A model of anytime algorithm performance for bi-objective optimization

Alexandre Borges de Jesus, Luis Paquete, Arnaud Liefooghe

Journal of Global Optimization manuscript No.
(will be inserted by the editor)

A model of anytime algorithm performance for bi-objective optimization

Alexandre D. Jesus · Luís Paquete · Arnaud Liefooghe

Received: date / Accepted: date

Abstract Anytime algorithms allow a practitioner to trade-off runtime for solution quality. This is of particular interest in multi-objective combinatorial optimization since it can be infeasible to identify all efficient solutions in a reasonable amount of time. We present a theoretical model that, under some mild assumptions, characterizes the "optimal" trade-off between runtime and solution quality, measured in terms of relative hypervolume, of anytime algorithms for bi-objective optimization. In particular, we assume that efficient solutions are collected sequentially such that the collected solution at each iteration maximizes the hypervolume indicator, and that the non-dominated set can be well approximated by a quadrant of a superellipse. We validate our model against an "optimal" model that has complete knowledge of the non-dominated set. The empirical results suggest that our theoretical model approximates the behavior of this optimal model quite well. We also analyze the anytime behavior of an ε-constraint algorithm, and show that our model can be used to guide the algorithm and improve its anytime behavior.

Keywords Multi-objective Optimization · Combinatorial Optimization · Anytime Algorithms · Anytime Behavior · ε-constraint

This is a post-peer-review, pre-copyedit version of an article published in *Journal of Global Optimization*. The final authenticated version is available online at: `https://dx.doi.org/10.1007/s10898-020-00909-9`

A. D. Jesus
University of Coimbra, CISUC, DEI, Coimbra, Portugal
Univ. Lille, CNRS, Centrale Lille, Inria, UMR 9189 - CRIStAL, F-59000 Lille, France
E-mail: ajesus@dei.uc.pt

L. Paquete
University of Coimbra, CISUC, DEI, Coimbra, Portugal
E-mail: paquete@dei.uc.pt

A. Liefooghe
JFLI - CNRS IRL 3527, University of Tokyo, 113-0033 Tokyo, Japan
E-mail: arnaud.liefooghe@univ-lille.fr

1 Introduction

Multi-objective combinatorial optimization (MOCO) problems, which arise in various real-life scenarios, deal with multiple, typically conflicting, objective functions and are commonly solved according to the notion of *efficiency* [5]. Under this notion there are usually several *efficient solutions* that represent the optimal trade-offs between the objective functions. The goal in MOCO is often to find the set of all efficient solutions, namely the *efficient set*. However, finding the complete set in a reasonable amount of time is often infeasible due to the large number of efficient solutions. In such cases, it can be more relevant to find a concise representation of the efficient set with respect to some notion of representation quality [25]. Still, the preferences of a practitioner in terms of desirable representation quality and/or runtime can, in many cases, be uncertain or vary between calls of the algorithm, e.g. a real-time system that has varying time available to report a result.

As such, anytime algorithms [3], which are characterized by their ability to return an approximation of improving quality at anytime of the search process, or for anytime budget, are an appealing concept since they allow to trade-off representation quality with runtime. Many algorithms for MOCO problems can be described as anytime algorithms, since they sequentially find improving solutions during runtime. For example, ε-constraint approaches [5] sequentially collect efficient solutions by solving a sequence of constrained single-objective optimization problems. However, not every algorithm will present a good *anytime behavior*, i.e. a good trade-off between representation quality and runtime [30].

In this work we aim to characterize the "optimal" anytime behavior of a class of anytime algorithms for bi-objective combinatorial optimization that collect, at each iteration, an efficient solution by solving a scalarized problem [5]. Optimal anytime behavior here means that the collected efficient solution at each iteration maximizes a measure of representation quality. In order to characterize this optimal anytime behavior, we propose a theoretical model that estimates the trade-off between runtime, given by the number of iterations performed by the algorithm, and representation quality, given by the hypervolume [31] of the collected solutions. This model acts under the assumption that the "shape" of the *non-dominated set*, i.e. the image of the efficient set in the objective space, can be well approximated by a quadrant of a superellipse. Although this seems to be a strong assumption, empirical results suggest that this is the case for linear sum objective functions, as is also illustrated in this work for two variants of the bi-objective knapsack problem.

The model works by collecting minimal information about the non-dominated set in order to approximate it with a piecewise linear function. Then, we consider an oracle that returns, at each call, a point from this piecewise linear approximation that maximizes the hypervolume indicator. We show that, for a particular case, the relation between the number of calls to the oracle and the hypervolume achieved can be expressed as a simple analytical formulation that depends on a "curvature" parameter of the superellipse. For the general case, we present a simple method that efficiently constructs the model by simulating the oracle.

To validate our theoretical model, we consider an "optimal" model that has complete knowledge of the non-dominated set. Empirical results on two variants of the bi-objective knapsack problem indicate that the theoretical model approximates the anytime behavior of this optimal model quite well. Moreover, we analyze

the anytime behavior of an ε-constraint approach [5], and show that how our model can efficiently guide this approach in order to improve its anytime behavior. In particular, at each iteration we use the location of the point returned by the oracle as a prediction of the location of the actual non-dominated point that maximizes the hypervolume contribution. This information can then be used to set the constraint in the scalarized single-objective problem solved by the ε-constraint algorithm at each iteration.

The rest of this article is organized as follows. In Section 2, we present the background related to anytime algorithms and representation quality. In Section 3, we introduce the relevant definitions. In Section 4, we describe the proposed theoretical model. In Section 5, we present the empirical study and discuss the results obtained. In Section 6, we present the concluding remarks and discuss possible directions for future work.

2 Background

In this section, we start by introducing the notion of anytime algorithms and their desirable properties. Then, we discuss different measures of representation quality, and how they relate to these desirable properties. We conclude by briefly discussing some related work.

2.1 Anytime algorithms

Anytime algorithms [3,30] allow to trade-off approximation quality with runtime, such that quality improves as a function of runtime. Such algorithms are relevant in various scenarios, for example in real-time systems where the available time for the algorithm varies between calls, or in composite systems where the allocated time for each algorithm varies between instances in order to improve the result of the overall system.

A distinction is often made between two categories of anytime algorithms, *interruptible* and *contract* [30]. Interruptible algorithms do not require a priori knowledge of the available time budget and return an approximation when interrupted at anytime of the search process. On the other hand, contract algorithms require that the time budget is known prior to execution, and may or may not return a solution if interrupted before the time budget ends. The anytime characteristic of the former lies on the fact that an approximation can be found at anytime by interrupting the algorithm, while on the latter an approximation can be found at anytime by providing any time budget to the algorithm.

Many search heuristics and exact approaches for combinatorial optimization problems naturally present the characteristics of interruptible algorithms. Still, not every such algorithm is interesting from an anytime perspective. Zilberstein [30] considers seven desirable properties of anytime algorithms:

1. *Measurable quality*, the quality of an approximation can be measured;
2. *Recognizable quality*, the quality of an approximation can be determined with little or no overhead at runtime;
3. *Monotonicity*, the quality of the approximation improves monotonically as a function of time;

4. *Consistency*, the quality of the approximation as a function of time is consistent for the same input parameters;
5. *Diminishing returns*, the improvement in quality for the approximation found is greater at earlier steps of runtime, and diminishes over time;
6. *Interruptibility*, the algorithm returns an approximation when interrupted during the execution; and
7. *Preemptability*, the algorithm can be resumed after being interrupted.

In the context of our work, we focus the following discussion on the first five properties, for which, two main aspects need to be considered. First, the choice of a measure of approximation quality, which is discussed in Section 2.2. Secondly, the behavior of the anytime algorithm with respect to its trade-off between approximation quality and runtime, namely its *anytime behavior*.

The anytime behavior is often characterized in terms of performance profiles [30,9]. Two categories of profiles are commonly considered: *expected performance profiles*, which map the expected quality as a function of time, and *probabilistic performance profiles*, that describe the probability of finding a solution of a certain quality after a specific amount of time. Performance profiles are often empirically built by running the algorithm on several instances. However, this empirical approach may be computationally expensive or infeasible if, for example, not enough instances are known to build a meaningful profile. As such, it can be more valuable to build a theoretical profile that describes a specific algorithmic behavior.

2.2 Representation quality

In this work, we consider that anytime algorithms return approximations that correspond to subsets of the efficient set, namely *representation sets*. Some scalarization techniques [5] such as ε-constraint approaches and weighted sum methods meet this assumption, since they can find efficient solutions by solving a sequence of scalarized single-objective problems. In this section we discuss different representation measures that describe the quality of a representation, and how they relate to the properties defined in the previous section.

Several ways of defining representation measure can be found in the literature:

- *Uniformity* [25], is a measure of how far apart the points in the representation are from each other, and is defined by the distance between the two closest points in the representation.
- *Coverage* [25], is a measure of how distant the points in the representation are from those in the non-dominated set, and is defined by the maximum distance between a non-dominated point and its closest point in the representation set.
- *ε-indicator (multiplicative)* [27,13], is defined by the smallest multiplicative factor that can be applied to the all points in the representation, such that each point of the non-dominated set is dominated by at least one point in the representation set.
- *Hypervolume* [15], corresponds to the measure of the multi-dimensional region dominated by the points in the representation set with respect to a reference point.

Clearly, the uniformity measure allows for measurable quality since it only requires that the points in the representation are known. On the other hand, the coverage and ε-indicator measures are only measurable if the non-dominated set is known, which is unrealistic in practice. Lastly, the hypervolume is measurable as long as the reference point is known, for example, if it is provided by the decision maker a priori, or if it can otherwise be defined by the practitioner, e.g. as the nadir point of the non-dominated set. This may not be possible in cases where no information is known regarding the objective space and objective functions.

With respect to the recognizable property, some overhead is commonly required to calculate the representation measure during runtime, unless the measure is calculated during the execution of the algorithm, e.g. to guide its search process, in which case we can reuse it. Whether this overhead is negligible or not depends on the considered application. Still, it is worth noting that the coverage indicator and the ε-indicator depend on the size of the non-dominated set, which can be non-negligible relatively to the size of the representation set. Moreover, the computational runtime of the hypervolume measure grows exponentially with respect to the number of objective functions [29], while the remaining measures grow linearly [25].

As for the monotonicity property, if we consider an algorithm that collects efficient solutions and does not forget previously found solutions, then all four indicators will improve monotonically as solutions are collected. However, only the hypervolume indicator is strictly monotonic when a previously unseen non-dominated point is found, under the common assumption that the reference point is strictly dominated by all non-dominated points [32]. Strict monotonicity is important to guarantee the diminishing returns property since, otherwise, there may be occasional iterations where a new solution is collected but no quality improvement is observed, interleaved with iterations that show an improvement of quality. Moreover, since the behavior during those occasional periods of runtime cannot be measured, it is not possible to compare different approaches on these periods.

2.3 Related work

The study of anytime algorithms and their behavior commonly focuses on two aspects. The first, *meta-level control*, is concerned with the optimal allocation of runtime for systems with one or more anytime algorithms. This is often done according to some utility function that models the practitioner preferences in terms of the trade-off between solution quality and elapsed runtime. This allocation may be performed prior to execution [1, 11], in particular when the anytime behavior is predictable and consistent between instances, or during execution [10] by monitoring the anytime behavior of the algorithm. It is worth noting that monitoring the algorithm can possibly take a significant amount of time, in which case the monitorization needs to be scheduled accordingly [7].

The second aspect is concerned with the development and improvement of anytime algorithms with respect to their anytime behavior [4, 19]. This is often done by manually analyzing the performance profiles of different algorithms and configurations. However, this requires human intervention, which can be undesirable or infeasible at times. As such, a more interesting approach is to consider methodologies to automatically compare the performance profiles of different al-

gorithms. In recent works, this has been done by considering the trade-off between solution quality and runtime as a bi-dimensional non-dominated set, which allows the practitioner to use bi-objective quality indicators to compare different performance profiles for heuristic methods [24, 18, 17]. In particular, these studies consider the hypervolume indicator [31] to describe anytime behavior as a scalar value, and use `irace` [16] to automatically tune the parameters of an algorithm with the goal of improving its anytime behavior. Up to our knowledge no studies exist on theoretical models of optimal anytime behavior for MOCO scalarization techniques, which can be used, for example, to improve the anytime behavior of these techniques, as we show in this paper.

3 Definitions

In this section, we introduce the relevant definitions for MOCO, and for the hypervolume as a representation measure.

3.1 Multi-objective combinatorial optimization

We consider MOCO problems with two maximizing objective functions, w.l.o.g., defined as follows

$$\max_{x \in X} \; f(x) = (f_1(x), f_2(x)) \tag{1}$$

where X denotes the feasible set of solutions, and $f_i : X \to \mathbb{R},\ i \in \{1, 2\}$ are the two objective functions. The image of X in the objective space is denoted by $Y = \{f(x) \mid x \in X\}$. We consider the following two orderings in the objective space

$$\begin{aligned} f(x) \geq f(x') \quad &\text{iff} \quad f_i(x) \geq f_i(x'),\ i \in \{1, 2\} \text{ and } f(x) \neq f(x') \\ f(x) > f(x') \quad &\text{iff} \quad f_i(x) > f_i(x'),\ i \in \{1, 2\} \end{aligned} \tag{2}$$

such that a solution $x \in X$ is said to be *efficient* if no other solution $x' \in X$ exists such that $f(x') \geq f(x)$, and *weakly efficient* if no other solution $x' \in X$ exists such that $f(x') > f(x)$. The image of these solutions in the objective space are called *non-dominated* and *weakly non-dominated* respectively.

The set of all efficient solutions is called the *efficient set* and denoted by X^*, and its image in the objective space is called the *non-dominated set* and denoted by Y^*. A *representation set* R corresponds to a subset of the efficient set, i.e. $R \subseteq X^*$, and its image in the objective space is denoted by $Y_R = \{f(x) \mid x \in R\}$.

3.2 Hypervolume indicator

To measure the quality of a representation set for a bi-objective optimization problem, we consider the *hypervolume indicator* [31], which corresponds to the measure of the bi-dimensional area dominated by the image of a representation set in the objective space $Y_R \subset \mathbb{R}^2$, bounded by a reference point $r \in \mathbb{R}^2$, that is

$$H(Y_R) = \lambda \left(\left\{ q \in \mathbb{R}^2 \mid \exists\ s \in Y_R \colon r \leq q \leq s \right\} \right) \tag{3}$$

where λ denotes the Lebesgue measure. We also introduce the notion of *hypervolume contribution*, which corresponds to the contribution of a point $q \in \mathbb{R}^2$ with respect to a set $Y_R \subset \mathbb{R}^2$, and is given by

$$\Delta H(q, Y_R) = H(Y_R \cup \{q\}) - H(Y_R) \tag{4}$$

Noteworthily, the hypervolume is a monotone submodular function [26]. As such, an algorithm that sequentially collects efficient solutions that maximize the hypervolume contribution provides, after collecting k solutions, an $(1 - 1/e)$-approximation to the maximal hypervolume value of a representation with size k [22], where e is the base of the natural logarithm. Moreover, the hypervolume is scaling independent [14], and as such the ordering defined by the hypervolume among all representation sets is not affected by a scaling in the objective space. However, the ordering between representation sets given by the hypervolume indicator depends on the choice of the reference point [14].

4 A theoretical model of anytime behavior

In this section, we propose a theoretical model of optimal anytime behavior for bi-objective optimization anytime algorithms that collect efficient solutions sequentially. We consider that optimal anytime behavior means that the collected solution at each iteration maximizes the hypervolume contribution. For clarity of exposition, the model is split into two parts.

First, in Section 4.1, we define a piecewise linear approximation of the non-dominated set under some assumptions. In particular, we assume that the objective values of the lexicographic optimal solutions are known and that the non-dominated set can be well approximated by the positive quadrant of a superellipse. Although it is not expected that the non-dominated set matches the positive quadrant of a superellipse exactly, our findings suggest that this gives a good approximation in practice for many problems with linear sum objective functions. Superellipses have also been studied in the context of multi-objective continuous optimization for the generation of test problems [6].

Then, we present two formulations of the model. In Section 4.2, we define an analytical formulation for the particular case where the piecewise linear approximation consists of two segments and is convex, and when the reference point for the hypervolume indicator is the nadir point. Then, in Section 4.3, we present a more general algorithm that works for both the convex and non-convex cases, for any reference point setting, and for a piecewise linear approximation defined by any number of linear segments.

4.1 Estimating the non-dominated set as a piecewise linear function

We assume that the non-dominated set, scaled down to the unit square $[0, 1]^2$, can be well approximated by the positive quadrant of a superellipse centered in the origin with both semi-diameters of length one. Formally, this is given by the following parametric equation

$$y_1{}^d + y_2{}^d = 1 \tag{5}$$

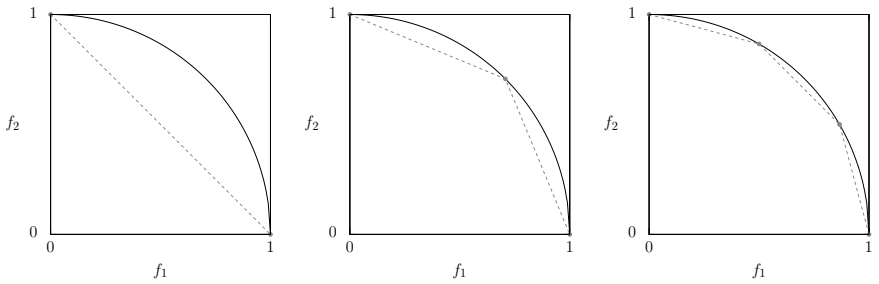


Fig. 1 Example of superellipse curve (continuous line) for $d = 2$ and corresponding piecewise linear approximation (dashed line) for $\ell \in \{1, 2, 3\}$, from left to right.

where $d > 0$ is a parameter that controls the curvature of the superellipse and $y_1, y_2 \in [0, 1]$. Alternatively, the values of y_1 and y_2 can also be defined with respect to a parameter $\theta \in [0, \pi/2]$ as follows

$$y_1 = \cos^{2/d} \theta \tag{6}$$

$$y_2 = \sin^{2/d} \theta \tag{7}$$

Note that, for $d < 1$ the resulting approximation is non-convex, whereas for $d \geq 1$ it is convex.

We then consider a piecewise linear approximation, denoted by G, with $\ell > 0$ linear segments defined between consecutive points in the curve approximation

$$p_i = \left(\cos^{2/d} \theta_i, \sin^{2/d} \theta_i\right) \tag{8}$$

where $i \in \{1, \ldots, \ell + 1\}$ and $\theta_i < \theta_{i+1}$, such that θ_i are set to evenly spaced values in the interval $[0, \pi/2]$. This setting of θ_i provides a good approximation for the purposes of our model. Still, other techniques are available to sample these points, see for example [23]. Figure 1 illustrates the curve given by Equation 5 for a particular value of d, and corresponding piecewise linear approximations G for a varying number of segments.

4.2 Analytical model

We define the behavior of an oracle that returns, at each call, a point in G that maximizes the hypervolume by a function $C(j)$ that denotes the hypervolume contribution of the j-th point returned by the oracle, and by a function $M(k)$ that denotes the relative hypervolume after k points, that is

$$M(k) = \frac{1}{H(G)} \sum_{j=1}^{k} C(j) \tag{9}$$

Under some assumptions we can define closed formulations for $C(j)$ and $M(k)$. In particular, we assume that: the approximation curve is convex ($d \geq 1$), the piecewise linear approximation has only two linear segments such that $\theta_1 = 0$,

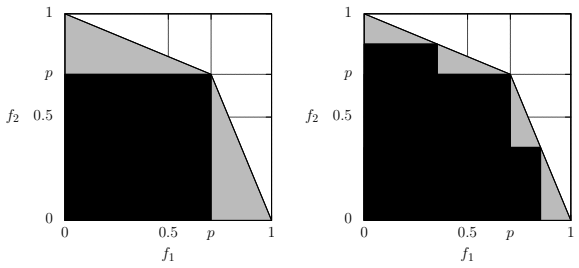


Fig. 2 Example of the dominated (in black) and uncovered (in gray) regions after collecting one (left) and three (right) points with the oracle.

$\theta_2 = \pi/4$, and $\theta_3 = \pi/2$, and the reference point for the hypervolume indicator is set to the nadir point, i.e. $r = (0, 0)$.

Under these assumptions, the piecewise linear approximation is defined by

$$G = \{(y_1, g(y_1)) \mid y_1 \in [0, 1]\} \tag{10}$$

$$g(y_1) = \begin{cases} \frac{p-1}{p} y_1 + 1 & , 0 \leq y_1 \leq p \\ \frac{p}{p-1} y_1 + \frac{p}{1-p} & , p < y_1 \leq 1 \end{cases} \tag{11}$$

such that

$$p = \cos^{2/d} \frac{\pi}{4} = \sin^{2/d} \frac{\pi}{4} \tag{12}$$

Then, the first point returned by the oracle for this piecewise linear approximation can be found by solving

$$\underset{y_1 \in [0,1]}{\arg\max} \; y_1 \cdot g(y_1) \tag{13}$$

which for $d \geq 1$ has a single global optimum at coordinates (p, p). As such, the hypervolume contribution of the first point collected by the oracle is $C(1) = p^2$.

Subsequent points can be found by considering the regions that are not dominated by any of the points collected. These regions are said to be *uncovered*. After the first point (p, p) has been identified, each uncovered region is defined by a right triangle, such that the hypotenuse contains the non-dominated points of the uncovered region. Figure 2 illustrates the uncovered and dominated regions, which are colored in gray and black respectively, after collecting one point (left-hand side) and three points (right-hand side).

In order to find the largest hypervolume contribution for an uncovered region defined by such a right triangle, we solve the following problem

$$\begin{aligned} \max \quad & y_1 \cdot y_2 \\ \text{s.t.} \quad & y_2 = -\frac{c_2}{c_1} \cdot y_1 + c_2 \\ & y_1 \in [0, c_1] \\ & y_2 \in [0, c_2] \end{aligned} \tag{14}$$

where c_1 and c_2 denote the length of the catheti of the right triangle on the f_1 and f_2 axes, respectively. This problem has a global optimum at coordinates

$(c_1/2, c_2/2)$, which gives an hypervolume of $c_1 \cdot c_2/4$, corresponding to half of the area of the uncovered region. Also note that, after collecting this optimal point for an uncovered region, the latter is further split into two smaller uncovered regions that are also right triangles. Each of these uncovered regions has catheti with half the length of the original triangle, and consequently covers a quarter of the area.

After collecting the first point, the resulting uncovered regions are two equivalent right triangles with catheti of size $(1-p)$ and p, and an area of $(1-p) \cdot p/2$. Then, it follows that the second and third largest hypervolume contributions are given by $C(2) = C(3) = (1-p) \cdot p/4$. After excluding the regions dominated by the two points that provide these contributions, one for each uncovered region, there are four equivalent uncovered regions as illustrated on the right-hand side of Figure 2. Each time the points that provide the largest hypervolume contribution for all equivalent uncovered regions are collected, the number of uncovered regions doubles. As such, a general equation for $C(j)$ is given by

$$C(j) = \begin{cases} p^2 & j = 1 \\ \frac{(1-p) \cdot p}{4^{\lfloor \log_2 j \rfloor}} & j \geq 2 \end{cases} \tag{15}$$

which expectedly shows that the relative hypervolume has a logarithmic rate of convergence. Finally, note that for reference point $r = (0, 0)$ and $\ell = 2$ linear segments, the hypervolume of the piecewise linear approximation is given by $H(G) = p$.

4.3 Algorithmic model

For the general case, i.e. any number of linear segments $\ell > 0$, any curvature parameter $d > 0$, and any reference point $r \in \mathbb{R}^2$, the previous analytical model does not hold. Instead, we present an algorithm that computes the sequence of maximal hypervolume contributions. This algorithm works by keeping an updated set of uncovered regions. Since, at each iteration, only one point from a single uncovered region is collected, we can keep a cache of the largest hypervolume contribution for each region to avoid repeated calculations. Additionally, to quickly find the next point with the largest hypervolume contribution, the uncovered regions can be kept in a priority queue with respect to the largest possible hypervolume contribution.

These uncovered regions are represented by a simple polygon such that the non-dominated set is defined by a simple polygonal chain, as illustrated in Figure 3. Note that each segment of the polygonal chain is part of the hypotenuse of a right triangle with its right angle on the origin, as illustrated in the same figure. Furthermore, we know from Equation 14 that the point with the largest hypervolume for a right triangle is $z = (c_1/2, c_2/2)$, where c_1, c_2 are the lengths of the catheti on axes f_1, f_2 respectively. Then, the point with maximal hypervolume contribution for a segment of the polygon chain is given by finding the closest point to z. To find the largest hypervolume contribution for an uncovered region we can loop over the segments and find the point with the largest hypervolume contribution. Figure 4 illustrates the three first steps of the algorithmic model for a non-convex piecewise approximation with $\ell = 2$ linear segments.

To describe the complexity of the algorithmic model for function $C(j)$, let us consider a piecewise linear approximation defined by ℓ segments. At each iteration,

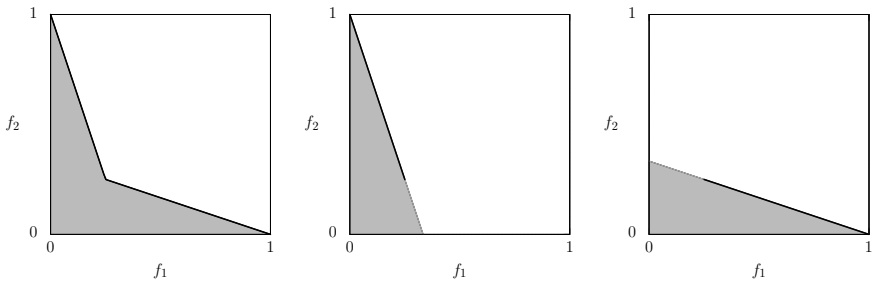


Fig. 3 On the left, an example of an uncovered region defined by a simple polygon (gray area). On the middle and on the right, the individual segments (black line) of the polygonal chain, and their respective triangular regions (gray area).

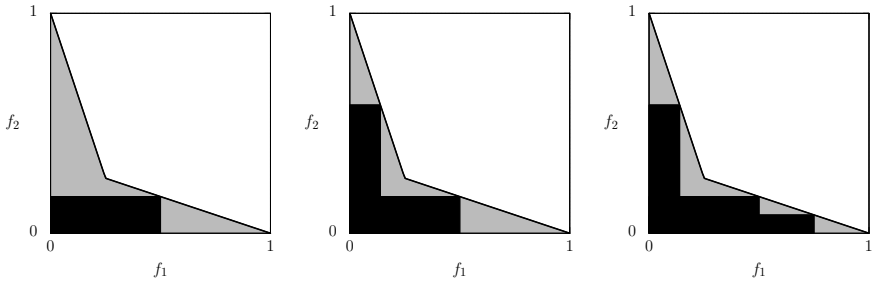


Fig. 4 First three steps of the algorithmic model for curvature parameter $d = 0.5$, number of linear segments $\ell = 2$, and reference point $r = (0, 0)$.

the algorithm needs to find the uncovered region with the largest hypervolume contribution, remove it from the data structure that contains all uncovered regions, split the region into two new uncovered regions and insert those regions into the data structure. As such, after j calls, there will be at most $j + 1$ uncovered regions in the data structure. To perform these operations efficiently we can consider a priority queue supported by a binary heap which has a worst case complexity of $O(\log j)$ for the insert and delete operations, and $O(1)$ worst case complexity to find the region with the largest hypervolume contribution. Still, at each iteration, the algorithm needs to compute the maximum hypervolume contribution within each of the two new uncovered regions before inserting them in the priority queue. Since finding the point for a linear segment that has the largest hypervolume contribution can be done in constant time, and there are at most ℓ segments in each uncovered region, the worst case complexity is given by $O(\ell)$ for each of the two regions. As a result, for function $C(j)$, the algorithmic model has a worst case complexity of $O(j\ell + j \log j)$.

For function $M(k)$, the algorithm first needs to compute the maximal hypervolume of the piecewise linear approximation. This can be done by calculating the area of the first uncovered region which is described by a polygon. The area of a polygon can be calculated, for example, with the surveyor's area formula [2]. Assuming that the points that make the linear segments of the piecewise approximation are given in a clockwise or counterclockwise order, which is the case for the

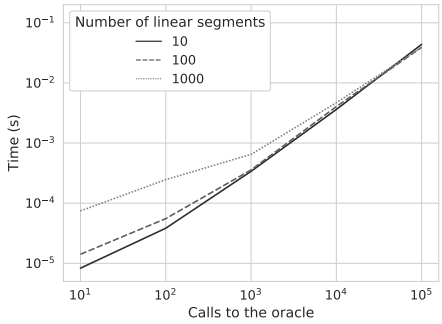


Fig. 5 Execution times in seconds of the algorithmic model for a varying number of calls to the oracle, a varying number of linear segments, reference point $r = (0, 0)$ and curvature parameter $d = 2$.

points described in Equation 8, the calculation of the area with the surveyor's area formula has a worst case complexity of $O(\ell)$. It is worth noting that to define the polygon, we may need to consider some extra points. However, there are at most $\ell - 1$ extra points between the segments, and two extra points between the extreme linear segments and the reference point. As such, the worst case complexity remains $O(\ell)$. Then, the algorithm needs the value of $C(j)$ for $j \in \{1, \ldots, k\}$. However, to calculate $C(j)$ for $j > 1$ we inevitably calculate values $C(j-1), C(j-2), \ldots, C(1)$ as well. Then for $M(k)$ we only need to call $C(k)$ and keep the intermediate values. As such, the worst case complexity for function $M(k)$ is given by $O(k\ell + k \log k)$.

An implementation of this algorithmic model is made available in [12]. Figure 5 shows the execution time in seconds of the implementation for a varying number of calls to the oracle and a varying number of linear segments. The reference point and curvature parameter d are fixed, since they have a negligible impact on the results. We observe that the algorithm is very fast (less than 0.1 seconds) for most practical scenarios, which we expect to have a number of linear segments and iterations within the values reported.

5 Experimental analysis

In this section, we compare the relative hypervolume obtained from our theoretical model described in Section 4 with an "optimal" model that has complete knowledge of the non-dominated set and selects a non-dominated point that maximizes the hypervolume contribution at each iteration. We also show the anytime behavior of two variants of an ε-constraint approach that collects efficient solutions by solving a sequence of constrained single-objective problems, such that one of the variants is guided by our theoretical model and the other is not. The guided variant takes into account the points found by the algorithmic model and uses them to set the constraint of the scalarized problem at each iteration.

We perform this study on an unconstrained bi-objective binary knapsack problem and a variant thereof. Throughout this section, it is assumed that the non-dominated set of the test instances is scaled to the unit square $[0, 1]^2$. In particular, this is done by applying a transformation of the form $f'_i(x) = (f_i(x) - y^l_i)/(y^u_i - y^l_i)$,

where y_i^l and y_i^u are the lower and upper bounds of the non-dominated set on the ith objective function. These bounds are found by finding the lexicographic optimal solutions.

5.1 Problems and instance generation

For the experimental study we consider the bi-objective unconstrained knapsack problem (UBKP). Formally, given a set of items, where each item $j = 1, \ldots, n$ has a value v_j and a weight w_j, the problem is defined as

$$\max_{x \in X} \left(f_1(x) = \sum_{j=1}^{n} v_j x_j, \;\; f_2(x) = -\sum_{j=1}^{n} w_j x_j \right) \tag{16}$$

where x_j denotes a binary variable that indicates whether or not item j has been chosen for the knapsack. The weights and values of the items are randomly generated according to a multivariate uniform distribution in the range $]0, 1[$, with correlation $\rho \in [-1, 1]$ following the procedure described in [28].

Noteworthily, for $\rho = 1$ the points in the non-dominated set fall on a line, which means that the approximation to a positive quadrant of the superellipse is parameterized by $d = 1$. As the value of ρ decreases we expect that the front can be approximated by a convex positive quadrant of a superellipse of increasing parameter d. To illustrate this, Figures 6(a)–(c) show the non-dominated set for instances of size $n = 100$ and correlation values $\rho \in \{-0.8, 0.0, 0.8\}$. In these figures, we can see that for $\rho = 0.8$ the shape of the non-dominated set almost resembles a line, and as ρ decreases, the non-dominated set resembles a more accentuated convex curve. The same figures show that the approximation set can be well approximated with the positive quadrant of a superellipse with curvature parameter $d = \log_p 0.5$, where p is found by considering the solution to the following optimization problem

$$\arg\max_{x \in X} \;\; \min \; \{f_1(x), f_2(x)\} \tag{17}$$

such that if $x' \in X$ is the optimal solution to this formulation, then the parameter is given by $p = (f_1(x') + f_2(x'))/2$.

As expected, and evidenced in Figures 6(a)–(c), the test instances generated for the UBKP do not show non-dominated sets that are approximated by a non-convex superellipse approximation. As such, to validate our model for such cases, we consider a variant of this problem characterized by a tight capacity constraint $c = 1$ on the number of items to be included in knapsack, that is

$$\sum_{j=1}^{n} x_i \leq c \tag{18}$$

We note that, for $c = 1$, this problem is easy to solve since the non-dominated set is composed by the empty solution and all the non-dominated solutions that contain only a single item. We use this problem as a simple example to study our model for non-convex fronts. We refer to this variant as the capacity-constrained UBKP (CCUBKP). To generate test instances for the CCUBKP, we sample the value and weight vectors from equally parameterized Weibull distributions. In particular we

consider a fixed scale parameter $\lambda = 1$, and a varying shape parameter s. Then, we sort the value vector in increasing order and the weight vector in decreasing order.

We expect that by varying the shape parameter s we can vary the curvature of the non-dominated set. In particular, it is expected that as s increases, so does the parameter d for the superellipse approximation. To illustrate this, we show in Figures 6(d)–(f) the non-dominated set of test instances generated for shape parameter values $s \in \{1.6, 2.0, 3.0\}$ and problem size $n = 2000$. A larger problem size for the CCUBKP was chosen in order to have a similar number of non-dominated points between the two problems. From these figures, it is clear that as s increases we get a non-dominated set that more closely resembles a line. To conclude, we show in Figures 6(d)–(f) that the non-dominated sets generated for the CCUBKP can also be well approximated with the positive quadrant of a superellipse. The parameterization of the positive quadrant is done in the same manner as in the case of the UBKP, by solving Problem 17.

5.2 Models and ε-constraint algorithms

In this section, we describe the theoretical model, optimal model, and ε-constraint algorithms used for the empirical study.

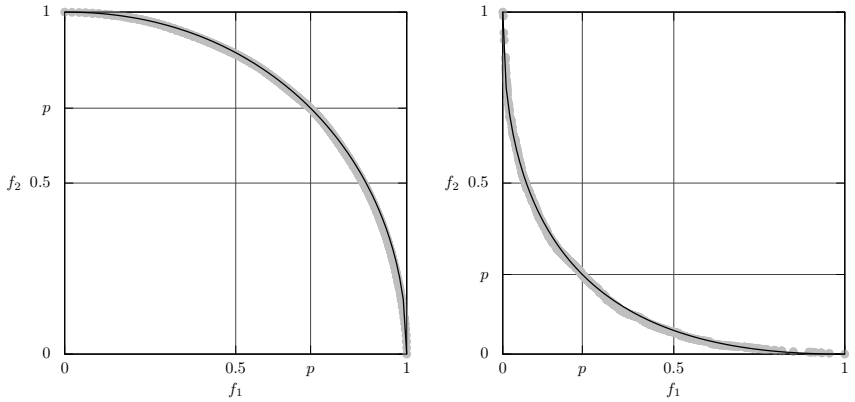
Theoretical model We consider Problem 17 to find the parameter d of the piecewise linear approximation. Moreover, the theoretical model is implemented according to the algorithmic model described in Section 4.3.

Optimal model The optimal model is implemented with a greedy approach that, given the non-dominated set, selects the non-dominated point that provides the largest hypervolume contribution at each iteration. In case there is more than one point with maximal hypervolume contribution, the model selects the lexicographically smaller with respect to the order (f_1, f_2). The non-dominated set is calculated using the Nemhauser-Ullman algorithm [21] for the UBKP, and with a simple enumeration algorithm for the CCUBKP.

ε-constraint algorithm We consider an ε-constraint [5] technique that solves a sequence of constrained single-objective problems by transforming one of the objectives into a constraint

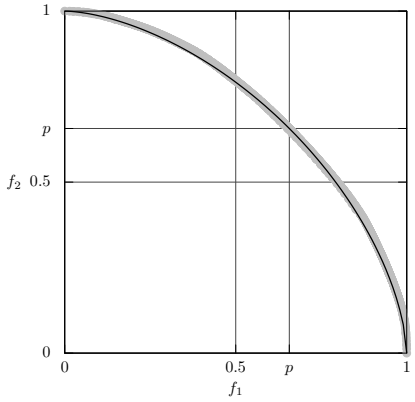
$$\begin{aligned} \arg\max_{x \in X} \; & f_1(x) \\ \text{s.t. } & f_2(x) \geq \varepsilon \end{aligned} \tag{19}$$

where ε is varied at each iteration of the algorithm, denoted by ε_k for the kth iteration. We start by setting $\varepsilon_1 = p$, and then our algorithm bisects the intervals $[0, \varepsilon_1]$ and $[\varepsilon_1, 1]$ to set the constraints $\varepsilon_2 = p/2$ and $\varepsilon_3 = p + (1 - p)/2$ for the second and third iterations, respectively. This bisection procedure is repeated for each point found until a predefined number of iterations is reached. Clearly, this bisection follows the same pattern as the analytical model, where at each iteration we bisect the uncovered regions.

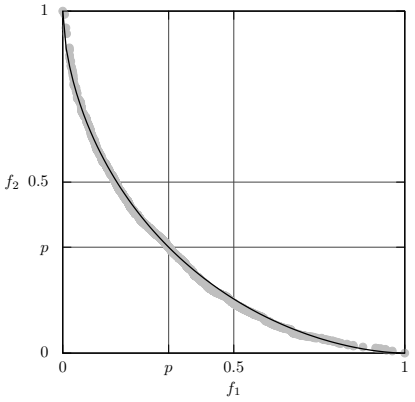


(a) $\rho = -0.8$

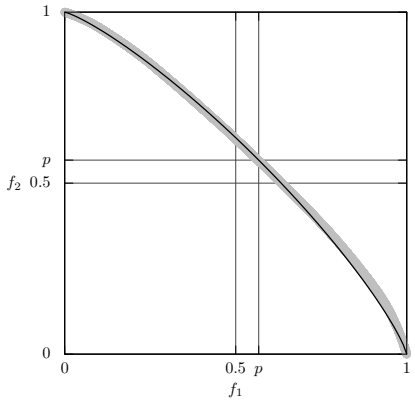
(d) $s = 1.6$



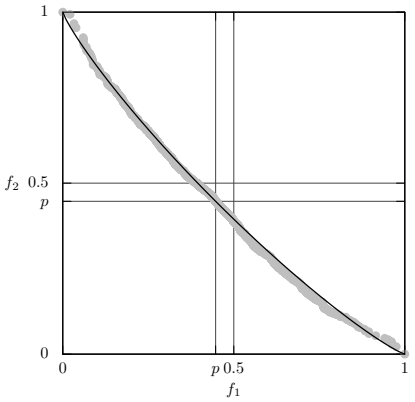
(b) $\rho = 0.0$



(e) $s = 2.0$



(c) $\rho = 0.8$



(f) $s = 3.0$

Fig. 6 Non-dominated set (points in gray) and respective superellipse approximation (straight black line) for instances of the UBKP with varying correlation ρ and problem size $n = 100$ (left), and for instances of the CCUBKP with varying shape parameter s and problem size $n = 2000$ (right).

Since Problem 19 can return a weakly efficient solution [5], we solve at each iteration a second single-objective constrained problem to guarantee an efficient solution, defined as

$$\begin{aligned} &\arg\max_{x \in X} \ f_2(x) \\ &\text{s.t. } f_1(x) \geq f_1(x') \end{aligned} \tag{20}$$

where $x' \in X$ denotes an optimal solution for Problem 19. Both optimization problems are solved using the GNU Linear Programming Kit MILP Solver [20].

Guided ε-constraint algorithm We consider an ε-constraint variant defined by the same formulations presented in Problems 19 and 20, but where the ε_k constraint at each iteration is set according to our model. In particular, the algorithmic model returns, at each iteration, the point in the piecewise linear approximation that maximizes the hypervolume contribution. Then, the value of ε_k is set to the second coordinate of the kth point returned by the model.

5.3 Results and discussion

In this section, we report the solution quality over runtime for the different models and algorithms described in the previous section. In particular, we consider theoretical models with two and ten linear segments, i.e. $\ell \in \{2, 10\}$. Moreover, we consider a theoretical model with $\ell = 10$ linear segments to use with the guided ε-constraint algorithm. In the reported results, runtime corresponds to the number of calls to the oracle for the theoretical and optimal models, and iterations for the two ε-constraint algorithms. The quality of a representation set R is expressed in terms of its relative hypervolume, that is, $H(Y_R)/H(Z)$, where $Z = G$ for the theoretical model, i.e. the approximation to the non-dominated set, and $Z = Y^*$ for the optimal model and ε-constraint algorithms, i.e. the actual non-dominated set of the problem instance. We also report quality in terms of the relative hypervolume deviation, that is, $(H(Z) - H(Y_R))/H(Z)$.

Figures 7 and 8 show the results for the first 128 iterations of the ε-constraint algorithms and calls to the oracle for the UBKP. In the former, the reference point is set to $r = (0, 0)$, while in the latter it is set to $r = (-1, -1)$. The instances considered have a problem size $n = 100$, and correlation values $\rho \in \{-0.8, 0.0, 0.8\}$. The corresponding non-dominated sets are shown in Figures 6(a)–(c). Figures 9 and 10 show the corresponding results for the CCUBKP. The instances considered have a problem size $n = 2000$, and varying shape parameter $s \in \{1.6, 2.0, 3.0\}$. The corresponding non-dominated sets are shown in Figures 6(d)–(f).

These results indicate that our model approximates quite well the performance of the optimal model. In particular, for $\ell = 10$ linear segments the theoretical model is very close to the optimal model. For $\ell = 2$ linear segments there is a larger difference during the first few steps due to the difference between the maximal hypervolume of the non-dominated set and of the piecewise linear approximation. This difference is smaller for reference point $r = (-1, -1)$ and when d is closer to 1 since the relative difference between the maximal hypervolume of the non-dominated set and of the piecewise approximation is smaller in those cases.

Furthermore, the results show that the basic ε-constraint algorithm has a good anytime behavior for $r = (0, 0)$ and d close to 1, but its behavior deteriorates for

other values. By contrast, the variant guided by our theoretical model shows very good anytime behavior on all the experiments.

To conclude, we note that these results are coherent for instances with different problem sizes, different parameters, and different reference points. Moreover, we expect the results to generalize for any problem where the scaled non-dominated set can be approximated by the quadrant of a superellipse.

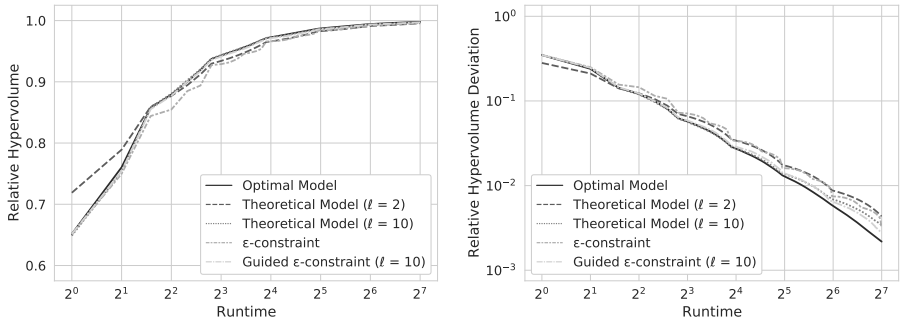
6 Conclusion

In this work, we presented a simple theoretical model to characterize the trade-off between runtime and hypervolume in the context of anytime algorithms for bi-objective optimization. The experimental results indicate that this theoretical model can finely approximate the anytime behavior of an optimal model with complete knowledge of the non-dominated set, especially when the number of linear segments is larger. We also show a potential application of our model, which is to guide scalarization techniques in order to improve their anytime behavior. In particular, we present an improved variant of an ε-constraint algorithm guided by our model that shows a particularly good anytime behavior.

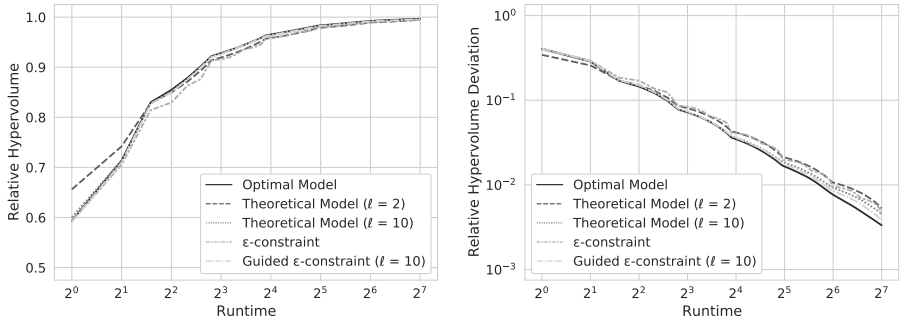
Another interesting application of our model is on algorithm survival analysis [8]. Based on our model it is possible to detect if an algorithm that follows a similar behavior is taking too long to find each efficient solution, and whether or not it will achieve some desirable quality within a desired time budget. This may justify a restart or a switch to a different search strategy. Moreover, our theoretical model can be used to define measurable metrics of comparison for the anytime behavior of techniques that collect efficient solutions iteratively.

As for future research, one possible direction is to extend the model for approximations of the non-dominated set other than a quadrant of a superellipse. In practice, this can already be accomplished by our algorithmic model by giving it any set of non-dominated, possibly disconnected, linear segments. However, the main challenge is to define a good piecewise linear approximation to the non-dominated set. A related idea is to consider a feedback loop between our model and a scalarization technique that sequentially finds efficient solutions. In particular, if the set of linear segments corresponds to an approximation of the non-dominated set, and throughout the execution of the scalarization technique we find the actual non-dominated points, then we can insert these points back into the model to increase the quality of the approximation in an online fashion. Lastly, our work has focused on the bi-objective case. Thus, a clear direction for future research is to extend it for more objectives. A model for more than two objective functions seems more challenging in part due to the splitting and tracking of the uncovered regions after a point with maximal hypervolume contribution is found.

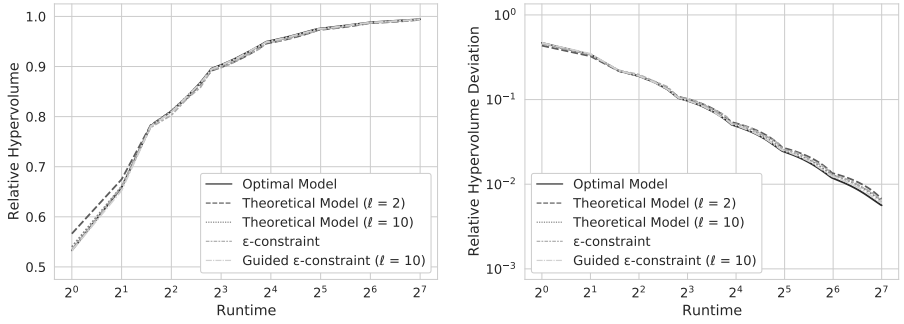
Acknowledgements The authors would like to acknowledge the anonymous reviewers whose comments contributed to improve the quality of the paper. This work was partially supported by COST Action CA15140 (STSM n. 39531), and by the PICS project MOCO-SEARCH co-funded by the French National Center for Scientific Research (CNRS) and the Portuguese Foundation for Science and Technology (FCT). The first author acknowledges the FCT for Ph.D. studentship SFRH/BD/132275/2017, co-funded by the European Social Fund and by the State Budget of the Portuguese Ministry of Education and Science. This work was partially funded by national funds through the FCT - Foundation for Science and Technology, I.P. within



(a) $\rho = -0.8$, $r = (0, 0)$, and $d \approx 2.099$

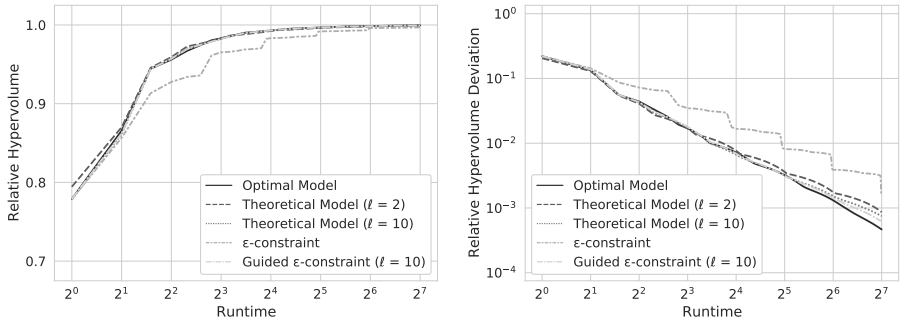


(b) $\rho = 0.0$, $r = (0, 0)$, and $d \approx 1.644$

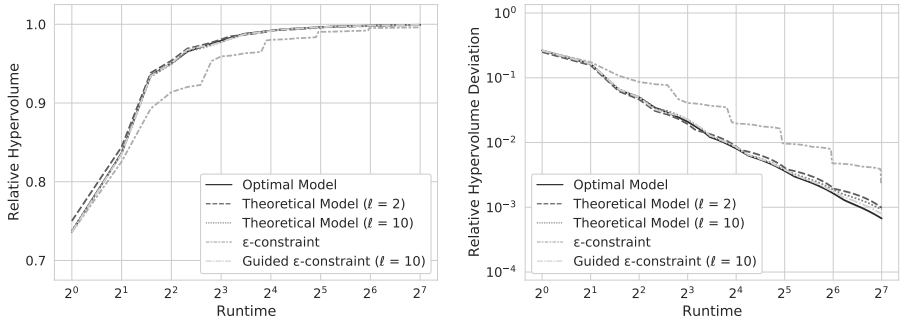


(c) $\rho = 0.8$, $r = (0, 0)$, and $d \approx 1.219$

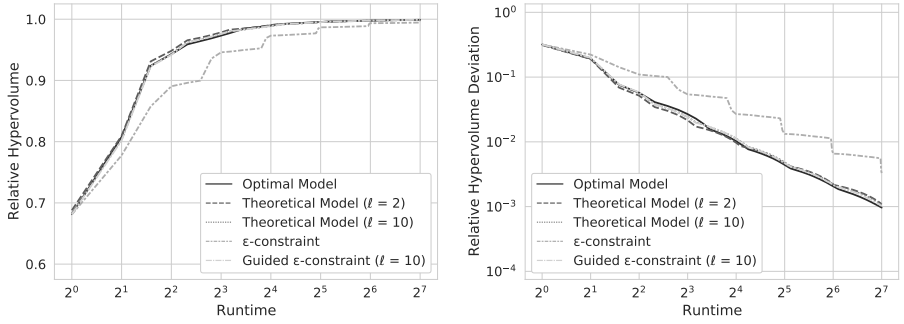
Fig. 7 Results for the UBKP with $n = 100$, reference point $r = (0, 0)$, and varying correlation $\rho \in \{-0.8, 0.0, 0.8\}$.



(a) $\rho = -0.8$, $r = (-1, -1)$, and $d \approx 2.099$

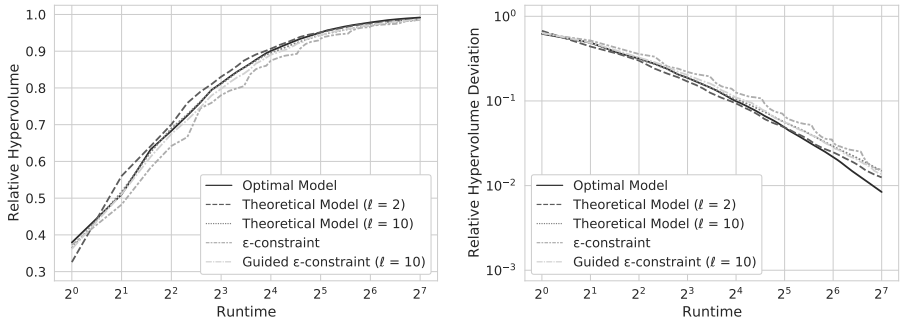


(b) $\rho = 0.0$, $r = (-1, -1)$, and $d \approx 1.644$

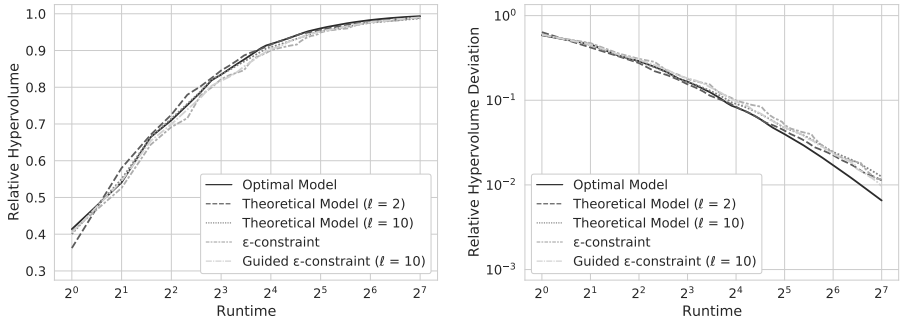


(c) $\rho = 0.8$, $r = (-1, -1)$, and $d \approx 1.219$

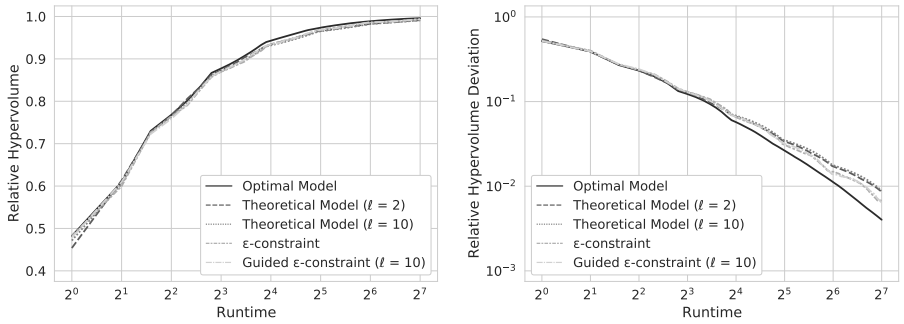
Fig. 8 Results for the UBKP with $n = 100$, reference point $r = (-1, -1)$, and varying correlation $\rho \in \{-0.8, 0.0, 0.8\}$.



(a) $s = 1.6$, $r = (0, 0)$, and $d \approx 0.476$

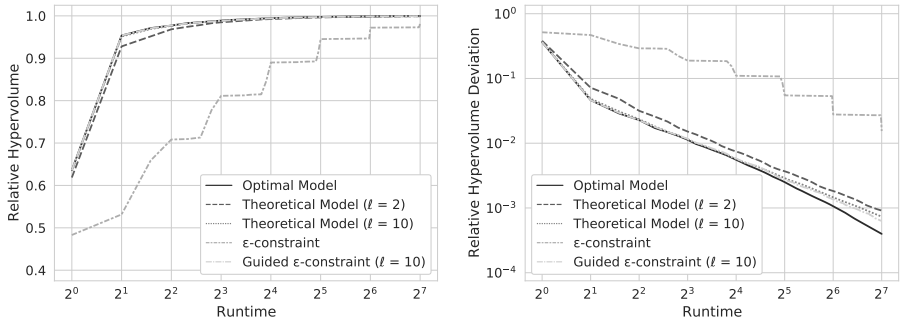


(b) $s = 2.0$, $r = (0, 0)$, and $d \approx 0.592$

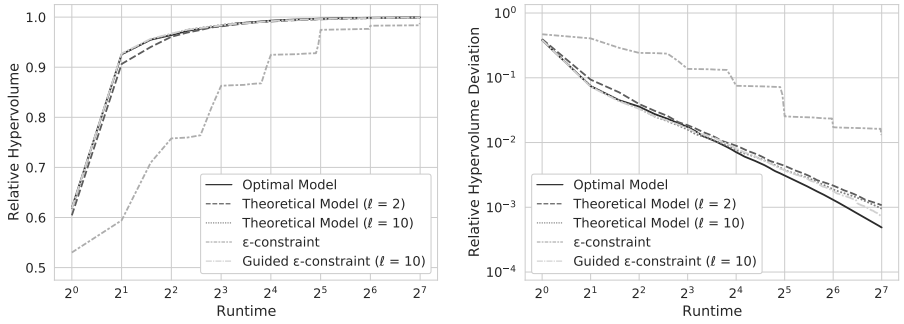


(c) $s = 3.0$, $r = (0, 0)$, and $d \approx 0.866$

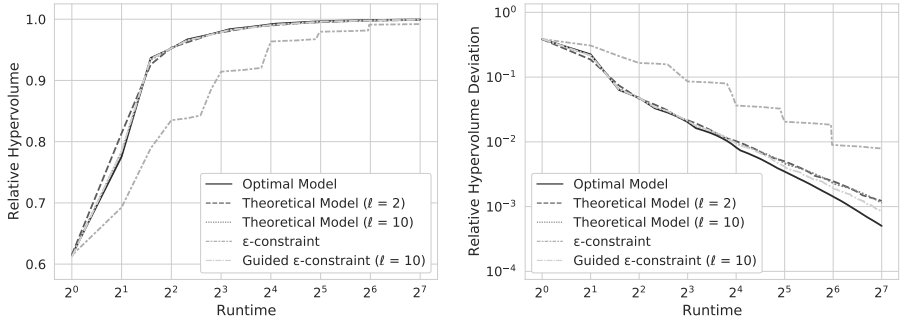
Fig. 9 Results for the CCUBKP with $n = 2000$, capacity constraint $c = 1$, reference point $r = (0, 0)$, and varying shape parameter $s \in \{1.6, 2.0, 3.0\}$.



(a) $s = 1.6$, $r = (-1, -1)$, and $d \approx 0.476$



(b) $s = 2.0$, $r = (-1, -1)$, and $d \approx 0.592$



(c) $s = 3.0$, $r = (-1, -1)$, and $d \approx 0.866$

Fig. 10 Results for the CCUBKP with $n = 2000$, capacity constraint $c = 1$, reference point $r = (-1, -1)$, and varying shape parameter $s \in \{1.6, 2.0, 3.0\}$.

the scope of the project CISUC - UID/CEC/00326/2020 and by the European Social Fund, through the Regional Operational Program Centro 2020.

References

1. Boddy, M., Dean, T.L.: Deliberation scheduling for problem solving in time-constrained environments. Artificial Intelligence **67**(2), 245–285 (1994). DOI 10.1016/0004-3702(94)90054-X
2. Braden, B.: The surveyor's area formula. The College Mathematics Journal **17**(4), 326–337 (1986). DOI 10.1080/07468342.1986.11972974
3. Dean, T., Boddy, M.: An analysis of time-dependent planning. In: Proceedings of the Seventh AAAI National Conference on Artificial Intelligence, AAAI'88, pp. 49–54. AAAI Press (1988)
4. Dubois-Lacoste, J., López-Ibáñez, M., Stützle, T.: Anytime Pareto local search. European Journal of Operational Research **243**(2), 369–385 (2015). DOI 10.1016/j.ejor.2014.10.062
5. Ehrgott, M.: Multicriteria Optimization, 2nd edn. Springer, Berlin, Heidelberg (2005). DOI 10.1007/3-540-27659-9
6. Emmerich, M.T.M., Deutz, A.H.: Test problems based on Lamé superspheres. In: Evolutionary Multi-Criterion Optimization, EMO 2007, pp. 922–936. Springer, Berlin, Heidelberg (2007). DOI 10.1007/978-3-540-70928-2_68
7. Finkelstein, L., Markovitch, S.: Optimal schedules for monitoring anytime algorithms. Artificial Intelligence **126**(1–2), 63–108 (2001). DOI 10.1016/S0004-3702(00)00072-2
8. Gagliolo, M., Legrand, C.: Algorithm survival analysis. In: Experimental Methods for the Analysis of Optimization Algorithms, pp. 161–184. Springer, Berlin, Heidelberg (2010). DOI 10.1007/978-3-642-02538-9_7
9. Hansen, E.A., Zilberstein, S.: Monitoring anytime algorithms. SIGART Bulletin **7**(2), 28–33 (1996). DOI 10.1145/242587.242593
10. Hansen, E.A., Zilberstein, S.: Monitoring and control of anytime algorithms: A dynamic programming approach. Artificial Intelligence **126**(1–2), 139–157 (2001). DOI 10.1016/S0004-3702(00)00068-0
11. Horvitz, E.J.: Reasoning about beliefs and actions under computational resource constraints. In: Proceedings of the Third Conference on Uncertainty in Artificial Intelligence, UAI-87, pp. 429–447. AUAI Press, Corvallis, Oregon (1987)
12. Jesus, A.D.: moco_abm v0.2.0 (2019). DOI 10.5281/zenodo.3548869
13. Jesus, A.D., Paquete, L., Figueira, J.R.: Finding representations for an unconstrained bi-objective combinatorial optimization problem. Optimization Letters **12**(2), 321–334 (2018). DOI 10.1007/s11590-017-1129-6
14. Knowles, J., Corne, D.: On metrics for comparing nondominated sets. In: Proceedings of the 2002 Congress on Evolutionary Computation, *CEC'02*, vol. 1, pp. 711–716. IEEE (2002). DOI 10.1109/CEC.2002.1007013
15. Kuhn, T., Fonseca, C.M., Paquete, L., Ruzika, S., Duarte, M.M., Figueira, J.R.: Hypervolume subset selection in two dimensions: Formulations and algorithms. Evolutionary Computation **24**(3), 411–425 (2015). DOI 10.1162/EVCO_a_00157
16. López-Ibáñez, M., Dubois-Lacoste, J., Stützle, T., Birattari, M.: The irace package: Iterated racing for automatic algorithm configuration. Operations Research Perspectives **3**, 43–58 (2016). DOI 10.1016/j.orp.2016.09.002
17. López-Ibáñez, M., Liao, T., Stützle, T.: On the anytime behavior of IPOP-CMA-ES. In: Parallel Problem Solving from Nature - PPSN XII, PPSN 2012, pp. 357–366. Springer, Berlin, Heidelberg (2012). DOI 10.1007/978-3-642-32937-1_36
18. López-Ibáñez, M., Stützle, T.: Automatically improving the anytime behaviour of optimisation algorithms. European Journal of Operational Research **235**(3), 569–582 (2014). DOI 10.1016/j.ejor.2013.10.043
19. Loudni, S., Boizumault, P.: Combining VNS with constraint programming for solving anytime optimization problems. European Journal of Operational Research **191**(3), 705–735 (2008). DOI 10.1016/j.ejor.2006.12.062
20. Makhorin, A.: GNU Linear Programming Kit - v4.65 (2018). URL `https://www.gnu.org/software/glpk/`
21. Nemhauser, G.L., Ullmann, Z.: Discrete dynamic programming and capital allocation. Management Science **15**(9), 494–505 (1969). DOI 10.1287/mnsc.15.9.494

22. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions. Mathematical Programming **14**(1), 265–294 (1978). DOI 10.1007/BF01588971
23. Pilu, M., Fisher, R.B.: Equal-distance sampling of superellipse models. In: Procedings of the British Machine Vision Conference, BMVC 1995, pp. 257–266. BMVA Press (1995). DOI 10.5244/C.9.26
24. Radulescu, A., López-Ibáñez, M., Stützle, T.: Automatically improving the anytime behaviour of multiobjective evolutionary algorithms. In: Evolutionary Multi-Criterion Optimization, EMO 2013, pp. 825–840. Springer, Berlin, Heidelberg (2013). DOI 10.1007/978-3-642-37140-0_61
25. Sayın, S.: Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. Mathematical Programming **87**(3), 543–560 (2000). DOI 10.1007/s101070050011
26. Ulrich, T., Thiele, L.: Bounding the effectiveness of hypervolume-based $(\mu + \lambda)$-archiving algorithms. In: Learning and Intelligent Optimization, LION 2012, pp. 235–249. Springer, Berlin, Heidelberg (2012). DOI 10.1007/978-3-642-34413-8_17
27. Vaz, D., Paquete, L., Fonseca, C.M., Klamroth, K., Stiglmayr, M.: Representation of the non-dominated set in biobjective discrete optimization. Computers & Operations Research **63**, 172–186 (2015). DOI 10.1016/j.cor.2015.05.003
28. Verel, S., Liefooghe, A., Jourdan, L., Dhaenens, C.: Analyzing the effect of objective correlation on the efficient set of mnk-landscapes. In: Learning and Intelligent Optimization, LION 2011, pp. 116–130. Springer, Berlin, Heidelberg (2011). DOI 10.1007/978-3-642-25566-3_9
29. Yildiz, H., Suri, S.: On Klee's measure problem for grounded boxes. In: Proceedings of the Twenty-Eigth Annual Symposium on Computational Geometry, SoCG '12, pp. 111–120. Association for Computing Machinery, New York, NY, USA (2012). DOI 10.1145/2261250.2261267
30. Zilberstein, S.: Using anytime algorithms in intelligent systems. AI Magazine **17**(3), 73–83 (1996). DOI 10.1609/aimag.v17i3.1232
31. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms — A comparative case study. In: Parallel Problem Solving from Nature — PPSN V, PPSN 1998, pp. 292–301. Springer, Berlin, Heidelberg (1998). DOI 10.1007/BFb0056872
32. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation **7**(2), 117–132 (2003). DOI 10.1109/TEVC.2003.810758