 1)$ | $F(x_0) = (8600, 7570)$ $d = (0, 1)$ | $F(x_0) = (9789, 8387)$ $d = (-1, 0)$

Fig. 2: Illustrative scenarios on a bi-objectives. The best-known PF approximation is in thin black points. Reference points are shown in red squared points. The output solutions are in shown as circled black point. The population is depicted using a variable color scale. $\delta = 500$.

MOEA/D has an important impact on the quality of the target solution. Accordingly, apart from the first iteration where the initial population is generated at random, we choose to systematically initialize MOEA/D with the population obtained with respect to the previous reference point. Due to the explicit diversity of the MOEA/D population, this initialization strategy revealed a reasonable choice in our initial experiments.

Illustrative scenarios To exemplify the possible scenarios, we experiment in the following the tuning method on the MOKP with different assumptions. This is in order to highlight the behavior of the framework under some possible representative scenarios and to identify the main raised issues.

We define the exemplary scenarios changing the input values of the fine tuning method, i.e., the initial optimal solution $F(x_0)$, the direction in objective space d and the step size δ. For each investigated scenario, we provide plots rendering the computed reference points, the projection of the selected solutions (x_i) in the objective space, and the final population of each of the two considered evolutionary algorithms, together with the best-known PF approximation. This is reported in Fig. 2 for a bi-objective MOKP instance from [33]. Notice that, since we are interested in the impact of the input parameters, we assume that the initial solution x_0 could be optimal or not, which implies that the first-obtained reference point can also be optimal or not. Thus, by simplicity we omit the first update of the reference point, and we consider that $Z_1 = F(x_0)$. At last, we consider 10 iterations of the proposed method, a population size of 150, and $7,500$ function evaluations when running the solving procedure in each iteration.

In Fig. 2 (left), we consider a Pareto optimal solution as a starting point and a fixed direction vector (provided by the DM) corresponding to the scenario where the second objective is to be refined repeatedly. We can clearly see that running the proposed framework is able to gradually improve the output solutions and to effectively steer the search along the desirable input direction. However, the output solutions are not necessary optimal, which we clearly attribute to the relatively few amount of computational effort used when running the evolutionary solving procedure. Interestingly, using the MOEA/D algorithm as a solving procedure (bottom) for the single-objective reference point target problem appears to work much better than the single-objective GA (top). We clearly attribute this to the diversity issues that the evolutionary process is facing when trying to find Pareto optimal solutions. This is confirmed in our second scenario depicted in Fig. 2 (middle), where the initial solution is chosen to be a non-optimal one. This second scenario also demonstrates that the proposed approach behaves in a coherent manner even if the solution considered in each iteration is not optimal. Notice that these two scenarios consider the same preferred direction, which is actually pointing to regions where there exist some non-dominated points. In Fig. 2 (right), we instead consider the scenario where a non-optimal direction $d = (-1, 0)$ is provided, that is a direction that points towards a dominated region of the objective space. This leads to the critical situation where the computed reference point might be dominated. Again, we notice that the proposed approach can handle this situation properly and that the MOEA/D-based solving procedure performs better than the GA.

4 Numerical results

In this section, we present some numerical results of our approach using the modified (fine tuning) MOEA/D as a solver, since it was shown to provide better performance than the fine tuning GA. In order to appreciate the behavior of the proposed method, we consider to compare it against the original MOEA/D algorithm. However, since the original MOEA/D is intended to compute an approximation of the whole PF, a special care has to be taken. First, the proposed method enables to only output a path of solutions based on the computed reference points. Hence, we consider a modification of the Inverted Generational Distance (IGD_Z) [19], which allows us to work with a set of reference points. More specifically, given the set Z of reference points and a reference set archive A, the distance of Z toward $F(A)$ is measured as follows:

$$IGD_Z(F(A), Z) := \frac{1}{|Z|} \sum_{i=1}^{|Z|} \min_{j=1,\ldots,|A|} dist(Z_i^*, F(a_j)), \tag{6}$$

where Z_i^* denotes the point from the PF which is closest to Z_i, i.e., $\|Z_i - Z_i^*\| = dist(Z, F(P_Q))$. Hereby, $dist$ measures the distance between point and set and between two sets as $dist(u, A) = \inf_{v \in A} \|u - v\|$ and $dist(B, A) = \sup_{u \in B} dist(u, A)$, where u and v denote points from sets A and B. Like this, the optimal IGD_Z value is always zero. Notice, however, that the evaluation of

Table 1: Parameter setting of the fine tuning method for numerical results. Number of knapsacks (KS), items, population size (P), maximal number of functions evaluations for each reference point (ZEvs), number of considered reference points ($|Z|$), step size δ, initial reference point Z_0 and desirable direction d_k.

KS	**Items**	**P**	**ZEvs**	$\|Z\|$	δ	Z_0	d_k
2	250	150	7500	10	300	$(10000, 8000)$	$(0, 1)$
2	500	200	10000	10	300	$(16000, 19000)$	$(1, 0)$
3	100	351	17600	10	200	$(4056, 3314, 3228)$	$(0, 1, 1)$
3	100	351	17600	10	200	$(4056, 3314, 3228)$	$(0, 0, 1)$
4	500	455	17500	10	300	$(13643, 14224, 16968, 16395)$	$(1, 1, 0, 0)$
4	500	455	17500	10	300	$(16716, 16867, 14178, 13234)$	$0, 0, 1, 1$

the IGD_Z value requires the knowledge of the exact PF. Instead, we use the best-available PF approximations.

The value of IGD_Z can be straightforwardly computed for our approach using the set of reference points computed at each iteration. For the original MOEA/D, we consider to first extract from the archive maintained by MOEA/D the nearest solutions (in the objective space) to the same reference points computed by our approach. Then, these solutions are considered in order to compute an IGD_Z value for the original MOEA/D. By comparing the IGD_Z values for our method as well as for the original MOEA/D, our intent is to highlight the benefits that can be expected when *locally* steering the search along a preferred direction in an interactive way, against computing a *global* approximation set form which we steer the search *a posteriori*. It is however worth-noticing that such a comparison is only conducted for the sake of illustrating the accuracy of our approach and its effective implementation which should not be considered as an alternative to existing (global) multi-objective optimization algorithms.

In the following, we consider some benchmark instances of the considered MOKP[3], as specified in Table 1, which also summarizes the different parameter setting used for the proposed method. MOEA/D was experimented using the same setting than in the original paper [32]. Notice that, overall, the same number of function evaluations are used for both the original MOEA/D and the proposed method. Table 2 shows the obtained results for the consider scenarios over 20 independent runs for each instance and algorithm.

We notice that, for $k = 2$, the original MOEA/D is able to obtain better results than the proposed Fine Tuning method. This is because MOEA/D can generate points close to the entire PF when the number of objectives is limited. However, we can observe that, the larger the number of objectives, the better the Fine Tuning approach. This is because MOEA/D requires more approximation points and function evaluations in order to cover the entire PF as the dimensionality grows, while the Fine Tuning approach is able to naturally focus on certain regions of the PF. We remark that this fact also improves the execution time,

[3] http://www.tik.ee.ethz.ch/sop/download/supplementary/testProblemSuite/

Table 2: Numerical results. Number of knapsacks (KS), items, population size (P), maximal number of functions evaluations (Ev), and IGD_Z; minimum, average, standard deviation (in small font) and maximum of 20 independent runs are presented for IGD_Z.

KS	Items	P	Ev	IGD_Z					
				Finite Tuning			original *MOEA/D*		
2	250	150	75000	69.86	80.73 (7.02)	90.22	**40.34**	**47.51** (4.73)	**59.01**
2	500	200	100000	241.45	271.04 (15.34)	291.85	**153.12**	**173.79** (10.89)	**192.08**
3	100	351	176000	35.84	**41.62** (3.56)	**47.93**	**34.05**	46.82 (5.13)	57.03
3	100	351	176000	**24.10**	**30.85** (4.82)	41.29	25.63	32.54 (4.35)	**40.79**
4	500	455	175000	**184.73**	**228.05** (26.95)	**289.81**	365.29	494.72 (66.79)	577.90
4	500	455	175000	**144.85**	**198.48** (20.12)	**231.05**	311.34	463.83 (70.58)	589.30

because the Fine Tuning approach does not require any external archive, while for MOEA/D to output a high-quality global PF approximation, an archive is actually used for the MOKP.

5 Conclusions and future work

In this work, we addressed a decision making tool for discrete many objective optimization problems, where we used the multidimensional multi-objective 0-1 knapsack problem as demonstrator. Since the number of non-dominated solutions grows (even exponentially) with the number of objectives k, it becomes difficult or even intractable to compute an approximation of the entire Pareto set for $k \geqslant 4$. Instead, it is likely that each of the chosen solutions obtained by a solver does not represent the most-preferred one from the set of given optimal alternatives. In order to overcome this issue, we proposed a local fine tuning method that allows the search process to be steered from a given solution along the Pareto front in a user-specified direction. More precisely, we presented a framework and two possible realizations of it: one by means of a GA for directly solving the dynamic reference point problem, and another one based on MOEA/D that focuses on a region of the Pareto front delimited by the reference point. Given that only a particular segment of the Pareto front is computed, one retrieves a much more accurate search efficiency compared against the classical method (i.e., aiming to compute *all* Pareto optimal solutions), which we have demonstrated on several benchmark problems. We think that this method can be used as a post-processing step to all existing many objective optimization solvers, and that this will actually help the decision maker to identify his/her most-preferred solution.

Though this work demonstrates as proof-of-principle the application of the novel approach, there is still much to be done. First of all, the tuning of the method is an issue to guarantee in order to get better solutions in lower time, as well as other solvers might be interesting to consider. Next, we think that the tuning method can be extended by other approaches to steer the search along

the Pareto front. Finally, it would be interesting to apply the novel method on many objective optimization problems derived from real-worlds applications in order to appreciate its impact on the decision making process.

References

1. Aguirre, H., Tanaka, K.: Many-objective optimization by space partitioning and adaptive ε-ranking on mnk-landscapes. In: Evolutionary Multi-Criterion Optimization. pp. 407–422. Springer (2009)
2. Alves, M., Almeida, M.: MOTGA: A multiobjective Tchebycheff based genetic algorithm for the multidimensional knapsack problem. Computers & operations research 34(11), 3458–3470 (2007)
3. Auger, A., Bader, J., Brockhoff, D., Zitzler, E.: Articulating user preferences in many-objective problems by sampling the weighted hypervolume. In: Proceedings of the 11th Annual conference on Genetic and evolutionary computation. pp. 555–562. ACM (2009)
4. Bader, J., Zitzler, E.: Hype: An algorithm for fast hypervolume-based many-objective optimization. Evolutionary computation 19(1), 45–76 (2011)
5. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. EJOR 181(3), 1653–1669 (2007)
6. Branke, J., Deb, K.: Integrating User Preferences into Evolutionary Multi-Objective Optimization, pp. 461–477. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
7. Brockhoff, D., Saxena, D.K., Deb, K., Zitzler, E.: On handling a large number of objectives a posteriori and during optimization. In: Multiobjective problem solving from nature, pp. 377–403. Springer (2008)
8. Cheng, R., Jin, Y., Olhofer, M., Sendhoff, B.: A reference vector guided evolutionary algorithm for many-objective optimization (2016)
9. Coello, C.: Handling preferences in evolutionary multiobjective optimization: a survey. In: Evolutionary Computation, 2000. Proceedings of the 2000 Congress on. vol. 1, pp. 30–37 (2000)
10. Coello, C., Lamont, G., Van Veldhuizen, D.: Evolutionary Algorithms for Solving Multi-Objective Problems. Springer, second edn. (2007)
11. Coello, C., Van Veldhuizen, D., Lamont, G.: Evolutionary algorithms for solving multi-objective problems, vol. 242. Springer (2002)
12. Deb, K.: Multi-objective optimization using evolutionary algorithms, vol. 16. John Wiley & Sons (2001)
13. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation 18(4), 577–601 (2014)
14. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. Evolutionary Computation, IEEE Transactions on 6(2), 182–197 (2002)
15. Di Pierro, F., Khu, S.T., Savic, D., et al.: An investigation on preference order ranking scheme for multiobjective evolutionary optimization. Evolutionary Computation, IEEE Transactions on 11(1), 17–45 (2007)
16. Ehrgott, M.: Multicriteria optimization. Springer, second edn. (2005)
17. Farina, M., Amato, P.: On the optimal solution definition for many-criteria optimization problems. In: Proceedings of the NAFIPS-FLINT international conference. pp. 233–238 (2002)

18. Giagkiozis, I., Fleming, P.: Pareto front estimation for decision making. Evol. Comput. 22(4), 651–678 (Dec 2014)
19. Hernández Mejía, J.A., Schütze, O., Cuate, O., Lara, A., Deb, K.: Rds-nsga-ii: a memetic algorithm for reference point based multi-objective optimization. Engineering Optimization pp. 1–18 (2016)
20. Ishibuchi, H., Sakane, Y., Tsukamoto, N., Nojima, Y.: Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations. In: Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on. pp. 1758–1763. IEEE (2009)
21. Knowles, J., Corne, D.: Quantifying the effects of objective space dimension in evolutionary multiobjective optimization. In: Evolutionary multi-criterion optimization. pp. 757–771. Springer (2007)
22. Marler, R., Arora, J.: The weighted sum method for multi-objective optimization: New insights. Structural and Multidisciplinary Optimization 41(6), 853–862 (2010)
23. Martín, A., Schütze, O.: A new predictor corrector variant for unconstrained bi-objective optimization problems. In: EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V, pp. 165–179. Springer (2014)
24. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer, Boston, MA, USA (1999)
25. Miettinen, K., Mäkelä, M.M.: Interactive multiobjective optimization system www-nimbus on the internet. Computers & Operations Research 27(7), 709–723 (2000)
26. Miettinen, K., Ruiz, F., Wierzbicki, A.: Introduction to Multiobjective Optimization: Interactive Approaches, pp. 27–57. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
27. Purshouse, R., Fleming, P.: On the evolutionary optimization of many conflicting objectives. Evolutionary Computation, IEEE Transactions on 11(6), 770–784 (2007)
28. Schütze, O., Martin, A., Lara, A., Alvarado, S., Salinas, E., Coello, C.A.: The directed search method for multiobjective memetic algorithms. Computational Optimization and Applications 63, 305–332 (2016)
29. Singh, Hemantand Isaacs, A., Ray, T.: A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems. Evolutionary Computation, IEEE Transactions on 15(4), 539–556 (2011)
30. Wierzbicki, A.: The use of reference objectives in multiobjective optimization. In: Multiple criteria decision making theory and application, pp. 468–486. Springer (1980)
31. Yang, S., Li, M., Liu, X., Zheng, J.: A grid-based evolutionary algorithm for many-objective optimization. Evolutionary Computation, IEEE Transactions on 17(5), 721–736 (2013)
32. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. Evolutionary Computation, IEEE Transactions on 11(6), 712–731 (2007)
33. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE transactions on Evolutionary Computation 3(4), 257–271 (1999)
34. Zou, X., Chen, Y., Liu, M., Kang, L.: A new evolutionary algorithm for solving many-objective optimization problems. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 38(5), 1402–1412 (2008)