Computers and Operations Research 104 (2019) 1–14

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

Lorenz dominance based algorithms to solve a practical multiobjective problem

Brahim Chabane[a,b], Matthieu Basseur[a], Jin-Kao Hao[a,c,*]

[a] LERIA, Université d'Angers, 2 Bd Lavoisier, Angers 49045, France
[b] GePI Conseil, Grand Maine - Allée du Grand Launay, Angers 49000, France
[c] Institut Universitaire de France, 1 rue Descartes, Paris 75231, France

ARTICLE INFO

Article history:
Received 9 October 2017
Revised 30 November 2018
Accepted 2 December 2018
Available online 3 December 2018

Keywords:
Multiobjective optimization
Fair optimization
Lorenz dominance
Action planning
Decision support
Heuristics

ABSTRACT

The set of Pareto nondominated solutions obtained in some practical cases of multiobjective optimization problems can be huge, rendering decision making difficult. Applying Lorenz dominance instead of Pareto dominance during the optimization process can help to alleviate this difficulty. Lorenz dominance is a refinement of Pareto dominance that integrates fairness in multiobjective optimization when objectives are considered equal and can help select only the well located solutions. By introducing a partial order among a set of Pareto-nondominated solutions, Lorenz dominance reduces the size of the nondominated front by keeping only fair solutions. In this work, we investigate the use of the infinite order Lorenz dominance within three new methods to solve a practical case of the multiobjective knapsack problem, which involves elaborating efficient action plans in social and medico-social structures. We assess the proposed methods on large problem instances with up to 8 objectives and 500 candidate actions and show their effectiveness in comparison with four leading reference algorithms.

1. Introduction

We are interested in a practical action planning problem in social and medico-social structures in France. This problem is a critical step to increase the efficiency of social and medico-social structures. Since the French law No. 2002-2 renovating social and medico-social actions, the social and medico-social sector has been experiencing fast evolutions due to several reasons (The Ministry of Social Affairs and Health, 2012). First, this law considers actions (e.g., planning, resource allocation, structure evaluation and coordination) as a fundamental basis for the management of these structures. Second, the services offered by these structures (more than 34 000 in 2017) become more and more diverse, complexifying the task of action planning. Third, the decline of budgets allocated to the structures in recent years, on the one hand, and the increase of aging population, on the other hand, push decision makers of these structures to find suitable ways to optimize their financial, human and material resources. So, decision makers are now faced to a challenging task of elaborating efficient multiobjective action plans with strong constraints like a tight budget. Even if the social and medico-social sector is increasingly computerized in recent years, the use of computing systems is often limited to daily managing tasks and there is no true decision support system able to assist the managers to make the best choices for the short-term and long-term action plans. In the context of resource restriction and lack of advanced optimization tools, decision making becomes extremely difficult. In this work, we present a multiobjective decision support system to assist managers to optimize their action plans. This work is a part of the "*MSQualité*" toolkit developed by the company *GePI*,[1] which is specialized in the social and medico-social sector in France.

The action planning problem involves elaborating optimized action plans in order to improve the overall management efficiency of a structure and its quality of service (social and medico-social structures should elaborate at least one action plan every five years). The aim is to identify a subset of actions among many candidate actions while optimizing many objectives and satisfying some imperative constraints (e.g., limited budget). Each action has a realization cost and can influence, positively or negatively, some or all the objectives. The global cost of the final solution (i.e., an action plan) should not exceed a predefined budget. Also, a threshold constraint could be added to each objective indicating the minimal objective value that a solution must attain.

* Corresponding author.
E-mail addresses: brahim.chabane@etud.univ-angers.fr (B. Chabane), matthieu.basseur@univ-angers.fr (M. Basseur), jin-kao.hao@univ-angers.fr (J.-K. Hao).

[1] http://www.gepi-conseil.com

https://doi.org/10.1016/j.cor.2018.12.003
0305-0548/

As we explain in Section 2, the action planning problem can be considered as a practical case of the multiobjective knapsack problem (MOKP), where actions represent items (objects) to be added in a knapsack constrained by its capacity (i.e., budget) while optimizing the given objectives. In the practical setting, we may need to consider up to eight objectives simultaneously. The literature offers, in addition to the Indicator-based methods discussed in Section 3.2, a large number of general methods for solving multiple and many objective optimization problems including the MOKP (Alves and Almeida, 2007; Barichard and Hao, 2003; Bazgan et al., 2009; Deb, 2001; Deb and Jain, 2014; Jaszkiewicz, 2002, 2004; Wang et al., 2015; Zhang and Li, 2007; Zitzler and Thiele, 1999). Among these methods, MOEA/D (Ishibuchi et al., 2015; Zhang and Li, 2007), MOGLS (Ishibuchi and Murata, 1998; Jaszkiewicz, 2001), and PICEA (Wang et al., 2015) are known to be relevant and strong representative methods. Considering the specific features of our practical action planning problem, our previous study (Chabane et al., 2017) focused on using the Indicator-based Multiobjective Local Search (IBMOLS) algorithm (Basseur and Burke, 2007; Basseur et al., 2012) combined with the $R2$ indicator (Brockhoff et al., 2012). Indeed, compared to many evolutionary multiobjective methods, IBMOLS has several interesting characteristics that make it suitable for our action planning problem. First, contrary to methods employing the Pareto dominance relation for solution assessment, IBMOLS uses a quality indicator for fitness assignment, and as such does not require any specific diversity preservation mechanism. Second, the decision maker can include preferences in the indicator definition to guide the search to generate relevant solutions for the decision maker. Third, IBMOLS requires only a small number of parameters (i.e., three), making it easy to use in the practical setting. Finally, as a population-based local search algorithm, IBMOLS is particular suitable for large-scale problems and has shown its capacity of solving different multiobjective problems (see Chabane et al., 2017 for application examples).

The study of Chabane et al. (2017) showed that IBMOLS coupled with $R2$ performs well for the considered action planning problem even on large size problem instances (up to 8 objectives and 500 actions). Also, the $R2$ indicator offers two ways for the decision maker to incorporate preferences easily: a reference point and a set of weight vectors. Indeed, we can use the position of the reference point and the weight vector directions of the $R2$ indicator to shrink the search space and guide the search process towards regions of interest in the objective space (Chabane et al., 2017). However, the $R2$-IBMOLS method suffers from two inconveniences for its use in practice: i) the number of nondominated solutions it generates can be very high (several thousands for a large size instance), making it difficult for the decision maker to choose the solution to be implemented; ii) the average runtime of $R2$-IBMOLS can become very high since it critically depends on the number of weight vectors of the $R2$ indicator being considered. Moreover, in the practical setting, the objectives of the action plan could have the same importance for the decision maker. In such a case, obtaining equitable solutions becomes, for the decision maker, as important as the optimization of the action plan. Unfortunately, this aspect is not considered in the $R2$-IBMOLS method. In this work, we aim to mitigate these problems and improve the $R2$-IBMOLS method in order to:

- generate only equitable nondominated solutions of the Pareto front by disqualifying unfair solutions.
- reduce the number of nondominated solutions generated by the method.
- reduce the runtime of the method.

The ultimate goal is to develop a decision support system to assiss managers of social and medico-social structures to elaborate effectively efficient action plans to continually improve the quality of their structures. We summarize the contributions of the work as follows.

First, we investigate three new methods relying on the infinite Lorenz dominance principle (Golden and Perny, 2010) to take into account fairness during the optimization process. We show that by using Lorenz dominance (L-dominance for short) within $R2$-IBMOLS, instead of Pareto dominance (P-dominance for short), we can significantly reduce the number of nondominated solutions and the runtime, without sacrificing efficiency. Indeed, L-dominance (also called equitable dominance) is a refinement of P-dominance, which considers only the well located solutions and leads to a reduced set of nondominated solutions (a Lorenz-optimal front) by selecting only the most equitable solutions. By " well located solutions", we mean the closest solutions to the reference point when decision maker's preferences are given and the solutions located in the center of the objective space when all the objective have the same importance. Indeed, in this study we consider two cases: (i) the decision maker has some preferences about the importance of the objectives. These preferences are expressed by the placement of the reference point in the objective space. In this case, the best solutions are those that are the closest to the reference point. (ii) The objectives have the same importance for the decision maker, in which case the best solutions are those guaranteeing equity of the objectives. These solutions are those situated in the center of the objective space (Fig. 2 of Section 6.2 shows such an example).

Second, the methods investigated in this work are integrated in the *MSQualité* toolkit, which provides the decision maker of a social and medico-social structure with an useful decision support tool to optimize their action plans.

Finally, one notices that studies related to fair optimization with L-dominance are limited compared to the huge body of research on Pareto optimization. This work comes to enrich the field of fair optimization while the proposed techniques can be advantageously integrated into other multiobjective optimization methods such as those mentioned above.

The remainder of the paper is organized as follow. Section 2 presents a formal model of the action planning problem. Section 3 recalls basic definitions about multiobjective optimization, the binary indicator search principle and the IBMOLS algorithm. Section 4 introduces the Lorenz dominance principle and related studies. Section 5 is dedicated to the proposed L-dominance based algorithms for the considered problem. Section 6 shows computational results, followed by concluding comments and perspectives.

2. Action plan optimization problem

This section presents the action plan optimization problem and gives a mathematical formulation of the problem. More details can be found in Chabane et al. (2017).

The action plan optimization problem studied in this work is a practical case of the MOKP, which is a well-known mathematical model with many applications. As mentioned in Section 1, since the law No. 2002-2, social and medico-social structures in France are constrained to continuously elaborate improvement projects. They should carry out at least one self-assessment every 5 years and one external assessment (assessment done by a person outside the structure) every 7 years. This assessment aims to evaluate the operating structure, its improvement project and the quality of service offered to the persons entering in the structure. At the end of the evaluation process, recommendations are given to the decision maker to elaborate a new project for the next period. For the new project, the decision maker defines the objectives to achieve, the resources to use, the constraints and the action plan to implement.

The origins of the actions are either issued from existing action plans in the structure or other similar structures, or are decided by the managers for continuous improvements. The objectives can be diverse, namely qualitative (such as "improve resident's quality of life") or quantitative (such as "increase the resident's autonomy"). Usually, structures do not have the required resources (especially the budget) to realize all feasible and desirable actions. Thus, decision makers must choose, within the available budget, a set of actions in order to maximize the overall quality of the structure while attaining the predefined objectives of the project.We note that, in the real case, we may need to consider up to eight objectives.

An action plan p can be defined as a subset of actions selected among a set of candidate actions A, in order to optimize a set F of conflicting objectives. p can be represented by a vector $p = (a_1, a_2, \ldots, a_n)$ with n equal to the size of A. a_i=1 if the action a_i is selected to be implemented with the action plan and a_i=0 otherwise. The set of the possible action plans (solutions) is denoted by P. Each objective is represented by a function f_j that associates to every action $a_i \in A$ its impact on the objective j. The impact that an action plan $p = (a_1, a_2, \ldots, a_n) \in \{0, 1\}^n$ has on an objective j is obtained by:

$$f_j(p) = \sum_{i=1}^{n} a_i f_j(a_i) \tag{1}$$

An objective j could have a constraint c_j determining the minimal threshold accepted for f_j, this value is fixed by the decision maker. In this work, we consider that all the objectives must be improved (i.e., $f_j(p) \geq c_j > 0$).

An additional constraint concerns the realization cost W of the solution that can not exceed some budget β fixed by the decision maker. Indeed, each action a_i has a realization cost ω_i which can take negative values since there may be actions with negative cost when it is about selling of objects or services for instance. Actions with no cost are also to be taken into account. The global cost of a solution p corresponds to the following cost sum of the actions of p:

$$\begin{cases} W(p) = \sum_{i=1}^{n} a_i\omega_i \\ W(p) \leq \beta \end{cases} \tag{2}$$

So, the optimization goal aims to find $p^* \in \arg\max_{p \in P} F(p)$ verifying:

$$\begin{cases} p^* \in \{0, 1\}^n \\ \forall_j \in \{1, m\}, f_j(p^*) \geq c_j \\ \sum_{i=1}^{n} a_i\omega_i \leq \beta \end{cases} \tag{3}$$

However, p^* is not unique since we deal with a multiobjective case. Instead, we obtain a set of nondominated solutions whose size could be huge and increases with the problem size and the number of objectives. So, it is important in practice to approximate only the optimal solutions that are the most interesting ones for the decision maker.

3. Binary quality indicator based multiobjective optimization

This section is dedicated to the optimization with binary quality indicators. First, we introduce the binary quality indicator principle followed by a focus on the $R2$ indicator. Then, we present the IBMOLS (Basseur and Burke, 2007) and $R2$-IBMOLS (Chabane et al., 2017) methods, which are the base of two of the three proposed methods in this work. Before describing the binary quality indicator, we give some useful definitions.

Let X denote the decision space of a general optimization problem, Y the corresponding objective space, and m the number of objective functions $f_1, f_2, \ldots, f_m$ that assign to each decision vector $x \in X$ a corresponding objective vector $y = \{f_1(x), f_2(x), \ldots, f_m(x)\} \in Y$. We consider through this section that the m objective functions have to be minimized.

Definition 1. The Pareto dominance relation on objective vectors of Y is defined for all y, z by: $y \prec_P z \iff [\forall j \in \{1, \ldots, m\}, y^j \leq z^j$ and $\exists k \in \{1, \ldots, m\}, y^k < z^k]$

The relation $y \prec_P z$ means that, according to the P-dominance relation $\prec_P$, the corresponding solution x_2 to the vector z is P-dominated by the corresponding solution x_1 to the vector y (x_1 is *"preferable to"* or *"better than"* x_2).

Definition 2. $x \in X$ is said to be Pareto optimal (P-optimal for short) if and only if it does not exist another solution $x' \in X$ dominating x.

The P-optimal set denoted by X_P contains all the P-optimal solutions. The image $f(x)$ of a P-optimal solution x in the objective space Y is called a *Pareto-nondominated point*. The image $Y_P = f(X_P)$ of the P-optimal set X_P in Y is called the *Pareto front*. Finding the real Pareto front is not an easy task, especially on large size problems, but approximating this front is usually possible, particularly with metaheuristics.

3.1. Binary quality indicator

The concept of binary quality indicators (Zitzler and Künzli, 2004) is a natural extension of the P-dominance relation on sets of objective vectors. To quantify the difference in quality between two approximation sets A and B in the space of Pareto set approximations Ω, a function $I : \Omega \times \Omega \longrightarrow \mathbb{R}$ assigns to a pair of approximation sets a real value quantifying their difference in quality. The function I can also be used to compare one approximation set A against a fixed reference (e.g., the set of P-optimal solutions). In this case, I represents a unary quality indicator that assigns to each approximation set a real number representing its distance to the reference which has to be minimized. Thus, the optimization goal is transformed to the identification of a set of approximations that minimizes I. Moreover, a quality indicator could be used to evaluate the difference in quality of two single solutions or a single solution against a population of solutions. This evaluation is usually used in the selection process of evolutionary algorithms. Indeed, during the selection process, the solution for deletion from the population should be the one with the worst value of the indicator being used with respect to the rest of the population.

Furthermore, in real word applications, the decision maker is not interested by all nondominated solutions since usually the final decision concerns a unique solution or a small number of solutions. So, several methods integrating decision maker preferences were developed (Bechikh et al., 2015; Wang et al., 2015). In addition to being an effective means for the optimization process, the quality indicator could be defined by the decision maker, according to his (her) preferences. There are several quality indicators in the literature, such as Epsilon indicator (Zitzler and Künzli, 2004), Hypervolume indicator (Lacour et al., 2017; Zitzler and Thiele, 1999) and $R2$ indicator (Brockhoff et al., 2012). As mentioned in Section 1, the $R2$ indicator is interesting because it offers two mechanisms to integrate decision maker preferences: a reference point and a set of weight vectors. Brockhoff et al. showed in Wagner et al. (2013) that the optimal distribution of the solutions can be affected by moving the reference point, restricting the weight space or skewing the weight vectors distribution. The $R2$ indicator is defined below (for more details about the $R2$ indicator and its properties, see Brockhoff et al., 2012; Brockhoff et al., 2014).

The R indicator family is based on utility functions which map a vector $\vec{y} \in R^m$ to a scalar utility value $u \in \mathbb{R}$ for assessing the relative quality of two Pareto front approximation sets (Brockhoff et al., 2012).

Definition 3. For a discrete and finite set U of utility functions, a uniform distribution p over U, and a reference set R, the $R2$ indicator value of a Pareto set approximation A is defined by:

$$R2(R, A, U) = \frac{1}{|U|} \sum_{u \in U} \left(\max_{r \in R}\{u(r)\} - \max_{a \in A}\{u(a)\} \right) \quad (4)$$

When R is constant, the $R2$ indicator can be defined as a unary indicator:

$$R2(A, U) = -\frac{1}{|U|} \sum_{u \in U} \max_{a \in A}\{u(a)\} \quad (5)$$

We use, throughout this paper, the standard weighted Tchebycheff function $u(z) = u_\lambda(\vec{z}) = -\max_{j \in \{1,\ldots,m\}} \lambda_j |z^*_j - z_j|$ within the $R2$ indicator as defined in Eq. (5), where $\lambda = (\lambda_1, \ldots, \lambda_m) \in \Lambda$ is a given weight vector and z^* is an utopian point.

3.2. Indicator-based multiobjective local search (IBMOLS)

Since the publication of the indicator-based evolutionary algorithm (IBEA) proposed by Zitzler and Künzli (2004), the use of indicator-based algorithms in evolutionary multiobjective optimization field is continuously increasing. Several methods and studies using quality indicators were proposed. Among them, we mention the following studies: an EMO algorithm using the hypervolume measure as selection criterion (Emmerich et al., 2005), improving hypervolume-based EMO algorithms by using objective reduction methods (Brockhoff and Zitzler, 2007), the HypE algorithm (Bader and Zitzler, 2011), $R2$-IBEA (Phan and Suzuki, 2013), $R2$-EMOA (Trautmann et al., 2013) and $R2$ indicator-based multiobjective search (Brockhoff et al., 2014). More recently, a simple and fast hypervolume indicator-based multiobjective evolutionary algorithm (FV-MOEA) was presented in Jiang et al. (2015). A $R2$-based multiobjective particle swarm optimizer ($R2$-MOPSO) was introduced in Li et al. (2015).

IBMOLS (Basseur and Burke, 2007) is another multiobjective algorithm combining a quality indicator and a local search mechanism. Indeed, local search is known to be efficient for many real-world applications, especially on large-scale problems. However, most of these algorithms are usually based either on the P-dominance relation or on aggregation methods. By contrast, IBMOLS uses the quality indicator principle for the fitness assignment without requiring any specific diversity preservation mechanism (this aspect should be considered in the indicator definition). Moreover, IBMOLS presents two main advantages: (i) a fixed population size is used during the local search enabling the algorithm to find multiple nondominated solutions in a single run, without any specific mechanism dedicated to control the number of nondominated solutions (problem encountered with the classical Pareto-based multiobjective local search (Paquete and Stützle, 2004)); (ii) IBMOLS requires only a small number of parameters: the population size and the quality indicator.

In Chabane et al. (2017), the IBMOLS approach is combined with the ϵ and $R2$ indicators to solve the multiobjective action plan problem and assessed on simulated data with 50 to 500 actions and 2 to 8 objectives. It was showed that $R2$-IBMOLS is efficient to solve the action plan optimization problem. Unfortunately, $R2$-IBMOLS is time consuming, especially on the large instances, and can generate a high number of solutions, which makes it difficult to use in practice. In this paper, we show that by fixing the size of $R2$-IBMOLS archive and using L-dominance instead of P-dominance to select the solutions to be archived, we obtain high-quality compromises on the one hand, and reduce the runtime and the number of generated nondominated solutions, on the other hand. Also, we propose two other alternative L-dominance-based algorithms to solve the problem with objectives having the same importance.

4. Lorenz dominance

The notion of Lorenz dominance (L-dominance) was first proposed in economics to measure the inequalities in income distributions. Then, in recent years, some L-dominance-based approaches integrating the concept of *equity* were proposed in the multiobjective optimization field (we talk about fair optimization). L-dominance refines P-dominance by selecting only the well located solutions. Moreover, the set of P-optimal solutions obtained in some multiobjective problems can be huge, making it difficult for the decision maker to evaluate the alternative choices. Applying L-dominance instead of P-dominance would be a suitable approach to alleviate this problem. L-dominance introduces a partial order relation among P-nondominated solutions to reduce the size of the output. Fig. 1 shows, for one solution y of a minimization problem, the difference between P-dominated area and L-dominated area (gray color). Since the resulting search space is reduced by the (Pigou-Dalton) *transfer principle* and the Lorenz transformation operated on the objective vectors (Definitions 5 and 6 below), L-dominance allows finding more efficient and well located solutions than P-dominance. This fact is well described in Dugardin et al. (2010).

As reported in Golden and Perny (2010), in order to choose between two nondominated solutions, we have to define a preference relation $\precsim$ on cost vectors, such that $y \precsim z$ means that the corresponding solution to the cost vector y is preferable to the corresponding solution to the cost vector z. Also, to formalize the fact that all the objectives are treated equivalently, we define the following axiom:

Definition 4. For a cost vector $y \in Y$ and any permutation π of {1,... ,m}, $(y_{\pi(1)}, \ldots, y_{\pi(m)}) \sim (y_1, \ldots, y_m)$, where $\sim$ is the indifference relation defined as the symmetric part of $\precsim$.

Furthermore, fair optimization should satisfy the (Pigou-Dalton) *transfer principle* (Sen, 1973) which states that a transfer of any small amount from one cost vector to any other relatively worse-off, while preserving the mean of the costs, could produce more distributed cost vector. As a property of the preference relation $\precsim$, the transfer principle is defined by the following axiom:

Definition 5. For a cost vector $y \in Y$ such that $y_i > y_j$ and for all ϵ such that $0 < \epsilon < y_i - y_j$, $y - \epsilon e_i + \epsilon e_j \prec y$ where e_i and e_j are respectively the ith and the jth unit vectors.

For example, $y = (7, 3, 4)$ and $z = (5, 3, 6)$ are both P-nondominated vectors, but the transfer principle implies that z is preferable to y ($z \precsim y$) because there exists a transfer of size $\epsilon = 2$ to pass from z to y.

In order to identify those vectors that can be compared using the transfer principle, we recall the definition of the *generalized Lorenz vector*, on which L-dominance is based.

Definition 6. For any cost vector $y \in Y$, the generalized Lorenz vector of y is the vector $L(y) = (y_1, y_1 + y_2, \ldots, y_1 + y_2 + \ldots + y_m)$, where $y_1 \geq y_2 \geq \ldots \geq y_m$ represent the components of y sorted in non-increasing order. The jth component of $L(y)$ is $L(y) = \sum_{i=1}^{j} y_i$.

Now, the generalized L-dominance is defined as follow:

Definition 7. The L-dominance relation is defined by:
$\forall y, z \in Y,\ y \prec_L z \Leftrightarrow L(y) \prec_P L(z)$.

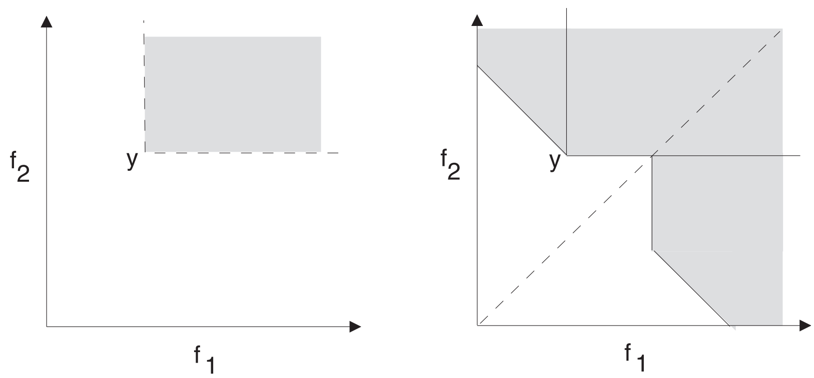

Fig. 1. P-dominated area (left). L-dominated area (right).

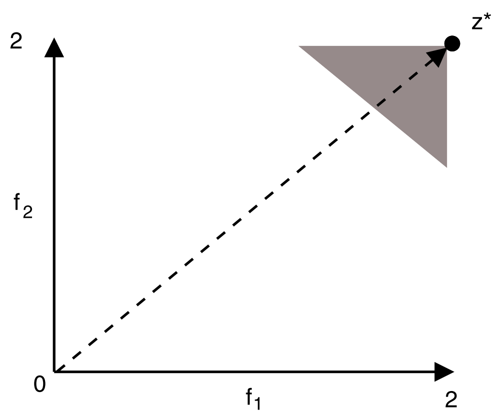

Fig. 2. Target region of the bi-objective space (gray area).

4.1. State of the art on Lorenz dominance based optimization

Lorenz dominance was defined in Kostreva and Ogryczak (1999b) to solve the linear multiple criteria optimization problem, and then was applied to location problems (Kostreva and Ogryczak, 1999a; Ogryczak, 2000a) and portfolio optimization problem (Ogryczak, 2000b). L-dominance was also applied to multiobjective optimization as we review below.

Kostreva et al. (2004) used ordered weighted averaging aggregations to derive equitably efficient solutions to both linear and nonlinear multiobjective problems. Perny et al. (2006) introduced a formal framework to define robustness in combinatorial problems using L-dominance to compare solutions according to multiple scenarios. They also proposed a new approach to find Lorenz-efficient solutions for two robust optimization problems and introduced, within L-dominance, the ordered weighted average as an axiomatically founded measure of robustness. Dugardin et al. (2010) used L-dominance with NSGA-II (Deb et al., 2002) in an algorithm called L-NSGA-II and assessed its performance against other algorithms. Golden and Perny (2010) introduced the notion of *infinite order Lorenz dominance (IOLD)* and showed that, using an ordered weighted average, it is possible to formulate the search of nondominated solutions as a single-objective optimization problem. Our work is based on IOLD, so we provide more description of this notion in the next section. Moghaddam et al. (2011) applied L-dominance to an adapted bi-objective simulated annealing algorithm (to solve a single machine scheduling problem) and showed significant improvements over a Pareto-based multiobjective simulated annealing algorithm. More recently, Galand and Lust (2015) proposed an adaptation of the classic two-phase method to generate Lorenz optimal solutions and evaluated experimentally their method on two bi-objective problems.

4.2. Infinite order Lorenz dominance (IOLD)

When two objective vectors y and z cannot be compared in terms of P-dominance, we perform the corresponding generalized Lorenz vectors $L(y)$ and $L(z)$ and compare instead $L(y)$ to $L(z)$. But, there may be no P-dominance between $L(y)$ and $L(z)$. In this case, the indetermination might be solved by comparing $L^2(y) = L(L(y))$ to $L^2(z) = L(L(z))$. To reduce the incomparability, this process can be iterated to higher levels. Golden and Perny (2010) introduced the kth order Lorenz vector $L^k(y)$ defined as follows:

$$L^k(y) = \begin{cases} y & if \quad k = 0 \\ L(L^{k-1}(y)) & if \quad k \geq 1 \end{cases} \tag{6}$$

and the kth order L-dominance is defined by:

$$\forall y, z \in Y, y \prec_L^k z \Leftrightarrow L^k(y) \prec_P L^k(z) \tag{7}$$

From these two last definitions, the authors defined a strict *infinite order dominance* (strict L^∞-dominance) as follows:

$$\prec_L^\infty = \bigcup_{k \geq 1} \prec_L^k \tag{8}$$

and proposed Algorithm 1 to compute $\prec_L^\infty$ for any two vectors y, z.

Algorithm 1 Strict L^∞-dominance.

```
y' ← y
z' ← z
while not(y' ≺_P z' or z' ≺_P y') do
    y' ← L(y')
    z' ← L(z')
end while
if (y' ≺_P z') then
    y ≺_L^∞ z
end if
if (z' ≺_P y') then
    z ≺_L^∞ y
end if
```

Algorithm 1 tries to select between vectors that are not discriminated by L-dominance. However, nothing proves that it terminates for any pair of vectors. To solve this problem, Perny and Golden formulated the search of nondominated solutions as a single-objective optimization problem and provided a direct mathematical definition of L^∞-dominance (Definition 8), making possible the comparison of any pair of vectors.

Definition 8. The strict L^{∞}-dominance has a strict numerical representation using the following ordered weighted average:

$$\mathcal{W}(y) = \sum_{k=1}^{m} \sin\left(\frac{(m+1-k)\pi}{2m+1}\right) y_k \quad (9)$$

This representation is given by the following property: $\forall y, z \in Y, y \prec_L^{\infty} z \Longleftrightarrow \mathcal{W}(y) < \mathcal{W}(z)$ (see the proof of this property and Eq. (9) in Golden and Perny (2010)).

Now, to compare any pair of vectors y and z according to a strict L^{∞}-dominance, it is not necessary to run Algorithm 1, we have just to compute and compare $\mathcal{W}(y)$ and $\mathcal{W}(z)$. When $\mathcal{W}(y) \neq \mathcal{W}(z)$, the vector having the smallest score for $\mathcal{W}$ strictly L^{∞}-dominates the other. Hence, Algorithm 1 would stop after a sufficiently large number of iterations. And when $\mathcal{W}(y) = \mathcal{W}(z)$, there is no strict dominance at any order of L-dominance application, and Algorithm 1 would never terminates in this case.

Eq. (9) presents two main advantages. First, it is easy to integrate $\mathcal{W}$ into an evolutionary algorithm. Indeed, with few parameters, it makes it possible to select individuals at a high level of comparison, and detect dominance relation, even when the difference is minimal between individuals. Second, using $\mathcal{W}$ in the selection process of an evolutionary algorithm significantly reduces the runtime of the algorithm, because it is no more necessary to check the dominance between the individuals of the population. Dominance verification is directly integrated in the definition of the equation.

5. Proposed approaches

In this section, we use the strict L^{∞}-dominance within three new algorithms dedicated to solve the action plan optimization problem. Nevertheless, ideas presented here could be easily adapted to solve other problems.

In the following, P denotes the current population of solutions and it is assumed that objective values of all solutions are normalized. To achieve this, the minimum m_j and maximum M_j values of each objective function f_j in the population P are computed first:

$$\begin{cases} m_j = min_{x \in P}(f_j(x)) \\ M_j = max_{x \in P}(f_j(x)) \end{cases} \quad (10)$$

Then each objective function j of each individual x of P is normalized as follows:

$$NF_j(x) = \frac{f_j(x) - m_j}{M_j - m_j} \quad (11)$$

where $NF_j(x)$ is the normalized jth objective function of the individual x.

Note that extreme values are updated after each local search step and only when a new solution is introduced in the current population. Each time the maximum value M_j or the minimum value m_j is changed for some objective function j, the normalized objective values of each solution $x \in P$ is updated.

5.1. The IOLD-R2-IBMOLS algorithm

The $IOLD-R2$-IBMOLS algorithm (Algorithm 2) combines the performance of $R2$-IBMOLS (Chabane et al., 2017), which offers two means to integrate decision maker preferences (reference point and set of weight vectors), with the infinite order Lorenz dominance that allows a comparison between nondominated solutions at a high level of L-dominance (see Section 4.2).

The algorithm begins by selecting nondominated solutions of a given initial population P. For this, we use the *infiniteOrderLorenzNonDominatedSolutions* function (having two parameters P and T). According to the values of $\mathcal{W}$ for all solutions

Algorithm 2 *IOLD – R2*-IBMOLS algorithm.

```
Input: P (initial population of size N), T (size of the archive)
Output: E (approximation set (|E| ≤ T))
E ⟵ infiniteOrderLorenzNonDominatedSolutions(P, T)
/* Compute fitness values of individual x in P */
for all x ∈ P do
  fit(x) = R2(P, Λ, z*) − R2(P\{x}, Λ, z*)
end for
/* Local search step */
for all x ∈ P do
  for all j ∈ {1, ., m} do
    updateMinMax(j) /* Update minimal m_j and maximal M_j (for
    objective functions normalization) */
  end for
  repeat
    x* ⟵ one unexplored neighbor of x
    P ⟵ P ∪ x*
    fit(x*) = R2(P, Λ, z*) − R2(P\{x*}, Λ, z*) /* Compute x* fit-
    ness */
    Update all z ∈ P fitness values
    w ⟵ the worst individual in P
    remove w from P
    Update all z ∈ P fitness values
  until all neighbors are explored or w ≠ x*
end for
E ⟵ infiniteOrderLorenzNonDominatedSolutions(E ∪ P, T)
if E does not change then
  return E
else
  perform another local search step
end if
```

of P, this function selects, at most, the T first L-nondominated solutions of P (see Algorithm 3). To select the closest solutions to

Algorithm 3 infiniteOrderLorenzNonDominatedSolutions(P, T) procedure.

```
Input: P (set of solutions); T(number of L-nondominated solutions to
be selected)
Output: S (set of L-nondominated solutions (|S| ≤ T))
for all x ∈ P do
  Compute W(x)
end for
Sort solutions of P in increasing order of W values /* for maximization
problem, sort in non-increasing order */
S' ⟵ T first solutions in P
S ⟵ L-nondominated solutions in S'
```

the reference point, $\mathcal{W}$ is used with the same weights as those used to fix the reference point (these weights are given by the decision maker). The parameter T allows the decision maker to control the number of solutions stored in the archive and the number of solutions returned by the algorithm. In this work, we use the same value of T for both, but one could use two different values. After that, the fitness of each solution x in P is evaluated as follows.

$$R2(x, \Lambda, z^*) = R2(P, \Lambda, z^*) - R2(P \backslash \{x\}, \Lambda, z^*) \quad (12)$$

Then, the $IOLD-R2$-IBMOLS algorithm applies a local search step to improve each solution x in P. A neighbor is accepted if its $R2$ indicator value is better than the worst solution in P. The neighborhood generation stops when the entire neighborhood of a considered solution is explored or once an improving solution is found (first neighboring solution that improves the quality of

P with respect to the $R2$ indicator). The neighborhood is not explored entirely for seeking the best neighbor for two main reasons: i) it often enables to speed up the convergence of the population, since most of the time only a small part of the neighborhood is generated. ii) Contrary to the selection of the best neighbor which leads to more deterministic local search steps, the selection of a first improving move allows us to reach different local optima (in the sense of multiobjective optimization) from a single initial solution. The entire local search is terminated when the archive E of L-nondominated solutions has not received any new solution during a complete local search step. We note that the extreme objective values within the population are computed after the initialization process as well as after each local search step. To compute the $R2$ indicator value of a solution x, the normalized values of objective functions $NF_j(x)$ are employed. Moreover, all fitness values of members of current population P are updated after every change of P (when a new neighbor is added to P and when an individual is deleted from P).

The main difference between $IOLD-R2$-IBMOLS and $R2$-IBMOLS is the introduction of the *infiniteOrderLorenzNonDominatedSolutions* function (Algorithm 3). Indeed, on the one hand, this function allows us to reduce the number of nondominated solutions to be presented to the decision maker by introducing the parameter T, ensuring that the selected solutions are the most preferable (with the use of $\mathcal{W}$). On the other hand, fixing the size of the archive allows reducing significantly the algorithm runtime, which is shown in the next section.

In the iterated version of $IOLD-R2$-IBMOLS (Algorithm 4), a current approximation set AS is maintained and updated. After each local search, a new initial population is created for the next $IOLD-R2$-IBMOLS execution, using the *initWalk* function.

Algorithm 4 Iterated $IOLD-R2$-IBMOLS algorithm.

```
Input: N (population size) ; T (size of the archive)
Output: AS (approximation set)
AS ⟵ ∅
while stopping condition not achieved do
    P ⟵ initWalk(AS, N)
    E ⟵ IOLD − R2-IBMOLS(P, T) /* Local search step */
    AS ⟵ infiniteOrderLorenzNonDominatedSolutions(AS ∪ E, T)
end while
Return AS
```

Even if the initial population is entirely created randomly for the first iteration, when we iterate the local search process, the *initWalk* function generates a new population P for the next iteration using information about good solutions obtained during the previous iterations. Indeed, the *initWalk* function applies random mutations on N randomly selected solutions of AS (each solution of AS can only be selected at most once). To each selected solution, the mutation is applied with a probability of $1/n$ (where n is the number of actions) and the mutated solution is added to the population if it is not present in the population and if it additionally verifies the budget constraint β and the objective thresholds. When $|AS| < N$, all solutions of AS are selected and the missing individuals of P are filled with new random solutions.

5.2. The IOLD-BMOLS algorithm

As mentioned in Section 4.2, we can use Eq. (9) to compare two solutions x_1 and x_2 at high level of L-dominance and the solution having the smallest score (the highest in the maximization case) for $\mathcal{W}$ is preferable to the other one. So, in the *IOLD*-BMOLS algorithm (Algorithm 5), we use $\mathcal{W}$ to compute the fitness of each solution x of the current population P, which is also applied in the selection process.

Algorithm 5 IOLD-BMOLS algorithm.

```
Input: P (initial population of size N); T(size of the archive)
Output: E (Pareto approximation set (|E| ≤ T))
E ⟵ infiniteOrderLorenzNonDominatedSolutions(P, T)
/* Calculate fitness values of individual x in P */
for all x ∈ P do
    fit(x) = W(x)
end for
/* Local search step */
for all x ∈ P do
    for all j ∈ {1, ., m} do
        updateMinMax(j) /* Update minimal m_j and maximal M_j (for
        objective functions normalization)*/
    end for
    repeat
        x* ⟵ one unexplored neighbor of x
        P ⟵ P ∪ x*
        fit(x*) = W(x*)/* Compute x* fitness */
        w ⟵ the worst individual in P /* Individual with the smallest
        value of W */
        remove w from P
    until all neighbors are explored or w ≠ x*
end for
E ⟵ infiniteOrderLorenzNonDominatedSolutions(E ∪ P, T)
if E does not change then
    return E
else
    perform another local search step
end if
```

Like $IOLD-R2$-IBMOLS, *IOLD*-BMOLS uses the *infiniteOrdredLorenzNonDominatedSolutions* function (Algorithm 3) to select all nondominated solutions of a given initial population P. Then it computes the fitness of each solution x in P with the formula of $\mathcal{W}$ given in Eq. (9). A local search step is applied in *IOLD*-BMOLS, like in $IOLD-R2$-IBMOLS, except that the worst individual from the population P is chosen for deletion. In $IOLD-R2$-IBMOLS, the worst individual is selected in relation to its value of the $R2$ indicator, but in *IOLD*-BMOLS, the $R2$ indicator is replaced with $\mathcal{W}$, so the worst individual is selected in relation to its $\mathcal{W}$ score.

IOLD-BMOLS has also an iterated version (Algorithm 6) with a similar structure to the iterated version of $IOLD-R2$-IBMOLS. We just replace $IOLD-R2$-IBMOLS by *IOLD*-BMOLS for the local search step.

Algorithm 6 Iterated *IOLD*-BMOLS algorithm.

```
Input: N (population size) ; T (size of the archive)
Output: AS (approximation set)
AS ⟵ ∅
while stopping condition not achieved do
    P ⟵ initWalk(AS, N)
    E ⟵ IOLD-BMOLS(P, T) /* Local search step */
    AS ⟵ infiniteOrderLorenzNonDominatedSolutions(AS ∪ E, T)
end while
Return AS
```

5.3. The IOLD-EA algorithm

We now propose to use $\mathcal{W}$ in the selection process of an *IOLD*-based evolutionary algorithm called: *IOLD*-EA (Algorithm 7). The formula $\mathcal{W}$ is used to compute the fitness of the solutions in the population P and to select the members of the next generation.

Table 1
Summary of the studied algorithms.

Method	Description
$IOLD-R2$-**IBMOLS**	IOLD-R2-IBMOLS combines R2-IBMOLS and *infiniteOrderLorenzNonDominatedSolutions* function (Algorithm 3). One iteration of $IOLD-R2$-IBMOLS is composed of: - one iteration of R2-IBMOLS. - selection of the T first L-nondominated solutions generated by the iteration of R2-IBMOLS. This selection is mad according to the value of each solution for the metric $\mathcal{W}$. In the iterated version of IOLD-R2-IBMOLS only the T first L-nondominated solutions are archived.
IOLD-**BMOLS**	In the *IOLD*-BMOLS algorithm, the $R2$ indicator of $R2$-IBMOLS is replaced with the formula $\mathcal{W}$ to compute the fitness of each solution x of the current population P. $\mathcal{W}$ is also used in the selection process. In the iterated version of *IOLD*-BMOLS, only the T first L-nondominated solutions are archived.
IOLD-**EA**	In *IOLD*-EA, the formula $\mathcal{W}$ is used to compute the fitness of the solutions in the population P and to select the members of the next generation of *IOLD*-EA algorithm. *IOLD*-EA has three parameters: an initial population P of size N, a given number of generations to perform *nbGen*, and a number of solutions to return to the decision maker T. The T first L-nondominated solutions to return to the decision maker are selected also according to their values for $\mathcal{W}$.

Algorithm 7 IOLD-EA algorithm.

```
Input: P (initial population of size N), nbGen (number of generations)
Output: E (Pareto approximation set)
t = 1
P_t = P
while t ≤ nbGen do
    Q_t = child(P_t)
    R_t = P_t ∪ Q_t
    for all x ∈ R_t do
        fit(x) = W(x)
    end for
    t = t + 1
    P_t ⟵ infiniteOrdredLorenzSelection(R_t, N)
end while
E ⟵ infiniteOrdredLorenzSelection(P_t, T)
```

Indeed, contrary to evolutionary algorithms based on calculation of P-dominance in the selection process, like NSGA-II (Deb et al., 2002) and NSGA-III (Deb and Jain, 2014), which could be runtime expensive (especially on large size problems), the use of $\mathcal{W}$ in the selection process allows us to reduce significantly the runtime of the algorithm (since we do not need to check the dominance between the individuals of the population).

IOLD-EA has three parameters: an initial population P of size N, a given number of generations to perform *nbGen*, and a number of solutions to return to the decision maker T. Then, for a given tth generation and a parent population $\mathcal{P}_t$, the function $child(\mathcal{P}_t)$ creates an offspring population $\mathcal{Q}_t$ of N individuals, by recombination and mutation of $\mathcal{P}_t$. The formula $\mathcal{W}$ is used to compute the fitness of each individual of the combined parents and offspring population $\mathcal{R}_t = \mathcal{P}_t \cup \mathcal{Q}_t$ (of size $2N$). To construct the new population $\mathcal{P}_{t+1}$, the *infiniteOrdredLorenzSelection* function (Algorithm 3) selects the N best individuals of $\mathcal{R}_t$ according to their computed scores for $\mathcal{W}$. In the end, the *infiniteOrderLorenzNonDominatedSolutions* function returns to the decision maker the T best solutions of the last population $\mathcal{P}_{nbGen}$. A detailed description of *IOLD*-EA is outlined in Algorithm 7.

A summary of the proposed algorithms is given in Table 1.

6. Experimental setup

In this section, we present a set of experiments allowing a comparison of the results obtained with the proposed algorithms ($IOLD-R2$-IBMOLS, *IOLD*-BMOLS and *IOLD*-EA) with four reference methods: L-NSGA-II (Dugardin et al., 2010), NSGA-III (Deb and Jain, 2014), $R2$-IBMOLS (Chabane et al., 2017) and MOEA/D (Zhang and Li, 2007). With these experiments, we aim to show that including L-dominance within a multiobjective algorithm allows us to obtain high quality solutions while guaranteeing fairness between objectives. Moreover, the simplicity of the *IOLD* formulation (Eq. (9)) makes it easy to use for the fitness assessment of multiobjective local search algorithms such as IBMOLS as well as in the selection process of an evolutionary multiobjective algorithm.

L-NSGA-II (Dugardin et al., 2010) is a L-dominance based version of the well-known NSGA-II algorithm (Deb et al., 2002) algorithm. The main difference between them is the use of L-dominance in the fast nondominated sorting procedure of NSGA-II. NSGA-III (or many-objective NSGA-II) (Deb and Jain, 2014) remains similar to the original NSGA-II algorithm with significant changes in its selection mechanism. The maintenance of diversity among population members in NSGA-III is aided by supplying and adaptively updating a number of well-spread reference points, whereas NSGA-II uses a crowding distance operator to select the solutions of the last front that maximizes its diversity. In NSGA-III, the crowding operator is replaced by a reference point based approach (for more details, see Deb and Jain, 2014). Finally, MOEA/D (Zhang and Li, 2007) decomposes a multiobjective optimization problem into several scalar optimization subproblems, which are simultaneously optimized by using information from their neighboring subproblems. In Ishibuchi et al. (2015), it is shown that MOEA/D with the weighted sum approach also works well on the many-objective knapsack problem.

6.1. Instances generation

Based on the action plan optimization model given in Section 2 and a study of ten real action plans, we have generated several partially structured instances[2] with different number of actions, $n \in \{50, 100, 150, 250, 500\}$ and different number of objectives to optimize, $m = 2, 3, 4, 5, 6, 8$. In our experiments, we limited the number of the objectives to eight because in practical cases we have rarely more. To design instances that are similar to the real action plans, for each objective function, an action has a chance of 50% to be neutral, 40% to have a positive impact and 10% to have a negative impact. Moreover, the cost of 40% of the actions is set to zero. The non-null action values are uniformly taken from the interval [0,100] (positively or negatively). The non-null action costs are uniformly taken in the interval $[-10^4, 10^4]$.

6.2. Experimental protocol

Using the instances proposed above, we have tested $IOLD-R2$-IBMOLS, *IOLD*-BMOLS, *IOLD*-EA, $R2$-IBMOLS, L-NSGA-II, NSGA-III and MOEA/D with the following parameters: for L-NSGA-II, NSGA-III, *IOLD*-EA and MOEA/D, we have used a population of size of 100, a mutation probability of $1/n$ (where n is the number of the actions). For $IOLD-R2$-IBMOLS, *IOLD*-BMOLS and $R2$-IBMOLS we have used the iterative version with a fixed popu-

[2] These instances are available at: http://www.info.univ-angers.fr/~hao/gepiplanning/R2-IBMOLS.zip

lation of size of 10. The choice of a population size of 10 solutions in our experiments is based on the recommendation given in Basseur et al. (2012) that IBMOLS performs well especially with a small population (no more than 15 individuals). For each algorithm, the initial population is generated randomly while satisfying the following constraints: (i) the costs of the individuals do not exceed the budget β; (ii) each individual x should improve all the objectives ($f_j(x) > 0\ \forall j \in \{1, \ldots, m\}$). Algorithm 8 shows the initial population generation procedure.

Algorithm 8 Initial population generation.

Input: N (population size), β (budget)
Output: P (initial population)
$P = \varnothing$
while $|P| < N$ **do**
  x = randomSolution()
  while cost$(x) > \beta$ **do**
    $x = x$ less one random action with positive cost
  end while
  for each objective function f_j **do**
    while $f_j(x) \leq 0$ **do**
      $x = x$ less one random action with negative value for f_j
    end while
  end for
  $P = P \cup \{x\}$
end while
Return P

Moreover, the following selection strategy is adopted: one random neighbor of each individual of the current population is selected to be a member of the child population in L-NSGA-II, NSGA-II and *IOLD*-EA or to integrate the current population of *IOLD*-BMOLS, *IOLD* – *R*2-IBMOLS and *R*2-IBMOLS. The neighborhood generation remains unchanged: the *i*th neighbor of the solution $x = (a_1, a_2, \ldots, a_n)$ is obtained by flipping the value of a_i and only the neighbors verifying the constraint β and the objective thresholds are considered as candidate (when the cost of the neighbor is greater than β, another neighbor is generated). For all instances, the budget constraint β is fixed to one million and the thresholds are fixed to 1 ($c_j \geq 1\ \forall j \in \{1, \ldots, m\}$).

To compute the *R*2 indicator value of each solution x of *IOLD* – *R*2-IBMOLS and *R*2-IBMOLS, Eq. (12) is used with the reference point $z^* = (2, 2, \ldots, 2)$ and 100 weight vectors ($|\Lambda| = 100$), uniformly distributed in the objective space. The same weight vectors are used in the MOEA/D algorithm with the weighted sum approach.

To generate these vectors, we have used the hypervolume-based algorithm proposed in Phan and Suzuki (2013). This algorithm uses the hypervolume indicator to produce weight vectors so that they uniformly disperse and maximize their hypervolume in the objective space. This method is interesting because it does not depend on the dimension of the objective space and works in the same way for both low-dimensional and high-dimensional spaces. However, this method can be time consuming if the weight vectors are generated at each iteration. In our experiments, the weight vectors are generated once and remain the same throughout each experiment. Also, in Phan and Suzuki (2013), the first weight vector is generated randomly, in our experiments it is fixed according the reference point. The reference point z^* is also used within NSGA-III. We note that the reference points of NSGA-III could be predefined in a structured manner or supplied by a decision maker and their number could be large. In ours experiments, we are focusing on the region of objective space given by z^* (Fig. 2). We have finally fixed the parameter T of *IOLD*-BMOLS, *IOLD* – *R*2-IBMOLS and the neighborhood size of MOEA/D to 10.

For the quality assessment, we have performed 30 runs of each method on each instance. The stopping condition for each run corresponds to 200*n*m steps of local search for *IOLD*-BMOLS, *IOLD* – *R*2-IBMOLS and *R*2-IBMOLS and 200*n*m generations for *IOLD*-EA, L-NSGA-II and NSGA-III (where n is the number of actions and m is the number of objectives). The experiments were performed on an Intel core i5-2400 CPU machine with 2 x 3.10Ghz frequency and 16Gb of RAM.

6.3. Computational results

In this section, we present the experimental results obtained on the simulated data with the experimental protocol described previously. Figs. 3 and 4 show approximation sets obtained with the proposed algorithms (*IOLD*-BMOLS, *IOLD* – *R*2-IBMOLS and *IOLD*-EA) over 30 runs compared to NSGA-III, L-NSGA-II, *R*2-IBMOLS and MOEA/D on three representative instances: an instance with 2 objectives and 50 or 500 actions (Fig. 3) and another instance with 4 objectives and 500 actions (Fig. 4). Each plot of Fig. 4 shows the obtained results for two objectives.

These figures show that the solutions obtained by *IOLD*-BMOLS, *IOLD* – *R*2-IBMOLS and *IOLD*-EA are nondominated by those obtained by NSGA-III, L-NSGA-II, *R*2-IBMOLS or MOEA/D. Moreover, according to the preferred direction (gray arrow) given by the reference point $z^* = (2, ., 2)$, the solutions obtained by *IOLD*-BMOLS, *IOLD* – *R*2-IBMOLS and *IOLD*-EA are better located in the bi-objective space of Fig. 3 and, also better located for three out of four objectives of Fig. 4. Indeed, on the 4th objective (f_4 in Fig. 4), *R*2-IBMOLS obtained better solutions than *IOLD*-BMOLS, *IOLD* – *R*2-IBMOLS and *IOLD*-EA. Also, we note that the blue surface of Fig. 4 is wide because *R*2-IBMOLS generates more nondominated solutions than the other algorithms (the archive of *R*2-IBMOLS was not limited in our experiments).

As our performance metric, we use the Euclidean distance from reference point z^* applied to the obtained approximation sets. Indeed, in our experiments, we target the central region of the objective space. This targeting is given by the equality of the component of the reference point $z^* = (2, 2, \ldots, 2)$ of *R*2-IBMOLS, *IOLD* – *R*2-IBMOLS and NSGA-III and by the L-dominance relation used in *IOLD*-EA, *IOLD*-BMOLS. So, the best approximations are formed by solutions that are the nearest to the reference point. To measure the distance of the whole approximation sets from the reference point, we have computed the minimal, the median and the maximal distances from the reference point to the approximation sets obtained by each algorithm.

Tables 2–4 report the comparison between *IOLD*-BMOLS, *IOLD* – *R*2-IBMOLS, *IOLD*-EA, NSGA-III, L-NSGA-II, *R*2-IBMOLS and MOEA/D in terms of mean values obtained for minimal, median and maximal Euclidean distances over 30 runs, using the set of 30 instances of different sizes (approximation set with a smaller distance value is better). The first column shows the instance name, indicating its main characteristics: m and n respectively correspond to the number of objectives and the number of actions considered. Each cell of the table contains the mean value for the corresponding distance over 30 runs. The values in bold style mean that the corresponding algorithm is better in average than the other algorithms for the considered instance and corresponding distance. The values in italic style indicate that the corresponding algorithm is better than the algorithm corresponding to the values in normal style, but it is worse than the algorithm corresponding to the values in bold style, for the corresponding instance and distance (i.e., the 2nd best algorithm).

Using the non-parametric *Mann–Whitney* test and the *Bonferroni* correction to adjust the individual significance levels, we perform a pair-wise comparison of the algorithms for the obtained minimal, median and maximal distances. We obtain the *p–value*

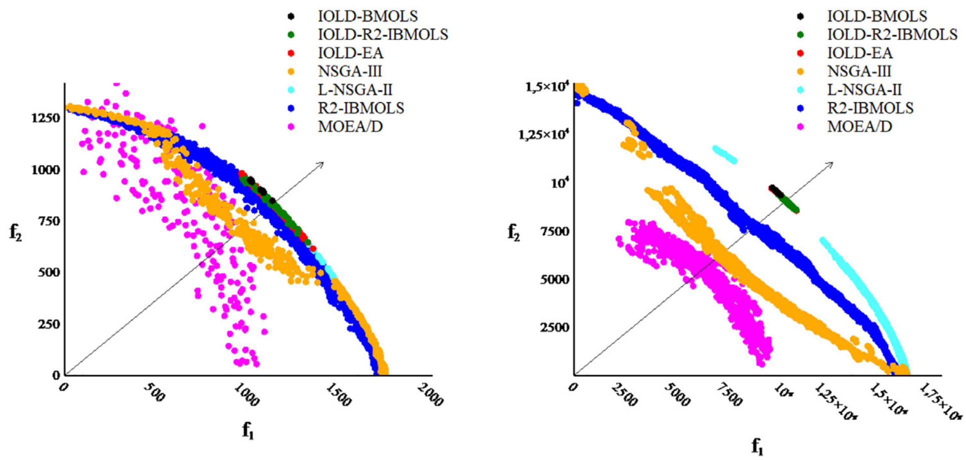

Fig. 3. Obtained solutions by the tested algorithms for the instances 2–50 (left) and 2–500 (right).

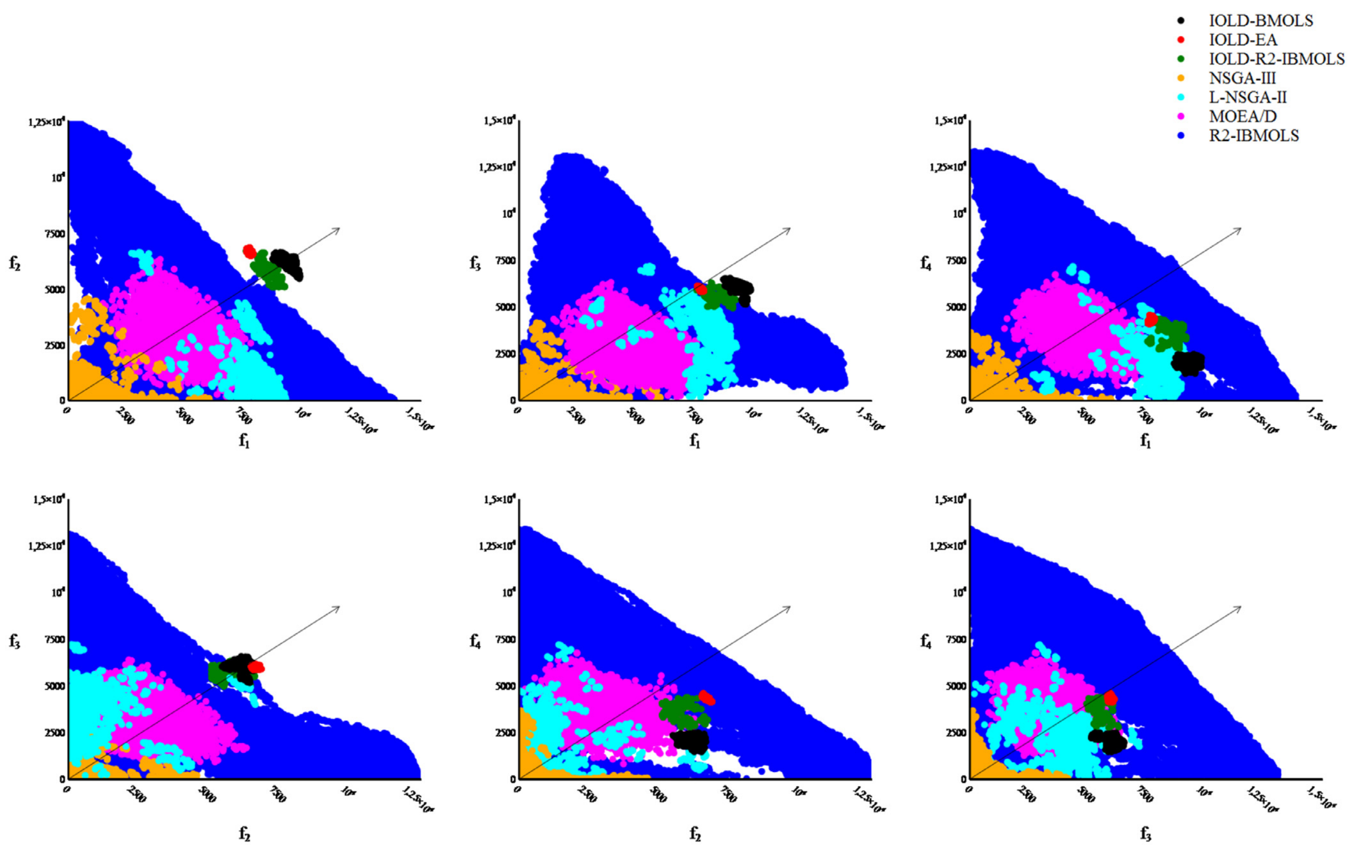

Fig. 4. Obtained solutions by the tested algorithms for the instance 4–500. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

corresponding to the lowest significance level for which the null-hypothesis is rejected. In our experiments, we say that algorithm A_1 outperforms algorithm A_2 if the *Mann–Whitney* test provides a confidence level greater than 95% ($p\text{–}value \leq 0.05$). Table 5 reports the obtained results for each pair of algorithms. Cells containing " yes" means that the algorithm in the corresponding line is statistically better than the algorithm in the correspond column. Cells containing "no" means that the algorithm in the corresponding line is not statistically better than the algorithm in the correspond column.

From Tables 2–5, we can conclude that *IOLD*-BMOLS, *IOLD*-EA and *IOLD* – *R*2-IBMOLS are more efficient than NSGA-III, L-NSGA-II, *R*2-IBMOLS and MOEA/D. Tables 2–4 show that, in general, the solutions obtained by *IOLD*-BMOLS, *IOLD* – *R*2-IBMOLS, *IOLD*-EA are closer to the specified region. For example, if we consider the minimal distance to the reference point (Table 2), *R*2-IBMOLS obtains best scores only for 4 instances (″2_50″,″2_100″,″2_250″ and ″4_100″) and L-NSGA-II obtains best scores just for three instances (″2_150″,″3_250″ and ″6_50″). NSGA-III obtains no best score at all and it is less efficient than MOEA/D, which is consistent with the finding in Ishibuchi et al. (2017) about the weak convergence ability of this algorithm on many objective problems. Considering our proposed algorithms, Table 2 also shows that *IOLD*-EA is more efficient than *IOLD*-BMOLS and *IOLD* – *R*2-IBMOLS. This conclusion

Table 2
Mean of the minimal Euclidean distances from the approximation sets to the reference point z^*.

Instance	*IOLD*-**BMOLS**	*IOLD* − *R*2-**IBMOLS**	*IOLD*-**EA**	NSGA-III	L-NSGA-II	*R*2-**IBMOLS**	MOEA/D
2_50	2.096	2.050	*1.999*	2.080	2.049	**1.912**	2.072
2_100	2.111	2.078	2.090	2.109	*2.018*	**2.003**	2.053
2_150	2.118	2.080	2.087	2.140	**1.911**	*1.937*	2.134
2_250	2.120	2.104	2.097	2.142	*2.034*	**1.991**	2.141
2_500	2.119	2.097	**2.009**	2.155	2.097	*2.032*	2.173
3_50	2.531	**2.394**	*2.469*	2.729	2.540	2.567	2.631
3_100	**2.389**	2.459	*2.413*	2.774	2.493	2.521	2.714
3_150	2.552	**2.377**	*2.449*	2.734	2.486	2.579	2.682
3_250	2.476	*2.472*	2.516	2.825	**2.462**	2.697	2.729
3_500	2.481	2.482	**2.400**	2.672	*2.429*	2.656	2.652
4_50	2.864	*2.849*	**2.669**	3.351	2.981	2.992	3.113
4_100	2.910	2.884	*2.784*	3.452	3.101	**2.768**	2.924
4_150	2.805	2.895	**2.737**	3.353	*2.763*	3.035	3.118
4_250	3.025	**2.664**	*2.713*	3.339	2.969	3.046	3.101
4_500	3.061	*2.926*	**2.810**	3.480	2.994	3.129	3.242
5_50	**3.117**	3.208	*3.194*	3.814	3.237	3.566	3.613
5_100	3.214	*3.158*	**3.033**	3.919	3.416	3.655	3.798
5_150	3.301	*3.106*	**3.082**	3.728	3.246	3.640	3.673
5_250	3.324	*3.171*	**2.930**	3.915	3.533	3.709	3.812
5_500	3.217	*3.149*	**3.144**	3.625	3.243	3.742	3.462
6_50	*3.467*	3.598	3.496	4.220	**3.378**	3.699	3.727
6_100	3.699	*3.582*	**3.509**	4.091	3.610	4.035	4.068
6_150	3.631	*3.569*	**3.358**	4.206	3.767	3.931	4.057
6_250	*3.530*	3.591	**3.328**	4.260	3.535	3.951	4.093
6_500	3.626	*3.517*	**3.308**	3.807	3.649	3.959	3.713
8_50	4.182	*4.127*	**4.005**	5.032	4.181	4.263	4.367
8_100	**4.076**	4.175	*4.079*	5.058	4.121	4.433	4.628
8_150	4.085	**4.024**	*4.056*	4.594	4.139	4.498	4.529
8_250	4.109	*4.079*	**3.910**	4.381	4.347	4.582	4.361
8_500	4.170	*4.051*	**3.986**	4.445	4.272	4.573	4.393

Table 3
Mean of the median Euclidean distances from the approximation sets to the reference point z^*.

Instance	*IOLD*-**BMOLS**	*IOLD* − *R*2-**IBMOLS**	*IOLD*-**EA**	NSGA-III	L-NSGA-II	*R*2-**IBMOLS**	MOEA/D
2_50	2.201	2.070	*2.087*	2.105	2.088	**2.019**	2.097
2_100	2.121	2.105	2.105	2.157	**2.065**	*2.071*	2.132
2_150	2.152	2.121	2.112	2.170	**2.054**	*2.060*	2.164
2_250	2.130	2.153	2.114	2.192	**2.075**	*2.076*	2.176
2_500	2.129	2.139	**2.095**	2.229	2.112	*2.102*	2.188
3_50	2.588	**2.444**	*2.565*	2.975	2.600	2.719	2.823
3_100	**2.452**	2.500	*2.475*	2.916	2.664	2.521	2.871
3_150	2.631	**2.437**	*2.546*	2.917	2.616	2.716	2.892
3_250	**2.520**	*2.527*	2.642	3.043	2.713	2.815	2.995
3_500	*2.527*	2.592	**2.478**	2.787	2.589	2.800	2.754
4_50	**2.932**	*2.984*	2.993	3.519	3.208	3.311	3.403
4_100	**2.984**	3.013	*2.997*	3.642	3.325	3.090	3.448
4_150	**2.913**	3.047	*2.989*	3.596	3.157	3.326	3.501
4_250	3.075	**2.965**	*2.970*	3.721	3.180	3.337	3.599
4_500	3.118	*3.056*	**3.005**	3.793	3.166	3.376	3.666
5_50	**3.384**	*3.391*	3.433	4.001	3.622	3.819	3.868
5_100	3.376	*3.333*	**3.297**	4.100	3.654	3.898	3.893
5_150	3.503	*3.364*	**3.294**	4.116	3.529	3.882	4.002
5_250	3.504	*3.342*	**3.242**	4.091	3.712	4.007	4.046
5_500	**3.312**	*3.338*	3.395	4.089	3.615	4.001	4.075
6_50	**3.554**	3.724	*3.718*	4.394	3.865	4.042	4.111
6_100	*3.757*	3.814	**3.704**	4.529	4.040	4.351	4.407
6_150	*3.703*	3.795	**3.669**	4.479	4.039	4.290	4.378
6_250	*3.688*	3.727	**3.602**	4.444	3.944	4.266	4.353
6_500	3.742	*3.668*	**3.542**	4.650	3.901	4.411	4.451
8_50	4.313	*4.251*	**4.164**	5.140	4.549	5.408	5.030
8_100	*4.322*	4.358	**4.281**	5.218	4.687	5.623	5.127
8_150	**4.207**	*4.214*	4.269	4.900	4.647	5.580	4.663
8_250	*4.212*	4.257	**4.137**	4.718	4.689	5.599	4.697
8_500	*4.273*	4.294	**4.170**	4.728	4.616	5.698	4.701

is confirmed by the *Mann-Whitney* test in Table 5. Indeed, *IOLD*-EA outperforms all the considered algorithms and *IOLD* − *R*2-IBMOLS outperforms all the other algorithms except *IOLD*-EA.

In the practical case, it is desirable to present a reduced number of high-quality solutions to the decision maker. Indeed, one solution could have several tens of actions and the decision maker should be able to choose the most adequate action plan easily. Table 6 reports the average number of nondominated solutions obtained with each algorithm over 30 runs and the corresponding average runtime (respectively *nbSol* and *t*). We note that the number of obtained solutions is highly variable for the algorithms. The parameter *T* of the *IOLD*-BMOLS, *IOLD* − *R*2-IBMOLS and *IOLD*-EA

Table 4
Mean of the maximal Euclidean distances from the approximation sets to the reference point z^*.

Instance	*IOLD*-**BMOLS**	*IOLD* − *R*2-**IBMOLS**	*IOLD*-**EA**	NSGA-III	L-NSGA-II	*R*2-**IBMOLS**	MOEA/D
2_50	2.249	2.241	*2.238*	2.250	**2.236**	2.247	2.287
2_100	**2.236**	**2.236**	2.245	*2.238*	2.243	2.247	2.293
2_150	**2.236**	2.338	**2.236**	2.238	*2.236*	2.244	2.278
2_250	**2.236**	2.269	**2.236**	*2.237*	2.243	2.251	2.261
2_500	**2.236**	*2.238*	*2.238*	2.259	**2.236**	2.241	2.273
3_50	*2.749*	**2.703**	2.812	3.099	2.884	2.928	3.042
3_100	*2.655*	**2.654**	2.690	3.201	2.962	2.997	3.087
3_150	*2.779*	**2.728**	2.860	3.223	2.966	3.003	3.113
3_250	**2.627**	*2.840*	2.856	3.377	2.912	3.013	3.204
3_500	*2.785*	2.994	**2.743**	3.196	2.968	3.007	3.184
4_50	3.275	**3.212**	*3.263*	3.711	3.556	3.578	3.601
4_100	**3.117**	3.246	*3.236*	3.766	3.596	3.439	3.677
4_150	**3.134**	3.284	*3.221*	3.805	3.472	3.595	3.756
4_250	**3.194**	*3.227*	3.241	3.816	3.482	3.589	3.782
4_500	*3.248*	**3.246**	3.282	3.922	3.490	3.600	3.837
5_50	3.803	**3.599**	*3.719*	4.161	4.041	4.100	4.127
5_100	*3.566*	**3.564**	3.648	4.244	4.005	4.115	4.165
5_150	3.628	*3.566*	**3.555**	4.258	3.923	4.124	4.193
5_250	3.689	*3.632*	**3.525**	4.211	4.016	4.245	4.202
5_500	**3.405**	*3.569*	3.683	4.218	3.967	4.219	4.210
6_50	3.971	**3.947**	*3.956*	4.583	4.312	4.483	4.497
6_100	*3.935*	4.095	**3.912**	4.667	4.484	4.580	4.612
6_150	**3.860**	4.311	*3.927*	4.625	4.413	4.563	4.603
6_250	**3.847**	3.950	*3.866*	4.585	4.412	4.554	4.551
6_500	3.920	*3.907*	**3.840**	4.747	4.200	4.665	4.691
8_50	*4.551*	4.568	**4.445**	5.326	5.035	5.656	5.111
8_100	*4.526*	4.655	**4.510**	5.370	5.131	5.656	5.167
8_150	**4.440**	*4.454*	4.531	5.278	5.096	5.656	5.213
8_250	**4.387**	4.432	*4.418*	5.290	4.996	5.656	5.266
8_500	4.432	*4.421*	**4.415**	5.131	4.941	5.743	5.107

Table 5
Statistical comparison of the algorithms with the *Mann–Whitney* test (the results presented in the table are the same for the three distances).

	IOLD-**BMOLS**	*IOLD*-**EA**	*IOLD* − *R*2-**IBMOLS**	L-NSGA-II	NSGA-III	*R*2-**IBMOLS**	MOEA/D
IOLD-**BMOLS**		no	no	yes	yes	yes	yes
IOLD-**EA**	yes		yes	yes	yes	yes	yes
IOLD − *R*2-**BMOLS**	yes	no		yes	yes	yes	yes
L-NSGA-II	no	no	no		yes	yes	yes
NSGA-III	no	no	no	no		no	no
*R*2-**IBMOLS**	no	no	no	no	yes		yes
MOEA/D	no	no	no	no	yes	no	

algorithms allows the decision maker to fix the number of desired solutions. In our experiments, we have fixed the parameter T to 10 solutions ($T = 10$). The number of nondominated solutions obtained by NSGA-III equals the population size being used (100 solutions in our experiments). This means that all selected solutions of the last generation are nondominated solutions. Depending on the size of the instances (especially the number of the actions), MOEA/D obtains a variable number of nondominated solutions.

Table 6 also shows that even for the smallest instance (“2_50”), NSGA-III, L-NSGA-II and *R*2-IBMOLS generate a high number of output solutions, making decision making difficult. However, the *infiniteOrderLorenzNonDominatedSolutions* procedure (Algorithm 3) could be used to order the generated nondominated solutions and select the most interesting one(s) for the decision maker.

Concerning the runtime, Table 6 shows that for all the considered instances, the proposed algorithms (*IOLD*-EA, *IOLD*-BMOLS and *IOLD* − *R*2-IBMOLS) are more time efficient compared to NSGA-III, L-NSGA-II and *R*2-IBMOLS, except MOEA/D. Indeed, MOEA/D is faster than the proposed algorithms. This gap could be due to the transformation operated on the objective vectors to get the corresponding Lorenz vectors in the algorithms using *IOLD*. We also note that the use of *IOLD* and the parameter T within *R*2-IBMOLS allows one to reduce greatly the runtime of *R*2-IBMOLS (up to a factor of 275 for the instance 3_500 and a factor of 106 for the instance 6_500).

7. Conclusion and perspectives

Within the context of elaborating practical and efficient action plans in social and medico-social structures, we have studied the contribution of the infinite order Lorenz dominance (IOLD), instead of the Pareto dominance, within three population-based multiobjective algorithms: *IOLD*-BMOLS, *IOLD* − *R*2-BMOLS and *IOLD*-EA. Based on this work, we draw two main conclusions. First, using the infinite order Lorenz dominance within indicator-based local search-based or evolutionary algorithms provides a viable method to find high-quality and fair compromise solutions for our action planning problem. Second, this approach provides the decision maker with an effective means to control the number of generated solutions, while helping to reduce significantly the runtime of the algorithm.

As indicated in the introduction, this work is part of the decision support system "*MSQualité*" developed by the Company GePI for social and medico-social sector. Integrating *IOLD* − *R*2-IBMOLS and *IOLD*-EA within "*MSQualité*" constitutes a valuable means that helps managers to elaborate effectively efficient action plans to

Table 6
Comparison of average number of generated solutions and average runtime for *IOLD*-BMOLS, *IOLD* – *R*2-IBMOLS, *IOLD*-EA, NSGA-III, L-NSGA-II, *R*2-IBMOLS and MOEA/D over 30 runs.

Instance	*IOLD*-**BMOLS**		*IOLD* – *R*2-**IBMOLS**		*IOLD*-**EA**		NSGA-III		L-NSGA-II		*R*2-**IBMOLS**		MOEA/D	
	nbSol	*t*	*nbSol*	*t*	*nbSol*	*t*	*nbSol*	*t*	*nbSol*	*t*	*nbSol*	*t*	*nbSol*	*t*
2_50	10	2.35×10^{-1}	10	1.56×10^{-1}	100	6.8×10^{-2}	100	3.16	44	3.74	95	3.22×10^{-1}	10	5.71×10^{-2}
2_100	10	4.68×10^{-1}	10	5.40×10^{-1}	100	1.85×10^{-1}	100	1.09×10^{1}	64	1.18×10^{1}	151	2.46	10	1.24×10^{-1}
2_150	10	1.33	10	1.18	100	3.48×10^{-1}	100	2.48×10^{1}	100	2.44×10^{1}	175	6.53	15	1.85×10^{-1}
2_250	10	5.23	10	3.62	100	7.50×10^{-1}	100	7.04×10^{1}	100	6.64×10^{1}	264	3.35×10^{1}	25	2.60×10^{-1}
2_500	10	3.81×10^{1}	10	1.88×10^{1}	100	2.48	100	3.11×10^{2}	100	2.70×10^{2}	473	3.51×10^{2}	37	7.17×10^{-1}
3_50	10	1.61×10^{-1}	10	8.46×10^{-1}	100	1.24×10^{-1}	100	5.76	100	8.52	820	1.30×10^{1}	12	1.99×10^{-1}
3_100	10	8.48×10^{-1}	10	4.12	100	3.25×10^{-1}	100	1.94×10^{1}	100	2.68×10^{1}	1160	9.38×10^{1}	34	2.66×10^{-1}
3_150	10	2.6	10	9.45	100	6.41×10^{-1}	100	4.43×10^{1}	100	5.62×10^{1}	1748	4.49×10^{2}	48	3.29×10^{-1}
3_250	10	1.09×10^{1}	10	5.80×10^{1}	100	1.36	100	1.26×10^{2}	100	1.39×10^{2}	3369	4.17×10^{3}	79	5.07×10^{-1}
3_500	10	7.42×10^{1}	10	3.21×10^{2}	100	2.83	100	5.37×10^{2}	100	5.98×10^{2}	7635	8.84×10^{4}	128	1.22
4_50	10	1.15	10	1.19	100	0.34	100	8.91	20	1.61×10^{1}	381	7.48×10^{1}	16	2.54×10^{-1}
4_100	10	3.01	10	1.77×10^{1}	100	1.12	100	3.31×10^{1}	28	4.96×10^{1}	666	7.99×10^{1}	37	3.63×10^{-1}
4_150	10	4.47	10	7.44×10^{1}	100	1.16	100	7.06×10^{1}	52	1.02×10^{2}	1551	6.88×10^{2}	91	4.44×10^{-1}
4_250	10	1.66×10^{1}	10	2.67×10^{2}	100	3.39	100	1.88×10^{2}	50	2.60×10^{2}	2385	3.45×10^{3}	114	7.23×10^{-1}
4_500	10	1.31×10^{2}	10	2.51×10^{4}	100	1.33×10^{1}	100	7.52×10^{2}	71	1.06×10^{3}	7591	1.03×10^{5}	259	2.13
5_50	10	2.50	10	1.18	100	9.50×10^{-1}	100	1.37×10^{1}	15	2.68×10^{1}	667	1.84×10^{1}	5	3.03×10^{-1}
5_100	10	5.57	10	4.22×10^{1}	100	1.12	100	4.21×10^{1}	65	7.78×10^{1}	1992	5.14×10^{2}	110	4.30×10^{-1}
5_150	10	6.69	10	1.38×10^{1}	100	2.25	100	8.63×10^{1}	73	1.59×10^{2}	3239	2.59×10^{3}	139	6.04×10^{-1}
5_250	10	1.82×10^{2}	10	5.37×10^{2}	100	9.93	100	3.05×10^{2}	89	3.99×10^{2}	5257	1.58×10^{4}	202	1.10
5_500	10	2.66×10^{2}	10	4.53×10^{3}	100	2.44×10^{2}	100	1.65×10^{3}	96	1.59×10^{3}	15,898	4.77×10^{5}	384	3.11
6_50	10	5.51	10	4.49	100	1.01	100	1.52×10^{1}	27	3.71×10^{1}	799	2.45×10^{1}	6	3.24×10^{-1}
6_100	10	9.99	10	5.66×10^{1}	100	1.16	100	9.16×10^{2}	60	4.81×10^{2}	1860	3.24×10^{2}	13	5.01×10^{-1}
6_150	10	1.29×10^{1}	10	3.40×10^{2}	100	3.30	100	1.88×10^{3}	96	2.30×10^{2}	4971	5.39×10^{3}	115	8.37×10^{-1}
6_250	10	4.99×10^{1}	10	1.54×10^{3}	100	1.21×10^{1}	100	6.63×10^{3}	87	1.53×10^{2}	10,658	5.46×10^{4}	130	1.40
6_500	10	2.46×10^{2}	10	1.23×10^{4}	100	2.96×10^{1}	100	2.06×10^{4}	100	2.38×10^{3}	31,068	1.32×10^{6}	549	4.31
8_50	10	2.20×10^{1}	10	2.27×10^{1}	100	1.44	100	2.46×10^{1}	87	6.60×10^{1}	2147	2.13×10^{3}	23	5.22×10^{-1}
8_100	10	8.78×10^{1}	10	1.87×10^{2}	100	1.98	100	1.44×10^{3}	80	1.30×10^{4}	4052	2.19×10^{4}	130	9.09×10^{-1}
8_150	10	5.56×10^{1}	10	1.38×10^{3}	100	3.63	100	2.88×10^{3}	100	1.31×10^{5}	10,588	3.08×10^{5}	170	1.34
8_250	10	6.13×10^{2}	10	2.86×10^{3}	100	1.60×10^{1}	100	1.01×10^{4}	100	1.97×10^{5}	40,404	2.12×10^{6}	219	3.37
8_500	10	4.08×10^{3}	10	2.20×10^{4}	100	3.43×10^{1}	100	3.16×10^{4}	100	2.03×10^{5}	58,413	2.89×10^{6}	1355	1.10×10^{1}

continually improve the quality of their structures. To further assist the decision maker, we plan to integrate a visualization tool like the chord diagram proposed in Koochaksaraei et al. (2017), which is based on circular layout. This visualization method provides an interesting way to present large volume of data and enables the decision maker to observe, in the 2-D space, the relations among elements of the data and the relations between the optimization objectives. Such a visualization might help the decision maker to detect harmony or conflicts between objectives/actions, consequently, group harmonious objectives or actions.

Finally, this work shows that the proposed approaches work well for the practical problem with many objectives, an interesting future study would be to experiment such methods on other many objective problems. Also, this work demonstrates that even compared to powerful multiobjective methods like MOEA/D and NSGA-III, algorithms based on Lorenz dominance (instead of Pareto dominance) compete favorably. Consequently, it would be interesting to investigate the idea of introducing fairness (based on L-dominance) within other popular methods such as MOEA/D (Zhang and Li, 2007), MOGLS (Ishibuchi and Murata, 1998; Jaszkiewicz, 2001) and PICEA (Wang et al., 2015). Indeed, fairness is an important issue in many multiobjective decision making problems, while studies on multiobjective optimization integrating fairness or equity are still scarce. As a result, introducing dedicated mechanisms (e.g., those studied in this work) in existing multiobjective methods will enlarge their applications to numerous situations where fair and equitable solutions are explicitly sought.

Acknowledgments

We are grateful to the reviewers for their helpful comments and suggestions that helped us to improve the work. This work was partially supported by the Association Nationale de la Recherche et de la Technologie through a CIFRE grant (number 0450/2013). We thank Rachid Naitali, the General Director of GePI, for his support.

References

Alves, M.J., Almeida, M., 2007. MOTGA: a multiobjective tchebycheff based genetic algorithm for the multidimensional knapsack problem. Comput. Oper. Res. 34 (11), 3458–3470.
Bader, J., Zitzler, E., 2011. Hype: an algorithm for fast hypervolume-based many-objective optimization. Evol. Comput. 19, 45–76.
Barichard, V., Hao, J.K., 2003. Genetic tabu search for the multi-objective knapsack problem. J. Tsinghua Sci. Technol. 8 (1), 8–13.
Basseur, M., Burke, M., 2007. Indicator-based multi-objective local search. In: Proceedings of IEEE Congress on Evolutionary Computation, pp. 3100–3107.
Basseur, M., Liefooghe, A., Burke, E., Lee, K., 2012. The efficiency of indicator-based local search for multi-objective combinatorial optimization problems. J. Heuristics 18 (2), 263–296.
Bazgan, C., Hugot, H., Vanderpooten, D., 2009. Solving efficiently the 01 multi-objective knapsack problem. Comput. Oper. Res. 36 (1), 260–279.
Bechikh, S., Kessentini, M., Ben Said, L., Ghédira, K., 2015. Preference Incorporation in Evolutionary Multiobjective Optimization: A Survey of the State-of-the-art. In: Hurson, A.R. (Ed.), *Advances in Computers*. Elsevier, pp. 141–207.
Brockhoff, D., Wagner, T., Trautmann, H., 2012. On the properties of the R2 indicator. In: Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation, GECCO'12, pp. 465–472.
Brockhoff, D., Wagner, T., Trautmann, H., 2014. R2 indicator-based multiobjective search. Evol. Comput. 23 (3), 369–395.
Brockhoff, D., Zitzler, E., 2007. Improving hypervolume based multiobjective evolutionary algorithms by using objective reduction methods. In: 2007 Proceedings of the Congress on Evolutionary Computation (CEC 2007), pp. 2086–2093.
Chabane, B., Basseur, M., Hao, J.K., 2017. R2-IBMOLS applied to a practical case of the multiobjective knapsack problem. Expert Syst. Appl. 71, 457–468.
Deb, K., 2001. Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons.
Deb, K., Jain, H., 2014. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints (NSGA-III). IEEE Trans. Evol. Comput. 18, 577–601.
Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6 (2), 182–197.
Dugardin, F., Yalaoui, F., Amodeo, L., 2010. New multi-objective method to solve reentrant hybrid flow shop scheduling problem. Eur. J. Oper. Res. 203, 22–31.

Emmerich, M., Beume, N., Naujoks, B., 2005. An EMO algorithm using the hypervolume measure as selection criterion. In: Coello, C.A.C., Aguirre, A.H., Zitzler, E. (Eds.), Evolutionary Multi-Criterion Optimization, Vol. 3410. Springer Berlin Heidelberg, pp. 62–76.

Galand, L., Lust, T., 2015. Exact methods for computing all lorenz optimal solutions to biobjective problems. In: Proceedings of the 4th International Conference on Algorithmic Decision Theory, pp. 305–321. LNCS 9346

Golden, B., Perny, P., 2010. Infinite order lorenz dominance for fair multiagent optimization. Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems 1, 383–390.

The Ministry of Social Affairs and Health, 2012. Le champ social et médico-social: une activité en forte croissance, des métiers qui se développent et se diversifient. Repères et Analyses 44.

Ishibuchi, H., Akedo, N., Nojima, Y., 2015. Behavior of multi-objective evolutionary algorithms on many-objective knapsack problems. IEEE Trans. Evol. Comput. 19, 264–283.

Ishibuchi, H., Akedo, N., Murata, Y., 1998. A multi-objective genetic local search algorithm and its application to flowshop scheduling. IEEE Trans. Syst. Man. Cybern. C Appl. Rev. 28 (3), 392–403.

Ishibuchi, H., Setoguchi, Y., Masuda, H., Nojima, Y., 2017. Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes. IEEE Trans. Evol. Comput. 21, 169–190.

Jaszkiewicz, A., 2001. Multiple objective genetic local search algorithm. In: Multiple Criteria Decision Making in the New Millennium. Springer Berlin Heidelberg, pp. 231–240.

Jaszkiewicz, A., 2002. On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - a comparative experiment. IEEE Trans. Evol. Comput. 6 (4), 402–412.

Jaszkiewicz, A., 2004. On the computational efficiency of multiple objective metaheuristics: the knapsack problem case study. Eur. J. Oper. Res. 158, 418–433.

Jiang, S., Ong, Y.S., Zhang, A.N., Tan, P.S., 2015. A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm. IEEE Trans. Cybern. 45 (10), 2202–2213.

Koochaksaraei, R.H., Meneghini, I.R., Coelho, V.N., Guimarães, F.G., 2017. A new visualization method in many-objective optimization with chord diagram and angular mapping. Knowl. Based Syst. 138, 134–154.

Kostreva, M.M., Ogryczak, W., 1999. Equitable approaches to location problems. In: Thill, J.C. (Ed.), Spatial Multicriteria Decision Making and Analysis: A Geographic Information Sciences Approach, pp. 103–126. Ashgate, Brookfield

Kostreva, M.M., Ogryczak, W., 1999. Linear optimization with multiple equitable criteria. RAIRO-Oper. Res. 33, 275–297.

Kostreva, M.M., Ogryczak, W., Wierzbicki, A., 2004. Equitable aggregations and multiple criteria analysis. Eur. J. Oper. Res. 158, 362–377.

Lacour, R., Klamroth, K., Fonseca, C., 2017. A box decomposition algorithm to compute the hypervolume indicator. Comput. Oper. Res. 79, 347–360.

Li, F., Liu, J., Tan, S., Yu, X., 2015. R2–MOPSO: a multi-objective particle swarm optimizer based on r2-indicator and decomposition. In: Proceedings of IEEE Congress on Evolutionary Computation (CEC 2015), pp. 3148–3155.

Moghaddam, A., Yalaoui, F., Amodeo, L., 2011. Lorenz versus pareto dominance in a single machine scheduling problem with rejection. In: Proceedings of the 6th International Conference on Evolutionary Multi-Criterion Optimization, Vol. 6576. Springer-Verlag, Berlin, Heidelberg, pp. 520–534. LNCS

Ogryczak, W., 2000. Inequality measures and equitable approaches to location problems. Eur. J. Oper. Res. 122, 374–391.

Ogryczak, W., 2000. Multiple criteria linear programming model for portfolio selection. Ann. Oper. Res. 97, 143–162.

Paquete, L.C.M., Stützle, T., 2004. Pareto local optimum sets in the biobjective traveling salesman problem: an experimental study. In: Gandibleux, X. (Ed.), Metaheuristics for Multiobjective Optimisation, pp. 177–199.

Perny, P., Spanjaard, O., Storme, L.X., 2006. A decision-theoretic approach to robust optimization in multivalued graphs. Ann. Oper. Res. 147, 317–341.

Phan, H.D., Suzuki, J., 2013. R2-IBEA: R2 indicator based evolutionary algorithm for multiobjective optimization. In: Proceedings of the 2013 Congress on Evolutionary Computation, pp. 1836–1845.

Sen, A., 1973. On Economic Inequality. Clarendon Press Oxford.

Trautmann, H., Wagner, T., D., B., 2013. R2-EMOA: focused multiobjective search using r2-indicator-based selection. In: Proceedings of the 7th International Learning and Intelligent Optimization Conference (LION). Springer, pp. 70–74.

Wagner, T., Trautmann, H., Brockhoff, D., 2013. Preference articulation by means of the R2 indicator. In: Evolutionary Multi-Criterion Optimization, Vol. 7811, pp. 81–95.

Wang, R., Purshouse, R.C., Fleming, P.J., 2015. Preference-inspired co-evolutionary algorithms using weight vectors. Eur. J. Oper. Res. 243 (2), 423–441.

Zhang, Q., Li, H., 2007. MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. 11 (6), 712–731.

Zitzler, E., Künzli, S., 2004. Indicator-based selection in multiobjective search. In: 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII), 3242, pp. 832–842. LNCS

Zitzler, E., Thiele, L., 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. Trans. Evol. Comput. 3 (4), 257–271.