Bergische Universität Wuppertal

Fakultät für Mathematik und Naturwissenschaften

Institute of Mathematical Modelling, Analysis and Computational Mathematics (IMACM)

Preprint BUW-IMACM 20/38

Luís Paquete, Britta Schulze and Michael Stiglmayr

Computing Representations using Hypervolume Scalarizations

September 23, 2020

http://www.imacm.uni-wuppertal.de

Computing Representations using Hypervolume Scalarizations

Luís Paquete

University of Coimbra, CISUC, Department of Informatics Engineering, Portugal

Britta Schulze, Michael Stiglmayr

School of Mathematics and Natural Sciences, University of Wuppertal, Germany

Abstract

In this article, we use the hypervolume indicator as a scalarizing function for multiobjective combinatorial optimization problems. In particular, we describe a generic solution approach that determines the nondominated set of a multiobjective optimization problem by solving a sequence of hypervolume scalarizations with appropriate choices of the reference point. Moreover, this solution technique can also provide a compact representation of the efficient set that is a $(1-1/e)$-approximation to the optimal representation in terms of the hypervolume in an a priori manner. We evaluate these concepts on a particular variant of the biobjective knapsack problem and present numerical results.

Keywords: hypervolume scalarization, representation, multiobjective discrete optimization, greedy approximation

1. Introduction

The *hypervolume indicator* measures the m-dimensional volume of the union of axis-parallel boxes each spanned by a nondominated point and the predefined reference point (Zitzler and Thiele, 1998). This indicator has shown to have interesting properties such as strict monotonicity with respect to the dominance relations, submodularity and others. It has also gained popularity as a performance assessment method of heuristics for multiobjective optimization problems as well as selection criterion and archiving strategy for multiobjective evolutionary algorithms (Beume et al., 2007; Bader and Zitzler, 2011). In recent years, several algorithmic studies have been conducted for the efficient computation of this indicator.

In this article, we investigate the use of the hypervolume indicator as a scalarization method for finding the complete efficient set of multiobjective discrete optimization problems. We establish equivalence results for the optimality of the *hypervolume scalarized* problem and the efficiency of its multiobjective counterpart. One main advantage of hypervolume scalarizations is that no convexity assumptions are required, but, on the other hand, they lead to particular quadratic formulations. We show an application of this scalarization technique to a biobjective cardinality constrained knapsack problem. Its hypervolume scalarization leads to a quadratic knapsack formulation, known as the *rectangular knapsack problem*, which can be linearised and for which an efficient approximation algorithm with quality guarantee is known (Schulze et al., 2020).

Our results also suggest that it is possible to find the efficient set of a multiobjective discrete optimization problem by solving a finite sequence of hypervolume scalarizations for appropriate choices of the reference point. We provide such a solution approach that iteratively bisects the objective space into two disjoint regions, each of which is bounded by a reference point and corresponds to a scalarized subproblem to be solved. One important aspect of this approach is that it is able to derive a succint representation of the efficent set, if terminated early. In recent years, there has been some effort on the development of solution techniques that return representations of the efficient set with certain bounds on the representation quality (Sayın, 2000, 2003; Hamacher et al., 2007; Sylva and Crema, 2007; Eusébio et al., 2014; Jesus et al., 2018; Kirlik and Sayın, 2018; Kidd et al., 2020). Optimal representations seem to be only possible to obtain if the efficient set is known in advance. For those cases, several exact procedures that allow to extract an optimal representation from the efficient set have been proposed in the literature (Auger et al., 2009; Vaz et al., 2015; Kuhn et al., 2016; Bringmann et al., 2017; Gomes et al., 2018). Noteworthly, our solution technique provides a representation that is an $(1 - 1/e)$-approximation to the optimal representation in terms of the hypervolume indicator. It is possible to achieve this ratio by using a greedy criterion on the next scalarized subproblem to be solved. To the best of our knowledge, this is the first solution approach that can provide a representation with a provable guarantee with respect to an optimal representation.

Our experimental analysis on the biobjective cardinality constrained knapsack problem indicates that an ILP solver may find difficulties on solving hypervolume scalarizations. For this reason, we propose a branch-and-bound method that combines pruning conditions based on the combinatorial structure of the scalarized problem and on the dominance structure of the biobjective optimization problem.

The article is organized as follows. Related work and main concepts on multiobjective optimization are presented in Section 2. In Section 3, the application of the hypervolume indicator as quality measure in subset selection and as scalarization method is motivated and central results are shown. Section 4 presents our dichotomic solution approach to solve multiobjective discrete optimization problems based on hypervolume scalarizations. Section 5 describes a case study on the cardinality constrained knapsack problem and Section 6 reports an experimental analysis of our approach. Finally, Section 7 concludes and discusses further work.

2. Basic concepts of multiobjective optimization and related work

In the following, we introduce basic concepts in multiobjective optimization and representation of the efficient set, and review the main contributions in the field with respect to these topics. A self-contained introduction to multiobjective optimization is given in Ehrgott (2005).

2.1. Multiobjective optimization

In multiobjective optimization, we consider the optimization of m objective functions $f_i \colon \mathcal{X} \longrightarrow \mathbb{R}$, $i = 1, \ldots, m$, over the set of feasible solutions $\mathcal{X} \subseteq \mathbb{R}^n$. Without loss of generality, we assume maximization in the following.

$$\begin{aligned} \text{vmax} \quad & f(x) = (f_1(x), \ldots, f_m(x)) \\ \text{s.t.} \quad & x \in \mathcal{X} \end{aligned} \tag{MOP}$$

In general, there exists no solution that is optimal for all functions f_i at once. It is well known that there is no canonical order of R^m for $m \geq 2$. In this article we rely on the Pareto concept of optimality, which is based on the component-wise order relations in $\mathbb{R}^m$. Let $p, q \in \mathbb{R}^m$:

$$\begin{aligned} p \geqq q &\iff p_i \geq q_i \ \text{ for } i = 1, \ldots, m \\ p \geq q &\iff p \geqq q \ \text{ and } p \neq q \\ p > q &\iff p_i > q_i \ \text{ for } i = 1, \ldots, m \end{aligned}$$

Given a set $\mathcal{P} \subset \mathbb{R}^m$, a point $p \in \mathcal{P}$ is called *nondominated point* if there exists no other point $q \in \mathcal{P}$ such that $q \geq p$.
Let $\mathcal{Y} = f(\mathcal{X})$ denote the image of the feasible set. Furthermore, let $\mathcal{Y}_N \subseteq \mathcal{Y}$ denote the set of all nondominated points (the nondominated set) in the objective space and $\mathcal{X}_E$ denote the set of all efficient solutions (the efficient set), i.e., solutions whose image belongs to $\mathcal{Y}_N$.
In Serafini (1987) nine different solution concepts for multiobjective optimization problems are discussed. In this article, we mostly consider the computation of $\mathcal{Y}_N$ or a minimal complete set, i.e., the set of nondominated points $\mathcal{Y}_N$ and one efficient solution $x \in \mathcal{X}_E$ for each nondominated point $y = f(x)$. However, most results also transfer to the computation of X_E.
A common general procedure to find $\mathcal{Y}_N$, or a subset of it, is to use a scalarizing function such that optimal solutions for this function are also efficient solutions to the multiobjective optimization problem (correctness). On the other hand, by varying the parameters of a scalarizing function the set of optimal solutions should contain any efficient solution (completeness). The solution techniques based on scalarizations vary the parameters of the scalarizing function in order to obtain different efficient solutions and nondominated points, respectively. The efficiency of such solution techniques depends on the properties of the underlying scalarization function. Examples of well-known scalarization-based solution techniques are dichotomic search using weighted sum (Aneja and Nair, 1979) and the ϵ-constraint method (Haimes et al., 1971). Note that the weighted

sum approach not necessarily satisfies the completeness property for non-convex problems.

2.2. Representation of the efficient set

Since the cardinality of $\mathcal{Y}_N$ often grows exponentially with the input size, computing a polynomial size subset $S \subset \mathcal{Y}_N$ as a representation may be preferable (Sayın, 2000). The representation problem consists of the following problem

$$S^* := \arg\max_{S \subset \mathcal{Y}_N} R(S)$$

where function R measures a representation quality that may relate to the preferences of a decision maker such as small cardinality, large distance between adjacent points in the representation (uniformity), closeness to other solutions in $\mathcal{Y}_N$ (coverage), some combination thereof, or some other correlated measure (Faulkenberg and Wiecek, 2010). In the following, we only consider methods that report representations for multiobjective discrete optimization problems containing only elements of $\mathcal{Y}_N$ and that are able to provide some representation quality guarantee.

A representation can be obtained directly from the set $\mathcal{Y}_N$ (*a posteriori*) or iteratively generated without the knowledge of $\mathcal{Y}_N$. Procedures that compute optimal representations for the former case are discussed in Vaz et al. (2015); Kuhn et al. (2016); Bringmann et al. (2017); Gomes et al. (2018). In the latter case, the approaches are based on the iterative solution of constrained scalarized problems such as the *box-method* in Hamacher et al. (2007); Kuhn et al. (2016); Boland et al. (2015) and other variants as described in Sylva and Crema (2007); Eusébio et al. (2014); Kidd et al. (2020). Although not able to obtain an optimal representation in general, these approaches provide an approximation that achieves a certain representation quality defined a priori, such as cardinality, uniformity and/or coverage.

3. Hypervolume indicator and scalarization

3.1. Hypervolume indicator and subset selection

The hypervolume indicator was initially proposed to evaluate the performance of multiobjective evolutionary optimization (EMO) algorithms in Zitzler and Thiele (1998) and became a versatile tool in multiobjective optimization. It is used, e.g., as fitness function within EMO algorithms or as a quality measure for representations/approximations of the nondominated set. We refer to the survey Guerreiro et al. (2020) for a detailed description of the hypervolume indicator, see also Zitzler et al. (2003); Auger et al. (2009, 2012); Ulrich and Thiele (2012).

Definition 3.1 (Hypervolume indicator)**.** *The hypervolume indicator $H(S)$ of a subset of images of feasible points $S \subset \mathcal{Y}$ measures the volume spanned between*

S and with a reference point $r \in \mathbb{R}^m$ (with $r \leqq p$ for all $p \in \mathcal{Y}$), i.e.:

$$H(S) := \mathrm{vol}\Big(\big(S - \mathbb{R}^m_+\big) \cap \big(\{r\} + \mathbb{R}^m_+\big)\Big)$$

The hypervolume indicator has interesting properties. It is scaling invariant and it is a non-decreasing submodular function. Moreover, the nondominated set has maximal hypervolume value among all subsets of feasible points. Several algorithms have been proposed to compute the hypervolume indicator.
Since we are considering the iterative computation of representative points, we need to quantify the individual hypervolume contribution of a point p given a set S of already selected points. This incremental hypervolume is denoted as *hypervolume indicator contribution* of p.

Definition 3.2 (Hypervolume Contribution)**.** *The hypervolume indicator contribution $H(p, S)$ of a point $p \in \mathcal{Y}$ with respect to a set $S \subset \mathcal{Y}$ given a reference point $r \in \mathbb{R}^m$ (with $r \leqq p$ for all $p \in \mathcal{Y}$) is defined as*

$$H(p, S) := H(S \cup \{p\}) - H(S)$$

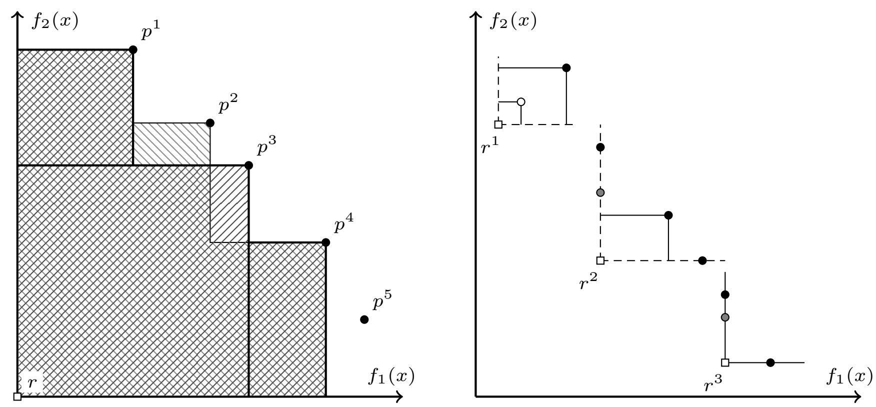
The representation problem that we address in this article consists of identifying a subset $S := \{p^1, \ldots, p^J\} \subset \mathcal{Y}$ with a given fixed cardinality J, that maximizes the hypervolume. In the following we will denote this hypervolume maximizing subset by $S^J_O \subseteq \mathcal{Y}_N$, i. e.:

$$S^J_O := \operatorname*{arg\,max}_{\substack{S \subseteq \mathcal{Y}_N \\ |S| = J}} H(S)$$

Even if the set of nondominated points $\mathcal{Y}_N$ is known, the problem of finding a hypervolume maximizing set S^J_O is NP-hard for $m \geq 3$ (Bringmann et al., 2017). Several exact algorithms have been proposed to solve this problem for $m = 2$ (Auger et al., 2009; Bringmann et al., 2014; Kuhn et al., 2016; Groz and Maniu, 2019), $m = 3$ (Bringmann et al., 2017) and for any dimension (Bringmann and Friedrich, 2010; Gomes et al., 2018; Groz and Maniu, 2019).
Furthermore, there is a considerable amount of heuristics and approximation algorithms for this problem (Bradstreet et al., 2007; Friedrich and Neumann, 2015; Basseur et al., 2016; Guerreiro et al., 2016; Bringmann et al., 2017). Of particular interest for our results is the greedy approach described in Bradstreet et al. (2007) and later improved in Guerreiro et al. (2016). The greedy algorithm chooses, at each iteration, the point that maximizes the hypervolume contribution with respect to the current point set, that is, in the ℓ-th iteration, a point $S_G(\ell)$, $\ell = 1, \ldots, J$ is selected according to

$$S_G(\ell) := \operatorname*{arg\,max}_{p \in \mathcal{Y}_N} H\Big(p, S^{\ell-1}_G\Big),$$

where $S^{\ell-1}_G = \{S_G(1), \ldots, S_G(\ell - 1)\}$. $S^J_G = \big\{S_G(1), \ldots, S_G(J)\big\} \subset \mathcal{Y}_N$ denotes the set of J points selected by the greedy algorithm. Note that the greedy



(a) Comparison of the hypervolume of the greedy solution $S_G^2 = \{p^3, p^1, p^4\}$ (▨) with $S_G(1) = \{p^3\}$ and $S_G(2) = \{p^1\}$ and that of the optimal solution $S_O^2 = \{p^1, p^2, p^4\}$ (▧)

(b) Correctness: Every optimal solution to (HS(MOP)) is weakly efficient for MOP (see r^1); if the corresponding point dominates the reference point, the solution is efficient (see r^2); if the optimal solution is weakly efficient but not efficient, the optimal objective function value of (HS(MOP)) is equal to zero (see r^3).

algorithm solves the problem to optimality for $J = 1$, i.e., $S_G(1) = S_O^1$. As an example, see Figure 1a for a comparison of optimal and greedy solution of a biobjective representation problem with five nondominated points and $J = 3$. Given that the hypervolume indicator is a non-decreasing submodular function, the corresponding representation problem consists of maximizing a submodular function with a cardinality constraint (Friedrich and Neumann, 2015). Given the approximation result of greedy algorithms for maximization problems with submodular non-decreasing functions (Nemhauser et al., 1978), the greedy algorithm described above has an approximation ratio of $1 - 1/e$ from the optimal representation in terms of the hypervolume indicator (Guerreiro et al., 2016). However, numerical results show that the performance of the greedy algorithm significantly outperforms the theoretical bound (Guerreiro et al., 2016; Torrico et al., 2020).

3.2. Hypervolume Scalarization

Since the hypervolume indicator can furthermore be used as a scalarizing function, as described in Hernandez et al. (2018); Schultes et al. (2020); Touré et al. (2019); Yang et al. (2019), one can, in principle, solve the hypervolume representation problem to optimality without pre-computing all nondominated points.

Let $f(X) := \{f(x) \colon x \in X\}$ with $X \subset \mathcal{X}$.

$$\begin{aligned} \max \quad & H(f(X)) \\ \text{s.t.} \quad & |X| \leq J \\ & X \subset \mathcal{X} \end{aligned} \tag{1}$$

By solving problem 1 it is possible to obtain a set of efficient solutions such that the corresponding subset of nondominated points maximize the hypervolume among all sets of at most J nondominated points (given that $J \leq |\mathcal{Y}|$).
Assuming the contrary, let X^* be an optimal solution of 1 and let a $\bar{x} \in X^*$ be a dominated solution of MOP, then there exists $x' \in \mathcal{X}$ such that $f(x') \geq f(\bar{x})$. Consequently, the hypervolume contribution of x' is larger than that of $\bar{x}$, i.e., $H(f(x'), f(X^* \setminus \{\bar{x}\})) \geq H(f(X^*))$, which contradicts the optimality of X^*.
Instead of optimizing the hypervolume of the representative set of points all-in-one, we consider the iterative scalarization of the multiobjective optimization problem (MOP) by maximizing the hypervolume contribution $H(f(x), Y)$ analogously to the greedy selection strategy. The hypervolume contribution can be determined by shifting the reference point $r \in \mathbb{R}^m$ accordingly.

$$\begin{aligned} \max \quad & \prod_{i=1}^{m} f_i(x) - r_i \\ \text{s.t.} \quad & f_i(x) \geq r_i \quad \forall i = 1, \ldots, m \\ & x \in \mathcal{X} \end{aligned} \tag{HS(MOP)}$$

Note that problem HS(MOP) is a proper scalarization, in the sense that every optimal solution of the scalarized problem HS(MOP) is weakly effcient for MOP (correctness), see also Figure 1b, and every efficient solution of MOP can be determined as an optimal solution of a corresponding scalarized problem HS(MOP) (completeness).

Theorem 3.3 (Correctness)**.** *Every optimal solution $\bar{x}$ of the hypervolume scalarized problem HS(MOP) is weakly efficient for the multiobjective optimization problem MOP. Moreover, if $f(\bar{x}) > r$, then $\bar{x}$ is efficient for MOP.*

Proof. Let $\bar{x}$ be an optimal solution of HS(MOP). Assume that $\bar{x}$ is not weakly efficient for MOP, i. e., there exists an $x' \in \mathcal{X}$ such that $f(x') > f(\bar{x})$. Since $f_i(x) - r_i \geq 0$ for every feasible solution of HS(MOP), it follows that

$$\prod_{i=1}^{m} f_i(x') - r_i > \prod_{i=1}^{m} f_i(\bar{x}) - r_i$$

which is a contradiction to the optimality of $\bar{x}$ for HS(MOP). Furthermore, assume that $f(\bar{x}) > r$ and $\bar{x}$ is not efficient, i. e., there exists $x' \in \mathcal{X}$ such that $f_i(x') - r_i \geq f_i(\bar{x}) - r_i$ for all $i = 1, \ldots, m$ and $f_j(x') - r_j > f_j(\bar{x}) - r_j$ for at least on $j \in \{1, \ldots, m\}$. With $f(\bar{x}) > r$, we obtain $\prod_{i=1}^{m} f_i(x') - r_i > \prod_{i=1}^{m} f_i(\bar{x}) - r_i$, contradicting the optimality of $\bar{x}$. □

An alternative proof can be given by showing that the hypervolume scalarization is a strictly increasing achievement scalarizing function. Furthermore, one can verify that it is strongly increasing if $f(\bar{x}) > r$. Then, the theory of achievement scalarizing functions yields the result, see e. g., Wierzbicki (1986a,b); Miettinen (1998).

Theorem 3.4 (Completeness)**.** *All efficient solutions of MOP can be found using the hypervolume scalarization.*

Proof. Every solution $\bar{x} \in \mathcal{X}_E$ of MOP is an optimal solution to the hypervolume scalarized problem HS(MOP) using $r = f(\bar{x})$ as reference point. □

This reference point selection strategy is not of practical use, since it requires the a priori knowledge of all nondominated points. In Schultes et al. (2020), the *Pareto front generating reference sets* are introduced for continuous multi-objective optimization problems, i.e., a set of reference points such that every nondominated point can be achieved by a hypervolume scalarized problem with respect to a reference point in this set.
Unlike the weighted sum scalarization, the hypervolume scalarization does not rely on convexity assumptions and can thus be applied in the context of discrete optimization problems without being restricted to supported efficient solutions.
In the following section, we describe a method that allows to find the nondominated set $\mathcal{Y}_N$ for a given MOP with $m = 2$ by solving a sequence of hypervolume scalarizations.

4. Hypervolume dichotomic scheme

In the following, we describe a dichotomic scheme to solve MOP for $m = 2$, which is shown in the pseudocode of Algorithm 1. The approach maintains a sequence of nondominated points, ordered with respect to the first objective, which is stored in S. Each point s in S, except for the first and the last one, gives rise to two search regions (Dächert et al., 2017). They are defined by the coordinates of s and of its predecessor and successor in S, respectively, which are called *local lower bounds*. These local lower bounds are used as reference points in the algorithm and, consequently, give rise to two new hypervolume scalarized problems. In this way, a Pareto front generating reference set is iteratively determined.
Let $(\overleftarrow{s_1}, \overleftarrow{s_2}) := pred(s, S)$ and $(\overrightarrow{s_1}, \overrightarrow{s_2}) := succ(s, S)$ be the points in S that are immediately before and after point s, respectively. To ensure that a successor and predecessor of a point always exist in the main loop, we consider that two border points, $(r_1, +\infty)$ and $(+\infty, r_2)$, where (r_1, r_2) is the initial reference point, are inserted into S in a preprocessing step. Then, the two reference points that define the next two hypervolume scalarized problems are $(r_1, r_2) := (s_1, \overrightarrow{s_2})$ and $(r_1, r_2) := (\overleftarrow{s_1}, s_2)$.

Algorithm 1 Hypervolume Dichotomic Scheme

Require: $(r_1, r_2), J$
$S = P = \emptyset$
$insert((r_1, +\infty), S)$
$insert((+\infty, r_2), S)$
$s \leftarrow solve(r_1, r_2)$
$insert(s, S)$
while $|S| - 2 < J$ **do**
    $(\overrightarrow{s_1}, \overrightarrow{s_2}) \leftarrow succ(s, S)$
    $(r_1, r_2) \leftarrow (s_1, \overrightarrow{s_2})$
    $p \leftarrow solve(r_1, r_2)$
    if $H(p, S) > 0$ **then**
        $enqueue(p, P)$
    $(\overleftarrow{s_1}, \overleftarrow{s_2}) \leftarrow pred(s, S)$
    $(r_1, r_2) \leftarrow (\overleftarrow{s_1}, s_2)$
    $p \leftarrow solve(r_1, r_2)$
    if $H(p, S) > 0$ **then**
        $enqueue(p, P)$
    $s \leftarrow dequeue(P)$
    $insert(s, S)$
return S

A nondominated point p generated from each of the two subproblems is stored in a data structure P if it has a positive hypervolume contribution value with respect to the current set of nondominated points stored in S. At the end of each iteration, a point s is removed from P and inserted into S. Then, if the number of elements in S, less the two initial border points, is equal to the desired size of the representation set, J, the algorithm terminates and reports the contents of S. Note that $J-1$ iterations of the while loop of Algorithm 1 are required, which corresponds to $2\,J-1$ hypervolume scalarized problems. The stopping condition can be changed to proceed until P is empty, which results in the computation of the complete nondominated set $\mathcal{Y}_N$.

In order to follow the greedy principle, P needs to be implemented as a priority queue defined with respect to the hypervolume contribution of its elements, that is, the next point to be dequeued is the one with the maximal hypervolume contribution with respect to S. Therefore, P stores a sequence of points with nonincreasing hypervolume contribution. In order to perform efficient insertions and queries, P can be implemented as a balanced binary tree.

It is clear that the efficiency of this dichotomic search depends on how efficient the scalarizations can be solved. In the following sections, we analyze this approach for a specific class of problems. An illustrative example of the algorithm is given in Section 5.3.

5. Case study: Cardinality constrained knapsack problem

In this section, we illustrate the application of hypervolume scalarization and our dichotomic scheme to biobjective knapsack problems with a cardinality constraint. This problem consists of two linear sum objective functions, which are to be maximized, and a cardinality constraint that allows only k items to be selected. The hypervolume scalarization of this problem for a reference point $r = (0,0)^\top$ corresponds to the *rectangular knapsack problem* for which a 4.5-approximation algorithm is known (Schulze et al., 2020). For general reference points, it has been shown that the scalarized problem is NP-hard.
The biobjective cardinality constrained knapsack problem can be formalized as follows.

Problem 5.1 (Biobjective cardinality constrained knapsack problem)**.**

$$\begin{aligned} \max \quad f(x) := & \left(\sum_{i=1}^{n} a_i\, x_i,\ \sum_{i=1}^{n} b_i\, x_i \right) \\ s.\,t. \quad & \sum_{i=1}^{n} x_i \leq k \\ & x_i \in \{0,1\} \qquad \forall j \in \{1,\dots,n\} \end{aligned} \tag{2oKP}$$

where $a, b \in \mathbb{R}^n_{\geq 0}$ *and* $k \in \mathbb{N}$, $k < n$.

We assume that all coefficients are non-negative and, therefore, all efficient solutions of 2oKP include exactly k items, i. e., the cardinality constraint is handled as an equality constraint.
The hypervolume scalarization leads to the following optimization problem with $r = (r_1, r_2)^\top$ as the reference point.

Problem 5.2 (Hypervolume scalarization of 2oKP)**.**

$$\begin{aligned} \max \quad h(x) := & \left(\sum_{i=1}^{n} a_i\, x_i \; - \; r_1 \right) \cdot \left(\sum_{i=1}^{n} b_i\, x_i \; - \; r_2 \right) \\ & = \sum_{i=1}^{n} \sum_{j=1}^{n} a_i b_j\, x_i x_j - \sum_{i=1}^{n} (r_2 a_i + r_1 b_i)\, x_i + r_1\, r_2 \\ s.\,t. \quad & \sum_{i=1}^{n} a_i\, x_i \geq r_1 \\ & \sum_{i=1}^{n} b_i\, x_i \geq r_2 \\ & \sum_{i=1}^{n} x_i = k \\ & x_i \in \{0,1\} \qquad \forall j \in \{1,\dots,n\} \end{aligned} \tag{HS(2oKP)}$$

In the following, we present two approaches to solve the hypervolume scalarized problem HS(2oKP): an integer linear programing formulation of the hypervolume scalarization that can be used within a generic ILP solver and a branch-and-bound approach that takes into account both the combinatorial structure of the scalarized problem and the dominance structure of the underlying biobjective optimization problem. We finish this section with an illustrative example presenting the workflow of Algorithm 1 to compute a representation of the nondominated set of a cardinality constrained biobjective knapsack problem.

5.1. An integer linear programing formulation

In order to transform the problem above into an integer linear program, we define $Q = a \cdot b^T$, i.e., $Q_{ij} = a_i \cdot b_j$ for all $i, j \in \{1, \dots, n\}$. The objective function is linearized by introducing n^2 new variables y_{ij}, $i, j \in \{1, \dots, n\}$, that attain value 1 if and only if $x_i = 1$ and $x_j = 1$, which is ensured by the following constraints (Watters, 1967)

$$\begin{aligned} y_{ij} &\leq y_{ii} \\ y_{ij} &\geq y_{ii} + y_{jj} - 1 \end{aligned}$$

Additional constraints are required to handle symmetry, i.e., $y_{ij} = y_{ji}$. Preliminary tests showed that further n redundant constraints may provide a tighter LP relaxation, namely

$$\sum_{\substack{i=1 \\ i \neq j}}^{n} y_{ij} \leq (k-1)\, y_{jj}, \quad \forall j \in \{1, \dots, n\}$$

which also have been used for the ILP formulation for a more general version of the quadratic knapsack problem (Caprara et al., 1999). We reach to the following ILP linearization with $O(n^2)$ constraints.

Problem 5.3 (Linearization)**.**

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{n}\sum_{j=1}^{n} Q_{ij}\, y_{ij} - \sum_{i=1}^{n}(r_2 a_i + r_1 b_i)\, y_{ii} + r_1\, r_2 \\
s.\,t. \quad & \sum_{i=1}^{n} a_i\, y_{ii} \geq r_1 \\
& \sum_{i=1}^{n} b_i\, y_{ii} \geq r_2 \\
& \sum_{i=1}^{n} y_{ii} = k \\
& \sum_{\substack{i=1 \\ i\neq j}}^{n} y_{ij} \leq (k-1)\, y_{jj} && \forall j \in \{1,\dots,n\} \\
& y_{ij} = y_{ji} && \forall i,j \in \{1,\dots,n\}, i<j \\
& y_{ij} \leq y_{ii} && \forall i,j \in \{1,\dots,n\}, i\neq j \\
& y_{ij} \geq y_{ii} + y_{jj} - 1 && \forall i,j \in \{1,\dots,n\}, i<j \\
& y_{ij} \in \{0,1\}
\end{aligned}
\tag{LIN}
$$

Equivalently, the objective function can also be formalized with a matrix $\widehat{Q}$ as follows

$$h(x) := \sum_{i=1}^{n}\sum_{j=1}^{n} \widehat{Q}_{ij}\, y_{ij} + k\, r_1\, r_2$$

such that

$$\widehat{Q}_{ij} = a_i b_j k - (r_2 a_i + r_1 b_j)$$

which stems from the fact that

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(r_2 a_i + r_1 b_j)\, y_{ij} = k \sum_{i=1}^{n}(r_2 a_i + r_1 b_i)\, y_{ii}$$

5.2. A combinatorial branch-and-bound algorithm

In the following, we introduce a combinatorial branch-and-bound algorithm to solve Problem HS(2oKP) as an alternative for ILP solvers to this problem. Rather than using LP relaxations to derive upper bounds, we consider two bounds of combinatorial nature that can be efficiently computed.

Without loss of generality, assume that all instances of Problem 2oKP are defined such that

$$a_1 \geq a_2 \geq \ldots \geq a_n$$

Let $\mathcal{S}_n$ denote the symmetric group of order n and $\pi \in \mathcal{S}_n$ denote a permutation of $\{1,\dots,n\}$. Consider π such that

$$b_{\pi(1)} \geq b_{\pi(2)} \geq \ldots \geq b_{\pi(n)}$$

Using the sorted coefficients a_i and $b_\pi(i)$, it is possible to derive the following upper bound $\mathcal{U}^A$ for Problem 2oKP

$$f(x) \leq \left(\sum_{i=1}^{k} a_i, \sum_{i=1}^{k} b_{\pi(i)} \right) := \mathcal{U}^A \tag{2}$$

for any $x \in \mathcal{X}_E$.
A related upper bound $\mathcal{U}^B$ for Problem HS(2oKP) can be derived as follows (Schulze et al., 2020).

$$h(x) \leq \left(\sum_{i=1}^{k} a_i - r_1 \right) \cdot \left(\sum_{i=1}^{k} b_{\pi(i)} - r_2 \right) := \mathcal{U}^B \tag{3}$$

We introduce the following definitions in the context of our branch-and-bound algortihm. Let $\bar{x} \in \{0,1\}^n$ such that $\bar{x}_j = 0$, for $j = \ell+1, \ldots, n$, and $\sum_{i=1}^{\ell} x_i \leq k$, and let

$$\bar{k} := \min \left\{ k - \sum_{i=1}^{\ell} x_i, n - l \right\}.$$

We define an upper bound $U^A(\bar{x})$ as follows

$$U^A(\bar{x}) := \left(\sum_{i=1}^{\ell} a_i \bar{x}_i + \sum_{j=\ell+1}^{\ell+\bar{k}} a_j, \sum_{i=1}^{\ell} b_i \bar{x}_i + \sum_{j \in J} b_{\pi(j)} \right) \tag{4}$$

where $J := \{j_1, \ldots, j_{\bar{k}}\} \subseteq \{\ell+1, \ldots, n\}$ for which it holds that

$$\pi(j_1) \ < \ \pi(j_2) \ < \ \ldots \ < \ \pi(j_{\bar{k}})$$

and

$$\pi(j_{\bar{k}}) \ < \ \pi(j) \qquad \forall j \in \{\ell+1, \ldots, n\} \setminus \{j_1, \ldots, j_{\bar{k}}\}.$$

Similarly, we define the following upper bound $U^B(\bar{x})$.

$$U^B(\bar{x}) := \left(\sum_{i=1}^{\ell} a_i \bar{x}_i + \sum_{j=\ell+1}^{\ell+\bar{k}} a_j - r_1 \right) \cdot \left(\sum_{i=1}^{\ell} b_i \bar{x}_i + \sum_{j \in J} b_{\pi(j)} - r_2 \right) \tag{5}$$

Both upper bounds $U^A(\bar{x})$ and $U^B(\bar{x})$ can be used for pruning incumbent solutions within the branch-and-bound framework. We introduce the following concepts that are required to specify the pruning conditions. A solution $\hat{x} \in \mathcal{X}$ is a *feasible extension* of $\bar{x}$ if and only if it is feasible for 2oKP and $\hat{x}_i = \bar{x}_i$ for $i = 1, \ldots, \ell$. Let $E^A(\bar{x})$ denote the set of all feasible extensions of $\bar{x}$ and let $E^B(\bar{x}) \subseteq E^A(\bar{x})$ denote the set of feasible extensions of $\bar{x}$ that are also feasible for HS(2oKP). Then, the following straightforward implications hold with respect to $\bar{x}$, given a feasible solution $x^* \in \mathcal{X}$:

C1) $U^A(\bar{x}) \leq f(x^*) \implies f(\hat{x}) \leq f(x^*)$, for $\hat{x} \in E^A(\bar{x})$

C2) $U^B(\bar{x}) \leq h(x^*) \implies h(\hat{x}) \leq h(x^*)$, for $\hat{x} \in E^B(\bar{x})$

An incumbent solution $\bar{x}$ that fulfills condition C1 cannot lead to any nondominated point of 2oKP different than $f(x^*)$ and, thus, it cannot improve the best known solution for HS(2oKP). Therefore, $\bar{x}$ can be pruned. If an incumbent solution $\bar{x}$ fulfills condition C2, none of the extensions in $E^B(\bar{x})$ can improve the objective function value as compared to $h(x^*)$ and, thus, it can be pruned as well.

Two further conditions follow from the interaction of the hypervolume scalarized problem and its biobjective related problem. For a given instance of HS(2oKP) with reference point (r_1, r_2) with respect to a point s and its predecessor $\overleftarrow{s}$ or successor $\overrightarrow{s}$ as computed in Algorithm 1, the following implication holds:

C3) $(r_1, r_2) \not\leq U^A(\bar{x}) \implies E^B(\bar{x}) = \emptyset$

If an incumbent solution $\bar{x}$ fulfills condition C3, none of the feasible extensions of $\bar{x}$ is feasible for the scalarization and it can be pruned.

Let a *local ideal point* be defined as $(s_1, \overleftarrow{s_2}) =: \mathring{p}$ or $(\overrightarrow{s_1}, s_2) =: \mathring{p}$, respectively. Note that point $\mathring{p}$ is not dominated by any feasible solution and it does not correspond to the pre-image of an efficient solution. Then, the following implication holds:

C4) $f(\hat{x}) \not\leq \mathring{p} \implies E^B(\bar{x}) = \emptyset$

Similar to C3, no feasible extension of an incumbent solution $\bar{x}$ that fulfills condition C4 can be feasible for HS(2oKP). Thus, it can be pruned in the branch-and-bound procedure.

The lower bound is computed by a greedy algorithm in a pre-processing phase that selects the variable x_i at each of the k steps that gives the largest increment in terms of the value of $h(x)$. The greedy solution is only considered as a lower bound if it is feasible with respect to the reference point constraint.

5.3. An example

The following example illustrates the application of the dichotomic search described in Algorithm 1 following the greedy scheme applied to this problem. Let $a = (11, 10, 9, 8, 7, 3, 2)$, $b = (3, 6, 5, 8, 1, 10, 7)$ and $k = 3$. The goal is to obtain a representation with four solutions. For the ILP linearization described in Section 5.1, we obtain the following matrix Q:

$$\begin{pmatrix} 33 & 66 & 55 & 88 & 11 & 110 & 77 \\ 30 & 60 & 50 & 80 & 10 & 100 & 70 \\ 27 & 54 & 45 & 72 & 9 & 90 & 63 \\ 24 & 48 & 40 & 64 & 8 & 80 & 56 \\ 21 & 42 & 35 & 56 & 7 & 70 & 49 \\ 9 & 18 & 15 & 24 & 3 & 30 & 21 \\ 6 & 12 & 10 & 16 & 2 & 20 & 14 \end{pmatrix}$$

iter.	(s_1, s_2)	(r_1, r_2)	(p_1, p_2)	$H(p, S)$	P	S	$H(S)$
0		$(0, 0)$	$(27, 19)$	513	$\emptyset$	$\rightarrow (27, 19)$	513
1	$(27, 19)$	$(27, 0)$	$(30, 14)$	42	$\rightarrow (30, 14)$		
		$(0, 19)$	$(21, 24)$	104	$\rightarrow (21, 24)$	$\rightarrow (21, 24)$	
					$(30, 14)$	$(27, 19)$	617
2	$(21, 24)$	$(21, 19)$	$(22, 21)$	2	$(30, 14)$		
					$\rightarrow (22, 21)$		
		$(0, 24)$	$(13, 25)$	13	$(30, 14)$	$(21, 24)$	
					$\rightarrow (13, 25)$	$(27, 19)$	
					$(22, 21)$	$\rightarrow (30, 14)$	659
3	$(30, 14)$	$(30, 0)$	$(30, 14)$	0	$(13, 25)$		
					$(22, 21)$		
		$(27, 14)$	$(29, 17)$	6	$(13, 25)$	$\rightarrow (13, 25)$	
					$\rightarrow (29, 17)$	$(21, 24)$	
					$(22, 21)$	$(27, 19)$	
						$(30, 14)$	672

Table 1: Iterations of Algorithm 1 for the example in Section 5.3

The value of the upper bounds is $\mathcal{U}^A = (30, 25)$ and $\mathcal{U}^B = 750$ for a reference point $(0, 0)$. The efficient set is

$$\mathcal{X}_E = \{(13, 25), (21, 24), (22, 21), (27, 19), (29, 17), (30, 14)\}$$

The set of representative nondominated points obtained by the dichotomic scheme is $\mathcal{X}_E \setminus \{(22, 21), (29, 17)\}$ with a hypervolume value of 672. Table 1 presents the points $s = (s_1, s_2)$, $p = (p_1, p_2)$, and the contents of sets S and P before (iteration 0) and at each iteration of the while loop (iterations 1 to 3) in Algorithm 1. The arrows indicate the point that was inserted into S and P. We also present the hypervolume contribution of each point p with respect to set S (column $H(p, S)$) as well as the hypervolume of set S (column $H(S)$). We exclude the initial border points of S. Figure 2 illustrates the location of the representation set (black circles), the nondominated set (black and gray circles) and reference points (white circles) that are found by the algorithm.

6. Numerical Tests

In this section, we present numerical results for the application of the hypervolume scalarization in combination with the dichotomic scheme described in Algorithm 1 using the greedy principle to find a representation set with a quality guarantee with respect to hypervolume for the biobjective knapsack problem with a cardinality constraint as described in the previous section. Of our particular interest is to understand the limits of our dichotomic scheme, both in

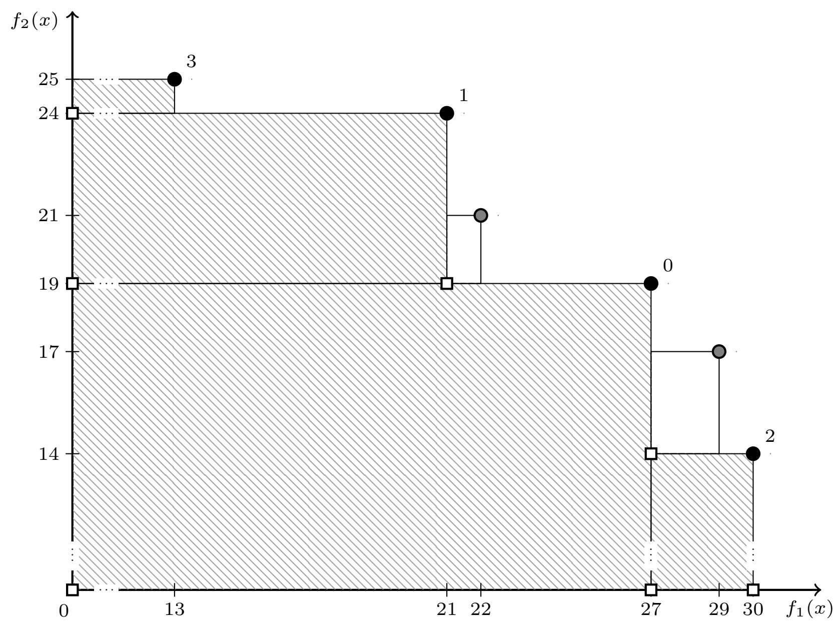


Figure 2: Illustration of the example in Section 5.3 with the representation set (black circles), the nondominated set (black and gray circles) and reference points (white rectangles); the value close to each nondominated point indicates the iteration number at which the point was inserted into S.

terms of computation time and representation quality, with respect to problem parameters and its dependency from the solution approach for the hypervolume scalarization. We analyzed the performance of the two approaches described in the previous section: i) a generic Integer Linear Programming (ILP) solver that solves the linearlized formulation described in Section 5.1 and ii) the combinatorial branch-and-bound described in Section 5.2 that takes into account both the combinatorial structure of the single-objective quadratic optimization problem and the dominance structure of the biobjective optimization problem. We considered two types of instances, uncorrelated and (positively) correlated instances, since we expect to have different performance on both types of instances, as also observed in a more general version of this problem (Bazgan et al., 2009; Figueira et al., 2013). In the first type, both a_i and b_i, $i = 1, \ldots, n$ follow a uniform discrete distribution within $[1, 100]$. In the correlated instances, we use the generation procedure described in (Bazgan et al., 2009) for general biobjective knapsack problems, that is, a_i follows a uniform discrete distribution within $[1, 100]$ and b_i follows a uniform discrete distribution within $[\max(90 - a_i, 1), \min(110 - a_i, 100)]$. A total of 30 instances were generated for each size $n = \{20, 30, 40, 50, 60\}$, constraint values $k = n/3$ and $n/2$, and representation sets of size $J = 5$, 15 and 25. For uncorrelated instances, no results are reported for $n = 20$ given that the size of the efficient set was very often too

n	k	$\lvert\mathcal{Y}_N\rvert$	J	time ILP	time BB	*ratio*	*wratio*
30	$n/3$	37	5	40.5	0.0	0.9982	0.8015
			15	148.9	0.1	0.9999	0.7860
			25	238.2	0.1	0.9999	0.7724
	$n/2$	39	5	69.5	0.0	0.9986	0.8727
			15	228.0	0.1	0.9999	0.8693
			25	350.4	0.1	0.9999	0.8378
40	$n/3$	55	5	202.9	0.1	0.9977	0.7937
			15	(87%) 542.0	0.2	0.9999	0.7760
			25	(70%) 686.8	0.2	0.9999	0.7778
	$n/2$	69	5	432.7	1.4	0.9986	0.8423
			15	-	1.5	0.9999	0.8330
			25	-	1.6	0.9999	0.8279
50	$n/3$	80	5	(73%) 763.3	5.2	0.9977	0.7771
			15	-	5.3	0.9999	0.7652
			25	-	5.6	0.9999	0.7515
	$n/2$	99	5	-	31.0	0.9987	0.8388
			15	-	32.4	0.9999	0.8300
			25	-	29.7	0.9999	0.8278
60	$n/3$	115	5	-	(87%) 118.3	0.9979	0.7582
			15	-	(87%) 122.7	0.9997	0.7538
			25	-	(87%) 121.2	0.9999	0.7551
	$n/2$	128	5	-	(70%) 321.7	0.9990	0.8273
			15	-	(67%) 348.3	0.9998	0.8173
			25	-	(67%) 354.6	0.9999	0.8150

Table 2: Size of the efficient set ($\lvert\mathcal{Y}_N\rvert$), average CPU time in seconds for the dichotomic scheme using an ILP solver (time ILP) and branch-and-bound (time BB), and greedy and worst representation ratio (*ratio* and *wratio*) for uncorrelated instances of size n, cardinality constraint k and representation size J.

small.

The dichotomic scheme, as described in Algorithm 1, was implemented in Python version 2.7. SCIP version 6.0.2, with default parameters, was used as ILP solver and the code for the branch-and-bound was implemented in `C` and compiled with `gcc` version 6.3.0, with flag `-O3`. The Python program generated the input data as text files to be read by the ILP solver and the branch-and-bound, which were called by using function `os.system()`. The running time is measured only with respect to the ILP solver and the branch-and-bound code using function `time.time()` in Python. We defined a cut-off limit of 1000 seconds for the total time of the dichotomic scheme.

The representation quality obtained by the dichotomic scheme is expressed as an approximation ratio $H(S_G)/H(S_O)$, where S_G is the representation set returned by the dichotomic scheme and S_O is the optimal representation set. We

n	k	$\lvert\mathcal{Y}_N\rvert$	J	time ILP	time BB	*ratio*	*wratio*
20	$n/3$	113	5	17.2	0.0	0.9851	0.3590
			15	72.4	0.1	0.9991	0.3139
			25	121.2	0.1	0.9997	0.2950
	$n/2$	145	5	18.5	0.1	0.9926	0.4980
			15	123.0	0.1	0.9988	0.4679
			25	210.3	0.2	0.9998	0.4467
30	$n/3$	249	5	265.7	15.1	0.9847	0.3688
			15	-	18.8	0.9984	0.3348
			25	-	19.3	0.9994	0.3235
	$n/2$	284	5	362.2	88.5	0.9910	0.4908
			15	-	104.9	0.9984	0.4632
			25	-	107.7	0.9995	0.4554

Table 3: Size of the efficient set ($|\mathcal{Y}_N|$), average CPU time in seconds for the dichotomic scheme using an ILP solver (time ILP) and the branch-and-bound (time BB) greedy and worst representation ratio (*ratio* and *wratio*) for correlated instances of size n, cardinality constraint k and representation size J.

applied an ϵ-constraint method (Haimes et al., 1971) to each instance in order to obtain the nondominated set, from which the optimal representation set S_O was extracted using the approach described in (Kuhn et al., 2016)[1]. The correctness of the output obtained by our dichotomic search was validated against the output generated by the algorithm described in Guerreiro et al. (2016)[2], which applies the same greedy principle to extract a representative set from a set of nondominated points. For reference, we have also reported the worst representation ratio, by computing the worst representation set in terms of hypervolume. This set is computed by using a modified version of the dynamic programming algorithm described in Auger et al. (2009).

The order in which each variable i is fixed in the branch-and-bound code is determined in a pre-processing step by its ranking with respect to the value of $\max(a_i, b_i)$. The value of the hypervolume scalarization is computed incrementally in linear time by using matrix $\hat{Q}$ (see Section 5.1), except in the case of reference point $(0, 0)$, which can be computed incrementally in a constant amount of time. The pruning conditions and the two upper bounds described in Section 5.2 are computed at each recursive step. The first computation of both upper bounds for each variable index and number of variables set to one is performed in time $O(k^2 + n)$. These values are stored in a table during the run in order to be re-used in succeeding steps in constant amount of time.

All experiments were conducted on a machine with Linux Debian 9.12 Operating System (64-bit), an Intel i5-7200U Dual-Core processor running at 2.5 GHz (base

[1] Code available in `https://eden.dei.uc.pt/~paquete/HSSP`

[2] Code available in `https://github.com/apguerreiro/gHSS`

frequency) and 8GB RAM.
Tables 2 and 3 present the CPU-time taken by both approaches, as well as the size of the nondominated set and the two approximation ratios, for uncorrelated and correlated instances, respectively. The values are averaged over all instances with the same combination of parameters. The values reported in brackets in the running times correspond to the percentage of instances that a program was able to solve, given that it was not able to solve all instances. For that cases, we report the censored mean of running time.
We can observe that the ILP solver spends much more time than branch-and-bound for both uncorrelated and correlated instances. The ILP solver cannot solve uncorrelated instances of size 50 and correlated instances of size 30 within the time limit, except for small k and for small cardinality of the representation set. Its performance is also very dependent of the cardinality of the representation set, as opposed to the branch-and-bound. A closer look into the time spent on solving each scalarization for uncorrelated instances indicates that the ILP solver presents a very large variance. In general, its time increases as the constraint based on the reference point becomes tighter.
Noteworthly, the approximation ratio is very close to 1.0, which indicates that the dichotomic search is able to return high-quality representations of the nondominated set. The worst approximation ratio is also high for uncorrelated instances but small for correlated instances. Note that the size of the nondominated set is much larger for the latter type of instances.

7. Concluding Remarks

In this article, we introduced a dichotomic scheme for biobjective discrete optimization problems based on hypervolume scalarizations. Differently from the dichotomic scheme based on the weighted sum scalarization, our dichotomic scheme does not rely on convexity assumptions. However, an hypervolume scalarization may be harder to solve than a weighted sum scalarization due to its quadratic formulation, in particular, for generic ILP solvers. Moreover, we conjecture that solving an hypervolume scalarization is, in general, an NP-hard problem. Future work consists of developing implicit enumeration approaches that can deal with hypervolume scalarizations. The possibility explored in this article consists of exploring both the hypervolume scalarized formulation and the dominance structure of the underlying multiobjective optimization problem. Our numerical results suggest that these type of approaches may surpass generic ILP solvers.
Further investigation consists of extending this framework for more than two objectives. This is less trivial since not only two neighboring subproblems exist with respect to a new point in the representation. These issues have been treated for the case weighted sum scalarization (Przybylski et al., 2010), thus we expect that a similar approach could be devised for the hypervolume scalarization.
Our dichotomic scheme can easily be modified to find the complete set of efficient solutions, by allowing to run until queue P is empty; see Algorithm 1. Moreover, by definition, it has a good anytime behaviour, that is, a good trade-off

between run time and representation quality, as measured in terms of hypervolume. The anytime models investigated in Jesus et al. (2020) indicate that anytime algorithms based on hypervolume scalarizations have a logarithmic rate of convergence in terms of relative hypervolume, if the runtimes for solving each scalarization are similar. These theoretical models can be used within our dichotomic scheme to detect deviations that may justify a switch to a different search strategy.

Acknowledgments. This work was supported by the bilateral cooperation project *Multiobjective Network Interdiction* funded by the Deutscher Akademischer Austauschdienst and Fundação para a Ciência e Tecnologia.

References

Aneja, Y., Nair, K., 1979. Bicriteria transportation problem. Management Science 25, 73–78.

Auger, A., Bader, J., Brockhoff, D., Zitzler, E., 2009. Investigating and exploiting the bias of the weighted hypervolume to articulate user preferences, in: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA. pp. 563–570.

Auger, A., Bader, J., Brockhoff, D., Zitzler, E., 2012. Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. Theoretical Computer Science 425, 75 – 103.

Bader, J., Zitzler, E., 2011. HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. Evolutionary Computation 19, 45–76.

Basseur, M., Derbel, B., Goëffon, A., Liefooghe, A., 2016. Experiments on greedy and local search heuristics for d-dimensional hypervolume subset selection, in: Proceedings of the 2016 Genetic and Evolutionary Computation Conference (GECCO 2016), ACM Press, Denver, Colorado, USA. pp. 541–548.

Bazgan, C., Hugot, H., Vanderpooten, D., 2009. Solving efficiently the 0-1 multi-objective knapsack problem. Computers & Operations Research 36, 260–279.

Beume, N., Naujoks, B., Emmerich, M.T.M., 2007. SMS-EMOA: Multiobjective Selection Based on Dominated Hypervolume. European Journal of Operational Research 181, 1653–1669.

Boland, N., Charkhgard, H., Savelsbergh, M., 2015. A criterion space search algorithm for biobjective integer programming: The balanced box method. INFORMS Journal on Computing 247, 735–754.

Bradstreet, L., Barone, L., While, L., 2007. Incrementally maximizing hypervolume for selection in multi-objective evolutionary algorithms, in: Proceedings of the 2007 IEEE Congress on Evolutionary Computation and Evolutionary (CEC 2007), pp. 3203–3210.

Bringmann, K., Cabello, S., Emmerich, M.T.M., 2017. Maximum volume subset selection for anchored boxes, in: 33rd International Symposium on Computational Geometry (SoCG 2017), Dagstuhl Publishing, Schloss Dagstuhl Leibniz-Zentrum für Informatik, Germany. pp. 22:1–22:15.

Bringmann, K., Friedrich, T., 2010. An efficient algorithm for computing hypervolume contributions. Evolutionary Computation 18, 383–402.

Bringmann, K., Friedrich, T., Klitzke, P., 2014. Two-dimensional Subset Selection for Hypervolume and Epsilon-Indicator, in: Proceedings of the 2014 Genetic and Evolutionary Computation Conference (GECCO 2014), ACM Press, Vancouver, Canada. pp. 589–596.

Caprara, A., Pisinger, D., Toth, P., 1999. Exact solution of the quadratic knapsack problem. INFORMS Journal on Computing 11, 125–137.

Dächert, K., Klamroth, K., Lacour, R., Vanderpooten, D., 2017. Efficient computation of the search region in multi-objective optimization. European Journal of Operational Research 260, 841–855.

Ehrgott, M., 2005. Muticriteria Optimization. 2nd ed., Springer.

Eusébio, A., Figueira, J.R., Ehrgott, M., 2014. On finding representative non-dominated points for bi-objective integer network flow problems. Computers & Operations Research 48, 1–10.

Faulkenberg, S., Wiecek, M., 2010. On the quality of discrete representations in multiple objective programming. Optimization and Engineering 11, 423–440.

Figueira, J.R., Paquete, L., Simões, M., Vanderpooten, D., 2013. Algorithmic improvements on dynamic programming for the bi-objective $\{0, 1\}$ knapsack problem. Computational Optimization and Applications 56, 97–111.

Friedrich, T., Neumann, F., 2015. Maximizing submodular functions under matroid constraints by evolutionary algorithms. Evolutionary Computation 23, 543–558.

Gomes, R., Guerreiro, A., Kuhn, T., Paquete, L., 2018. Implicit enumeration strategies for the hypervolume subset selection problem. Computers & Operations Research 100, 244–253.

Groz, B., Maniu, S., 2019. Hypervolume subset selection with small subsets. Evolutionary Computation 27, 611–637.

Guerreiro, A.P., Fonseca, C.M., Paquete, L., 2016. Greedy hypervolume subset selection in low dimensions. Evolutionary Computation 23, 521–544.

Guerreiro, A.P., Fonseca, C.M., Paquete, L., 2020. The Hypervolume Indicator: Problems and Algorithms. arXiv e-prints , arXiv:2005.00515arXiv:2005.00515.

Haimes, Y., Lasdon, L., Wismer, D., 1971. On a bicriterion formulation of the problems of integrated system identification and system optimization. IEEE Transactions on Systems, Man, and Cybernetics 1, 296–297.

Hamacher, H., Pedersen, C., Ruzika, S., 2007. Finding representative systems for discrete bicriterion optimization problems. Operations Research Letters 35, 336–344.

Hernandez, V.A.S., Schütze, O., Wang, H., Deutz, A., Emmerich, M., 2018. The set-based hypervolume newton method for bi-objective optimization. IEEE Transactions on Cybernetics , 1–11.

Jesus, A., Paquete, L., Liefooghe, A., 2020. A model of anytime algorithm performance for bi-objective optimization. Journal of Global Optimization .

Jesus, A.D., Paquete, L., Figueira, J.R., 2018. Finding representations for an unconstrained bi-objective combinatorial optimization problem. Optimization Letters 12, 321–334.

Kidd, M.P., Lusby, R., Larsen, J., 2020. Equidistant representations: Connecting coverage and uniformity in discrete biobjective optimization. Computers & Operations Research 117, 104872.

Kirlik, G., Sayın, S., 2018. Bilevel programming for generating discrete representations in multiobjective optimization. Mathematical Programming 169, 585–604.

Kuhn, T., Fonseca, C.M., Paquete, L., Ruzika, S., Duarte, M.M., Figueira, J.R., 2016. Hypervolume Subset Selection in Two Dimensions: Formulations and Algorithms. Evolutionary Computation 24, 411–425.

Miettinen, K., 1998. Nonlinear Multiobjective Optimization. Springer.

Nemhauser, G., Wolsey, L., Fisher, M., 1978. An analysis of approximations for maximizing submodular set functions - I. Mathematical Programming 14, 265–294.

Przybylski, A., Gandibleux, X., Ehrgott, M., 2010. A recursive algorithm for finding all nondominated extreme points in the outcome set of a multiobjective integer programme. INFORMS Journal on Computing 22, 371–386.

Sayın, S., 2000. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. Mathematical Programming 87, 543–560.

Sayın, S., 2003. A procedure to find discrete representations of the efficient set with specified coverage errors. Operations Research 51, 427–436.

Schultes, J., Stiglmayr, M., Klamroth, K., Hahn, C., 2020. Hypervolume Scalarization for Shape Optimization to Improve Reliability and Cost of Ceramic Components. Technical Report. University of Wuppertal. URL: https://www.imacm.uni-wuppertal.de/fileadmin/imacm/preprints/2020/imacm_20_04.pdf.

Schulze, B., Stiglmayr, M., Paquete, L., Fonseca, C.M., Willems, D., Ruzika, S., 2020. On the rectangular knapsack problem: approximation of a specific quadratic knapsack problem. Mathematical Methods of Operations Research doi:10.1007/s00186-020-00702-0. (in press).

Serafini, P., 1987. Some considerations about computational complexity for multi objective combinatorial problems, in: Jahn, J., Krabs, W. (Eds.), Recent Advances and Historical Development of Vector Optimization, Springer. pp. 222–232.

Sylva, J., Crema, A., 2007. A method for finding well-dispersed subsets of non-dominated vectors for multiple objective mixed integer linear programs. European Journal of Operational Research 180, 1011–1027.

Torrico, A., Singh, M., Pokutta, S., 2020. On the unreasonable effectiveness of the greedy algorithm: Greedy adapts to sharpness. arXiv:2002.04063v1.

Touré, C., Hansen, N., Auger, A., Brockhoff, D., 2019. Uncrowded hypervolume improvement, in: Proceedings of the Genetic and Evolutionary Computation Conference on - GECCO'19, ACM Press. pp. 638–646. doi:10.1145/3321707.3321852.

Ulrich, T., Thiele, L., 2012. Bounding the effectiveness of hypervolume-based $(\mu + \lambda)$-archiving algorithms, in: Learning and Intelligent Optimization - 6th International Conference, LION 6, Paris, France, January 16-20, 2012, Revised Selected Papers, pp. 235–249.

Vaz, D., Paquete, L., Fonseca, C.M., Klamroth, K., Stiglmayr, M., 2015. Representation of the non-dominated set in biobjective discrete optimization. Computers & Operations Research 63, 172 – 186.

Watters, L.J., 1967. Reduction of integer polynomial programming problems to zero-one linear programming problems. Operations Research 15, 1171–1174.

Wierzbicki, A.P., 1986a. A methodological approach to comparing parametric characterizations of efficient solutions, in: Lecture Notes in Economics and Mathematical Systems. Springer Berlin Heidelberg, pp. 27–45. doi:10.1007/978-3-662-02473-7_4.

Wierzbicki, A.P., 1986b. On the completeness and constructiveness of parametric characterizations to vector optimization problems. OR-Spektrum 8, 73–87. doi:10.1007/bf01719738.

Yang, K., Emmerich, M., Deutz, A., Bäck, T., 2019. Efficient computation of expected hypervolume improvement using box decomposition algorithms. Journal of Global Optimization 75, 3–34. doi:10.1007/s10898-019-00798-7.

Zitzler, E., Thiele, L., 1998. Multiobjective optimization using evolutionary algorithms - A comparative case study, in: Parallel Problem Solving from Nature - PPSN V, 5th International Conference, Amsterdam, The Netherlands, September 27-30, 1998, Proceedings, pp. 292–304.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G., 2003. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE Transactions on Evolutionary Computation 7, 117–132.