Hypervolume Subset Selection in Two Dimensions: Formulations and Algorithms

Tobias Kuhn
Mathematical Institute, University of Kaiserslautern, Germany

Carlos M. Fonseca
CISUC, Department of Informatics Engineering, University of Coimbra, Portugal

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

Stefan Ruzika
Mathematical Institute, University of Koblenz-Landau, Campus Koblenz, Germany

Miguel M. Duarte
CISUC, Department of Informatics Engineering, University of Coimbra, Portugal

José Rui Figueira
CEG-IST, Instituto Superior Técnico, Universidade de Lisboa, Portugal

doi:10.1162/EVCO_a_00157

Abstract

The hypervolume subset selection problem consists of finding a subset, with a given cardinality k, of a set of nondominated points that maximizes the hypervolume indicator. This problem arises in selection procedures of evolutionary algorithms for multiobjective optimization, for which practically efficient algorithms are required. In this article, two new formulations are provided for the two-dimensional variant of this problem. The first is a (linear) integer programming formulation that can be solved by solving its linear programming relaxation. The second formulation is a k-link shortest path formulation on a special digraph with the Monge property that can be solved by dynamic programming in $\mathcal{O}(k(n-k)+n\log n)$ time. This improves upon the result of $\mathcal{O}(n^2k)$ in Bader (2009), and slightly improves upon the result of $\mathcal{O}(nk+n\log n)$ in Bringmann et al. (2014b), which was developed independently from this work using different techniques. Numerical results are shown for several values of n and k.

Keywords

Multiobjective optimization, subset selection, hypervolume, k-link shortest path.

1 Introduction

Given a set of nondominated points in objective space, the hypervolume indicator measures the dominated region of this space bounded by some reference point (Zitzler and Thiele, 1998). Because of its properties (Knowles and Corne, 2002; Zitzler et al., 2003; 2008), this indicator has been instrumental both in the assessment of the performance of multiobjective evolutionary algorithms (Knowles et al., 2000; Zitzler et al., 2000) and in the development of multiobjective selection and archiving procedures for such algorithms (Fleischer, 2003; Knowles et al., 2003; Knowles and Corne, 2003; Zitzler and Künzli, 2004; Beume et al., 2007; Bader and Zitzler, 2011).

Manuscript received: May 14, 2014; revised: February 21, 2015; accepted: May 5, 2015.

The hypervolume subset selection problem (HSSP) consists of finding a subset of k elements from a set of n nondominated points that maximizes the hypervolume indicator. This has been considered computationally expensive for an arbitrary number of objectives (Bader and Zitzler, 2011). Therefore, the main focus has been on finding the $n-k$ elements that contribute the least in terms of hypervolume in a greedy manner, e.g., in Huband et al. (2003); Beume et al. (2007); Igel et al. (2007); see also approximation results by Bringmann and Friedrich (2009) and an asymptotically optimal algorithm for finding all contributions of every element in the given set in $\Theta(n \log n)$ for two and three dimensions (Emmerich and Fonseca, 2011).

So far, the tightest time complexity bound for solving the HSSP to optimality for $d > 2$ dimensions and arbitrary k is $\mathcal{O}(n^{d/2} \log n + n^k)$ (Bringmann and Friedrich, 2010). For the particular case of $d = 2$, to the best of our knowledge, only two approaches have been proposed. Bader (2009) introduced a dynamic programming algorithm with $\mathcal{O}(n^2 k)$ time complexity. This algorithm is based on the fact that the contribution to the hypervolume indicator of the leftmost point of a given nondominated subset depends only on its immediate neighbor. Independently from the present work, Bringmann et al. (2014b) proposed another approach to the same problem with a time complexity of $\mathcal{O}(n(k + \log n))$. This algorithm computes, in the ℓth iteration, all maximal hypervolume indicator values using at most ℓ points with respect to n appropriately chosen reference points. This can be done for each reference point by computing the maximum of $\mathcal{O}(n)$ different linear function evaluations. The running time is achieved by using a linear time algorithm to compute the upper envelope of lines.

In this article, we propose two different formulations for the two-dimensional case of the HSSP: an integer programming formulation and a k-link shortest path formulation. Both formulations are based on a preprocessing step, which makes a partition of the dominated region into different areas induced by the set of nondominated points. We show that the polyhedron of the linear programming relaxation of the first formulation is integral, which allows linear programming methods, such as simplex and interior-point methods, to solve the HSSP. In the k-link shortest path formulation, the arc costs have a special property, called the Monge property. This property allows us to solve the HSSP with a simple dynamic programming approach in $\mathcal{O}(k(n-k) + n \log n)$ time, which slightly improves upon the result of Bringmann et al. (2014b). This time complexity suggests that our approach should perform better for large values of k.

The remainder of this article is organized as follows. In Section 2 we introduce concepts, definitions, and notation used throughout this article. In Section 3 we explain the crucial preprocessing step for calculating the weights used in the integer programming formulation and for the introduction of the arc costs in the k-link shortest path formulation. In Section 4 we introduce an integer programming formulation for the HSSP and prove the integrality of the polyhedron of its linear programming relaxation. In Section 5 we present the k-link shortest path formulation, which is used to achieve an improved complexity bound for the HSSP. An experimental analysis is provided in Section 6. Finally, in Section 7 we provide some conclusions and avenues for future research.

2 Terminology

In the following, some concepts, definitions, and the notation used in this article are given. Let $z^1, z^2 \in \mathbb{R}^q$. We define the following ordering relations on $\mathbb{R}^q$:

$$z^1 \geqq z^2 :\Leftrightarrow z_i^1 \geqslant z_i^2 \text{ for } i = 1, 2, \ldots, q,$$
$$z^1 \geq z^2 :\Leftrightarrow z^1 \geqq z^2 \text{ and } z^1 \neq z^2,$$
$$z^1 > z^2 :\Leftrightarrow z_i^1 > z_i^2 \text{ for } i = 1, 2, \ldots, q.$$

DEFINITION 1 (SET OF NONDOMINATED POINTS): *A point $z'' \in \mathbb{R}^q$ dominates $z' \in \mathbb{R}^q$ if $z'' \geq z'$. Let $N = \{z^1, \ldots, z^n\} \subseteq \mathbb{R}^q$ denote a set of nondominated points, in which no point in N is dominating another point in N.*

DEFINITION 2 (HYPERVOLUME INDICATOR): *Let $N = \{z^1, \ldots, z^n\}$ and let z^{ref} be a reference point satisfying $z^{ref} < z^i$ for all $i = 1, \ldots, n$. The set*

$$D(N) := \bigcup_{i=1}^{n} \left\{ z \in \mathbb{R}^q : z^{ref} \leqq z \leqq z^i \right\}$$

is called the dominated region of N (w.r.t. z^{ref}), and the hypervolume indicator of N (w.r.t. z^{ref}) is defined as $S(N) := \lambda\left(D(N)\right)$, where $\lambda(\cdot)$ denotes the Lebesgue measure in $\mathbb{R}^q$.

The *hypervolume indicator* maps a set of nondominated points to the size of the region in the corresponding space dominated by this set and bounded below by a reference point.

DEFINITION 3 (HSSP): *Let $N = \{z^1, \ldots, z^n\}$ and let $k \in \{1, \ldots, n\}$. The hypervolume subset selection problem (HSSP) consists of selecting a subset $N' \subseteq N$ with $|N'| = k$ such that the value of the hypervolume indicator $S(N')$ on the subset is maximal, i.e.,*

$$S(N') = \max_{\substack{N'' \subseteq N \\ |N''| = k}} S(N'').$$

In the following sections, we assume a set of nondominated points $N := \{z^1, \ldots, z^n\} \subseteq \mathbb{R}^2$, some reference point $z^{ref} \in \mathbb{R}^2$, and a desired cardinality $k \in \{1, \ldots, n\}$ to be given. We further assume that the points in set N are sorted in increasing order of the first component, i.e., $z_1^i < z_1^j$ for $i < j$, which can be achieved in $\mathcal{O}(n \log n)$ time.

3 Preprocessing: Decomposition of the Dominated Region

This section describes the preprocessing step, which is crucial for the definition of the variables and the weights in the integer programming formulation and the definition of the arc costs in the k-link shortest path formulation. The dominated region $D(N)$ can be partitioned into certain rectangles. Let A_{ij}, $i \geqslant j$, be the rectangle defined by the subregion of $D(N)$, which is exclusively dominated by all points in $\{z^j, \ldots, z^i\}$ and no other point in N. An example of this partition is given in Figure 1. For every such rectangle, we define w_{ij} as the area $\lambda\left(A_{ij}\right)$ of rectangle A_{ij}. If we define $z_1^0 := z_1^{ref}$ and $z_2^{n+1} := z_2^{ref}$, the rectangle A_{ij} can be written as

$$A_{ij} = \left\{ z \in \mathbb{R}^2 : \begin{pmatrix} z_1^{j-1} \\ z_2^{i+1} \end{pmatrix} \leqq z \leqq \begin{pmatrix} z_1^{j} \\ z_2^{i} \end{pmatrix} \right\}.$$

Hence, we get $w_{ij} = (z_1^j - z_1^{j-1}) \cdot (z_2^i - z_2^{i+1})$ and we can calculate all the weights w_{ij}, $i \geqslant j$, in $\mathcal{O}(n^2)$ time.

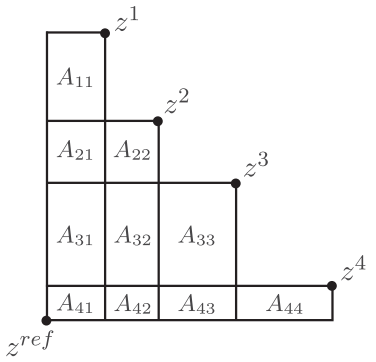


Figure 1: Partition of the dominated region for a given set $N = \{z^1, z^2, z^3, z^4\}$.

4 An Integer Programming Formulation

This section presents an integer programming (IP) formulation for the HSSP and shows that we can efficiently solve this formulation by solving its linear programming relaxation. It is assumed that the reader is familiar with the foundations of linear programming; otherwise, we refer the reader to, e.g., the book by Bazaraa et al. (2011). Following the notation in Section 3, we denote with w_{ij} the area of A_{ij}, $i \geqslant j$, where A_{ij} is the rectangle defined by the subregion of $D(N)$, which is exclusively dominated by $\{z^j, \ldots, z^i\}$ and no other point in N. The following IP formulation models the corresponding HSSP:

$$(IP_k) \quad \max \quad \sum_{i=1}^{n} \sum_{j=1}^{i} w_{ij} x_{ij} \tag{1}$$

$$\text{subject to} \quad \sum_{\ell=1}^{n} x_{\ell\ell} = k \tag{2}$$

$$\sum_{\ell=j}^{i} x_{\ell\ell} \geqslant x_{ij} \qquad i = 2, \ldots, n;\ j = 1, \ldots, i-1 \tag{3}$$

$$x_{ij} \in \{0, 1\} \qquad i = 1, \ldots, n;\ j = 1, \ldots, i.$$

Thereby, variable $x_{\ell\ell}$ is equal to 1 if and only if z^ℓ is selected, and variable x_{ij} determines whether the subregion A_{ij} is covered by some point in $\{z^j, \ldots, z^i\}$, which is guaranteed by the constraints (3). Constraint (2) ensures the compliance of the selection of exactly k points, and the objective function (1) calculates the value of the current hypervolume indicator, which has to be maximized.

For proving that this IP can be efficiently solved with the help of the corresponding linear programming (LP) relaxation, we need the notion of a totally unimodular matrix, which will lead to a famous result from integer optimization.

DEFINITION 4 (TOTALLY UNIMODULAR MATRIX): *A matrix $A \in \mathbb{R}^{p \times q}$ is called totally unimodular if the determinant of each square submatrix of A belongs to $\{0, 1, -1\}$.*

THEOREM 1 (INTEGRALITY) (Nemhauser and Wolsey, 1999): *Let $b, b' \in \mathbb{Z}^p$, $d, d' \in \mathbb{Z}^q$. If $A \in \mathbb{R}^{p \times q}$ is totally unimodular and $P := \{x \in \mathbb{R}^q : b' \leqq Ax \leqq b, d' \leqq x \leqq d\} \neq \emptyset$, then P is an integral polyhedron, i.e., each of its nonempty faces contains an integral point.*

Consider now the LP relaxation. We show that the constraint matrix of this LP in some standard form is totally unimodular. The LP relaxation is given by the following formulation:

$$(LP_k) \qquad \max \quad \sum_{i=1}^{n}\sum_{j=1}^{i} w_{ij}x_{ij} \tag{4}$$

$$\text{subject to} \quad \sum_{\ell=1}^{n} x_{\ell\ell} = k \tag{5}$$

$$\sum_{\ell=j}^{i} x_{\ell\ell} - x_{ij} - s_{ij} = 0 \qquad i = 2,\ldots,n;\ j = 1,\ldots,i-1 \tag{6}$$

$$0 \leqslant x_{ij} \leqslant 1 \qquad i = 1,\ldots,n;\ j = 1,\ldots,i$$

$$s_{ij} \geqslant 0 \qquad i = 2,\ldots,n;\ j = 1,\ldots,i-1,$$

where the new variables s_{ij} are *surplus variables* (Bazaraa et al., 2011).

If we rearrange the columns in a certain way, first the variables $x_{\ell\ell}$, $\ell = 1,\ldots,n$, and then the variables x_{ij} and s_{ij}, $i = 2,\ldots,n$, $j = 1,\ldots,i-1$, according to the ordering of the constraints (6), the structure of the constraint matrix corresponding to the constraints (5) and (6) of (LP_k) is given by

$$\left(\begin{array}{c|c|c} 1\ldots1 & 0\ldots0 & 0\ldots0 \\ \hline C & -I & -I \end{array}\right),$$

where C is a $\frac{n(n-1)}{2} \times n$-matrix and $-I$ is the negative of the $\frac{n(n-1)}{2} \times \frac{n(n-1)}{2}$ identity matrix. Let us denote by $\tilde{C}$ the submatrix $\binom{e}{C}$, where $e \in \mathbb{R}^n$ is the vector of all 1s, and by D the submatrix $\left(\begin{array}{c|c} 0\ldots0 & 0\ldots0 \\ \hline -I & -I \end{array}\right)$. Observe that D is totally unimodular and $\tilde{C}$ has the *consecutive 1s property* (Nemhauser and Wolsey, 1999) and thus is also totally unimodular.

THEOREM 2: *The constraint matrix of (LP_k) is totally unimodular.*

PROOF: Let B denote an arbitrary squared submatrix of the constraint matrix of (LP_k).

Case 1: B is completely contained in $\tilde{C}$ or completely contained in D and therefore $\det(B) \in \{0, \pm 1\}$, since both matrices are totally unimodular.

Case 2: B possesses $s > 0$ and $t > 0$ columns from matrix $\tilde{C}$ and matrix D, respectively, w.l.o.g. no duplicate column from D.

Choose some column $j > s$ from B belonging to D, and expand the determinant of B with respect to the jth column (*Laplace expansion*). Since this column has only one nonzero entry, say b_{ij}, we get $\det(B) = (-1)^{i+j+1} \cdot \det(M_{ij})$, where M_{ij} is the minor of matrix B formed by eliminating row i and column j from B. The minor M_{ij} also corresponds to a squared submatrix of the constraint matrix. If we follow the above Laplace expansion, after t steps, we get a submatrix $\tilde{B}$ of B matching *Case 1*, i.e., $\det(\tilde{B}) \in \{0, \pm 1\}$. Then, by construction, we get $\det(B) = \pm\det(\tilde{B}) \in \{0, \pm 1\}$.

Since B was an arbitrarily chosen squared submatrix, we have shown the totally unimodular property of the constraint matrix. □

COROLLARY 3 (INTEGRALITY): *The polyhedron corresponding to (LP_k) is integral. In particular, (IP_k) can be solved by solving its linear relaxation (LP_k) with an appropriate LP solver.*

PROOF: This follows from Theorems 1 and 2 and the following upper bound on the surplus variables:

$$s_{ij} = \sum_{\ell=j}^{i} x_{\ell\ell} - x_{ij} \leqslant k - x_{ij} \leqslant k \quad i = 2, \ldots, n;\ j = 1, \ldots, i-1. \qquad \square$$

5 A k-link Shortest Path Formulation with the Monge Property

In the following, we show that the HSSP can be modeled using a k-link shortest path formulation in a directed graph (digraph). The corresponding shortest path problem with a cardinality constraint can then be solved using a dynamic programming (DP) approach. This digraph has a special structure, the Monge property, that allows us to solve the HSSP in $\mathcal{O}(k(n-k) + n \log n))$ time. Without loss of generality, we assume $k < n$, since for the case $k = n$, only the preprocessing step is needed to obtain the optimal objective value (see also Algorithm 2).

We first explain the construction of the digraph $G = (V, E)$ related to given set N. The graph construction is based on the observation that for each choice of a subset $\{z^{s_1}, \ldots, z^{s_k}\}$, $s_i \leqslant s_j$ for $i < j$, the contribution to the hypervolume indicator of the consecutive points $\{z^{s_i+1}, \ldots, z^{s_{i+1}-1}\}$ for two indices with $s_i + 1 < s_{i+1}$ only depends on the coordinates of the points z^{s_i} and $z^{s_{i+1}}$. For each element $z^c \in N$, we create a node $c \in V$. In addition, we also add two other nodes 0 and $n+1$ to V, as source and target nodes, respectively. We add the arcs $e_{uv} := (u, v)$, for all $u, v \in \{0, \ldots, n+1\}$ with $u < v$ to E. According to the notation in the preprocessing step (see Section 3), the cost c_{uv} of an arc e_{uv} is defined as follows:

$$c_{uv} := \sum_{i=u+1}^{v-1} \sum_{j=u+1}^{i} w_{ij},$$

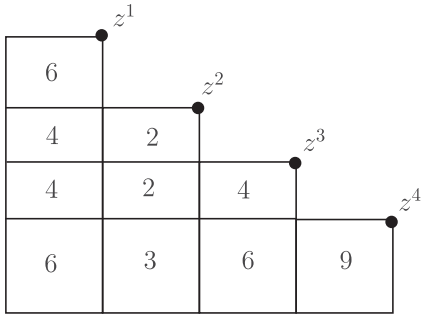
where $c_{u,u+1} = 0$ for all $u \in \{0, \ldots, n\}$.

By definition, we get that the cost c_{uv} describes the contribution to the hypervolume indicator of the whole set $\{z^{u+1}, \ldots, z^{v-1}\}$, which is the *exclusive volume* of the set $\{z^{u+1}, \ldots, z^{v-1}\}$ and denoted by $EV(z^{u+1}, z^{v-1})$. An example for the graph construction is depicted in Figure 2.

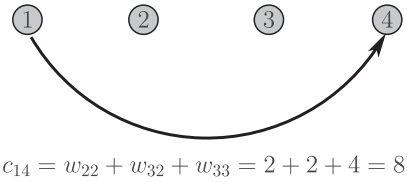
From the construction of the graph, we immediately get the following property.

REMARK 1: Each choice in the HSSP of a subset $\{z^{s_1}, \ldots, z^{s_{k-1}}\}$ of N with cardinality $k-1$ corresponds to a path in our constructed digraph with exactly k arcs that starts in node 0, visits the nodes s_1 to s_{k-1}, and ends in the node $n+1$. Since the cost of a used arc e_{uv} corresponds to the exclusive volume of the jumped-over nodes $u+1$ to $v-1$, the hypervolume contribution $S(N)$ minus the total cost of the path corresponds then to the hypervolume contribution of the corresponding subset of N. Hence, the k-link shortest path problem on our constructed digraph models the HSSP with desired cardinality $k-1$.

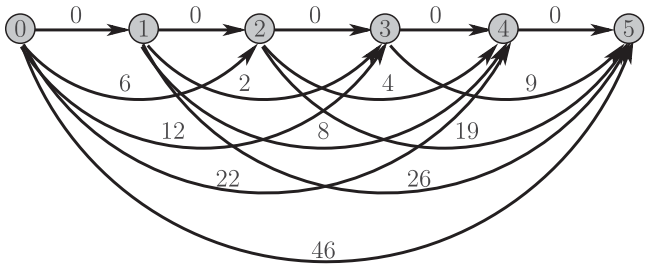
Since in our special k-link shortest path problem the *Bellman principle of optimality* is valid, we can use a straightforward DP approach to solve this problem (see Algorithm 1). In each iteration, the length $D(\ell, v)$ of the optimal path for the problem of finding the ℓ-link shortest path from 0 to v is calculated. However, this would not lead directly



(a) Partition of $D(N)$ with assigned weights $w_{ij}, i \geqslant j$



(b) Calculation of the arc cost c_{14}



(c) Constructed graph with all calculated costs

Figure 2: Example for the graph construction.

Algorithm 1 DP for the special k-link shortest path problem

Input: $G = (V, E)$ from above, $k \in \{1, \ldots, n\}$
Output: $D(k, n+1)$ length of the optimal path from 0 to $n+1$ with k arcs

1: $D(1, v) := c_{0v}$ for all $v \in \{1, \ldots, n-k+2\}$
2: **for** $\ell = 2, \ldots, k$ **do**
3: **for** $v = \ell, \ldots, n+1-k+\ell$ **do**
4: $D(\ell, v) := \min_{u=\ell-1,\ldots,v-1} \{D(\ell-1, u) + c_{uv}\}$

to a much better running time than Bader's DP algorithm, since finding the minimum in line 4 is in a naïve way done in $\mathcal{O}(n-k)$, resulting in an overall running time in $\mathcal{O}((n-k)^2 k)$.

In the following, we show that the time complexity can be improved by proving some special structure for this digraph, the so-called *(concave) Monge property* (Aggarwal et al., 1994).

THEOREM 4 (MONGE PROPERTY): *Consider the following arcs*

$$e_{ij},\ e_{i,j-1},\ e_{i+1,j},\ e_{i+1,j-1}$$

for some i, j with $j > i+2$ (see Figure 3). Then we have

$$c_{ij} > c_{i,j-1} + c_{i+1,j} - c_{i+1,j-1}.$$

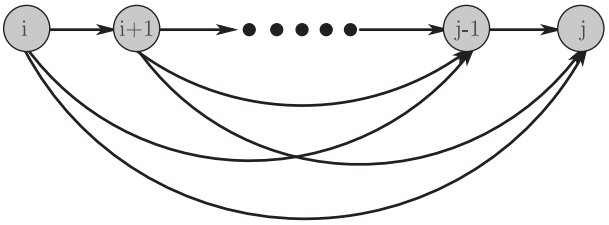


Figure 3: Selected arcs in Theorem 4.

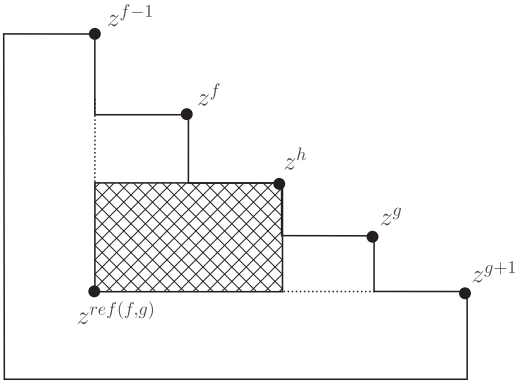


Figure 4: Example for $B^{(f,g)}(z^h)$ (shaded area).

PROOF: For $1 \leqslant f \leqslant h \leqslant g \leqslant n$, we define $B^{(f,g)}(z^h)$ as the area of the rectangle induced by the two corner points z^h and the special reference point $z^{ref(f,g)} := \binom{z_1^{f-1}}{z_2^{g+1}}$ with $z_1^0 = z_1^{ref}$ and $z_2^{n+1} = z_2^{ref}$ (compare Figure 4).

We immediately get the following three formulas:

$$EV(z^{i+1}, z^{j-1}) = EV(z^{i+2}, z^{j-2}) + B^{(i+1,j-2)}(z^{i+1}) + B^{(i+1,j-1)}(z^{j-1})$$
$$EV(z^{i+1}, z^{j-2}) = EV(z^{i+2}, z^{j-2}) + B^{(i+1,j-2)}(z^{i+1})$$
$$EV(z^{i+2}, z^{j-1}) = EV(z^{i+2}, z^{j-2}) + B^{(i+2,j-1)}(z^{j-1}).$$

Moreover, we know

$$\begin{aligned} B^{(i+2,j-1)}(z^{j-1}) &= B^{(i+1,j-1)}(z^{j-1}) - B^{(i+1,j-1)}(z^{i+1}) \cap B^{(i+1,j-1)}(z^{j-1}) \\ &= B^{(i+1,j-1)}(z^{j-1}) - w_{j-1,i+1}. \end{aligned}$$

With these, we can state the following chain:

$$\begin{aligned} c_{ij} &= EV(z^{i+1}, z^{j-1}) \\ &= EV(z^{i+2}, z^{j-2}) + B^{(i+1,j-2)}(z^{i+1}) + B^{(i+1,j-1)}(z^{j-1}) \\ &= EV(z^{i+2}, z^{j-2}) + B^{(i+1,j-2)}(z^{i+1}) + B^{(i+2,j-1)}(z^{j-1}) + w_{j-1,i+1} \\ &= EV(z^{i+1}, z^{j-2}) + EV(z^{i+2}, z^{j-1}) - EV(z^{i+2}, z^{j-2}) + w_{j-1,i+1} \\ &= c_{i,j-1} + c_{i+1,j} - c_{i+1,j-1} + w_{j-1,i+1} \qquad (7) \\ &> c_{i,j-1} + c_{i+1,j} - c_{i+1,j-1}. \end{aligned}$$

□

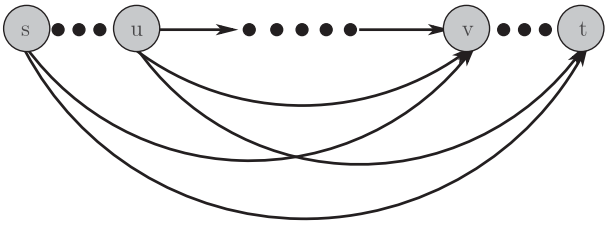


Figure 5: Selected arcs in Corollary 5.

Adapting a proof of Aggarwal and Park (1989), we can state the following equivalent property.

COROLLARY 5: *Consider the following arcs*

$$e_{st},\ e_{sv},\ e_{ut},\ e_{uv}$$

with $s < u < v < t$ *(see Figure 5). Then we have*

$$c_{st} > c_{sv} + c_{ut} - c_{uv}.$$

In particular, we get the following formula

$$c_{st} = c_{sv} + c_{ut} - c_{uv} + (z_1^u - z_1^s) \cdot (z_2^v - z_2^t) \tag{8}$$

with $z_1^0 = z_1^{ref}$ *and* $z_2^{n+1} = z_2^{ref}$.

PROOF: From formula (7), we get

$$c_{ij} + c_{i+1,j-1} = c_{i,j-1} + c_{i+1,j} + w_{j-1,i+1}$$

for all $i, j \in \{0, \ldots, n+1\}$ with $j > i + 2$. Thus, for $j > u + 1$,

$$\sum_{i=s}^{u-1} \left(c_{ij} + c_{i+1,j-1}\right) = \sum_{i=s}^{u-1} \left(c_{i,j-1} + c_{i+1,j} + w_{j-1,i+1}\right),$$

which will give us, by canceling identical terms,

$$c_{sj} + c_{u,j-1} = c_{s,j-1} + c_{uj} + \sum_{i=s}^{u-1} w_{j-1,i+1}.$$

Summation over j yields

$$\sum_{j=v+1}^{t} \left(c_{sj} + c_{u,j-1}\right) = \sum_{j=v+1}^{t} \left(c_{s,j-1} + c_{uj} + \sum_{i=s}^{u-1} w_{j-1,i+1}\right)$$

implying

$$c_{st} + c_{uv} = c_{sv} + c_{ut} + \sum_{j=v+1}^{t} \sum_{i=s}^{u-1} w_{j-1,i+1}.$$

Note that the summation index j fulfills $j \geqslant v + 1 > u + 1$. The term

$$\sum_{j=v+1}^{t} \sum_{i=s}^{u-1} w_{j-1,i+1}$$

Algorithm 2 Calculation of the costs c_{0v} and $c_{v,n+1}$

Input: $N = \{z^1, \ldots, z^n\}$
Output: The costs c_{0v} and $c_{v-1,n+1}$ for all nodes $v \in \{1, \ldots, n+1\}$.
1: $c_{01} := 0$
2: $c_{n,n+1} := 0$
3: **for** $v = 1, \ldots, n$ **do**
4: $\quad c_{0,v+1} := c_{0v} + (z_1^v - z_1^{ref}) \cdot (z_2^v - z_2^{v+1})$
5: **for** $v = n, \ldots, 2$ **do**
6: $\quad c_{v-1,n+1} = c_{v,n+1} + (z_1^v - z_1^{v-1}) \cdot (z_2^v - z_2^{ref})$

is identical to the area of the rectangle

$$\left\{ z \in \mathbb{R}^2 : \begin{pmatrix} z_1^s \\ z_2^t \end{pmatrix} \leqq z \leqq \begin{pmatrix} z_1^u \\ z_2^v \end{pmatrix} \right\},$$

proving formula (8).

REMARK 2: For our later algorithm, we can use formula (8) to calculate each cost c_{uv} on the fly. By setting $s = 0$ and $t = n + 1$, we get for $0 < u < v < n + 1$

$$c_{uv} = c_{0v} + c_{u,n+1} - c_{0,n+1} + (z_1^u - z_1^{ref}) \cdot (z_2^v - z_2^{ref}).$$

Hence, we only need to precompute all costs c_{0v} and $c_{u,n+1}$ for all $0 \leq u < v \leq n + 1$, which can be done in $\mathcal{O}(n)$ time (see Algorithm 2).

Going back to Algorithm 1, to find for a fixed $\ell \leqslant k$ the new entries $D(\ell, v)$, $v = \ell, \ldots, n + 1 - k + \ell$, we have to find in a matrix M^ℓ, where only the entries $M^\ell(u, v) := M^\ell_{uv} := D(\ell - 1, u) + c_{uv}$, $v \in \{\ell, \ldots, n + 1 - k + \ell\}$, $u \in \{\ell - 1, \ldots, v - 1\}$ are relevant, for each column v the minimal value, which is then assigned to $D(\ell, v)$. Ignoring all irrelevant columns and rows, matrix M^ℓ can be considered a square matrix in $\mathbb{R}^{(n-k+2)\times(n-k+2)}$. For an efficient searching algorithm that finds these minimal values for each column, we first need to define the notion of a totally monotone matrix.

DEFINITION 5 (TOTALLY MONOTONE MATRIX): *Consider a matrix $A \in \mathbb{R}^{p\times q}$. For a column $1 \leqslant j \leqslant q$, let min(j) denote the index of the greatest row containing the minimum value of column j. Matrix A is called monotone if $1 \leqslant j_1 < j_2 \leqslant q$ implies min(j_1) $\leqslant$ min(j_2). Moreover, matrix A is called totally monotone if each submatrix is monotone.*

THEOREM 6 (MATRIX-SEARCHING ALGORITHM) (Aggarwal et al., 1987): *Let $A \in \mathbb{R}^{p\times q}$, $p \geqslant q$, denote a totally monotone matrix. Then the matrix-searching algorithm of Aggarwal et al. (1987) finds the minimal entries in all columns in $\mathcal{O}(p)$ time.*

The matrix-searching algorithm considers recursively submatrices while reducing in each iteration, with the help of clever comparisons of selected entries, the currently considered submatrix to a square matrix where irrelevant rows are deleted.

THEOREM 7: *M^ℓ is totally monotone for a fixed $\ell \in \{2, \ldots, k\}$.*

PROOF: Choose some arbitrary submatrix with rows $i_1, i_2, \ldots, i_s$, w.l.o.g. without empty columns.

Choose two columns from the submatrix $j_1 < j_2$. Suppose now that $\min(j_1) > \min(j_2)$ and let $i_g := \min(j_2)$ and $i_h := \min(j_1)$.

Then, we get the four entries

$$M^\ell(i_g, j_1) = D(\ell - 1, i_g) + c_{i_g, j_1}, \qquad M^\ell(i_g, j_2) = D(\ell - 1, i_g) + c_{i_g, j_2},$$
$$M^\ell(i_h, j_1) = D(\ell - 1, i_h) + c_{i_h, j_1}, \qquad M^\ell(i_h, j_2) = D(\ell - 1, i_h) + c_{i_h, j_2},$$

and because of our assumption, we know $M^\ell(i_h, j_2) > M^\ell(i_g, j_2)$ and moreover $M^\ell(i_g, j_1) \geqslant M^\ell(i_h, j_1)$, i.e., we have

$$D(\ell - 1, i_h) + c_{i_h, j_2} > D(\ell - 1, i_g) + c_{i_g, j_2}$$
$$-D(\ell - 1, i_h) - c_{i_h, j_1} \geqslant -D(\ell - 1, i_g) - c_{i_g, j_1},$$

which gives us summed up the result:

$$c_{i_g, j_2} - c_{i_g, j_1} < c_{i_h, j_2} - c_{i_h, j_1}. \tag{9}$$

Furthermore, we are now in the situation $i_g < i_h < j_1 < j_2$. From Corollary 5, we immediately get

$$c_{i_g, j_2} - c_{i_g, j_1} > c_{i_h, j_2} - c_{i_h, j_1}.$$

This leads together with expression (9) to a contradiction and we get $\min(j_1) \leqslant \min(j_2)$. □

COROLLARY 8: *Using the matrix-searching algorithm from Theorem 6 in the DP approach (see Algorithm 1), the k-link shortest path problem in the constructed digraph can be solved in $\mathcal{O}(k(n-k))$ time.*

Hence, incorporating the idea from Remark 2 and the complexity for the sorting mentioned at the end of Section 2, we can state the following result.

THEOREM 9: *The two-dimensional HSSP can be solved in $\mathcal{O}(k(n-k) + n \log n)$ time.*

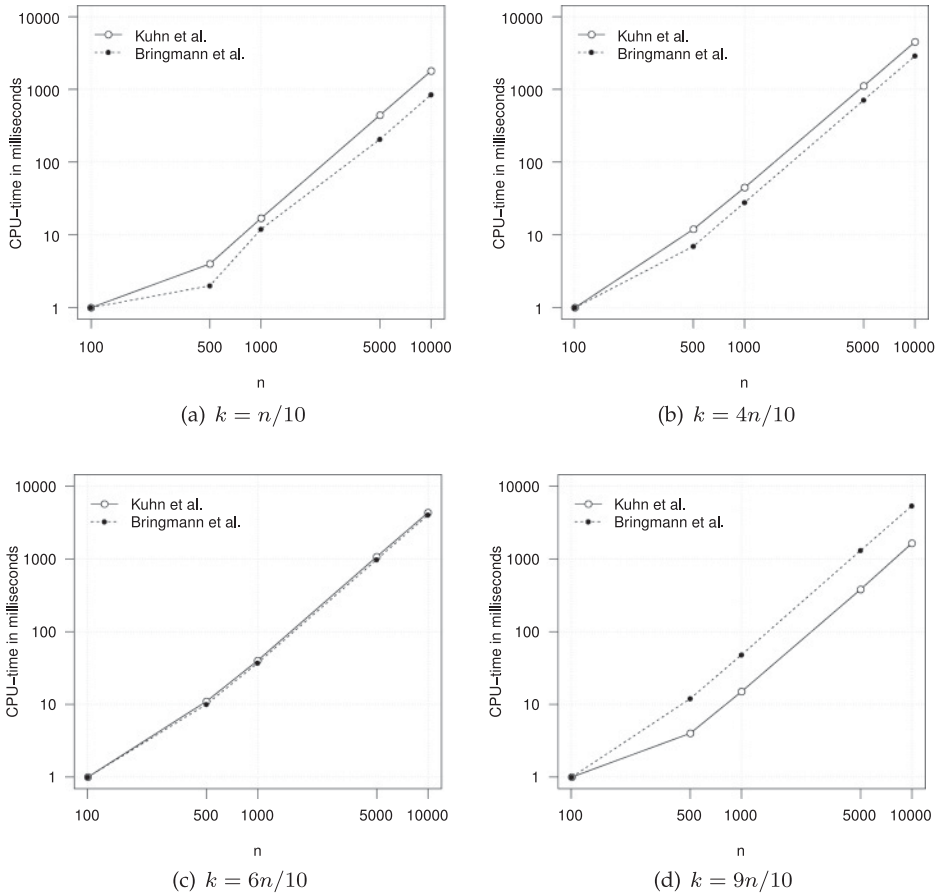
REMARK 3: Note that a simple modification of the algorithm induced from Theorem 9 can obtain the solution of the HSSP for all $k = 1, \ldots, n$ in $\mathcal{O}(n^2)$ time.

6 Experimental Analysis

The algorithm from Section 5 has been implemented and is available at https://eden.dei.uc.pt/~paquete/HSSP. We ran our approach on randomly generated instances in order to study the influence of n and k on its performance. For comparison purposes, we also ran the implementation described in Bringmann et al. (2014b).[1] Both programs were written in Java and used the same data structures as much as possible. The instances were generated according to the description given in Bringmann et al. (2014b). Each instance was run three times before measuring the running time in order to eliminate fluctuations. We considered $n \in \{100, 500, 1000, 5000, 10000\}$ and $k \in \{n/10, 2n/10, \ldots, 9n/10\}$. For each pair (n, k), we ran each implementation on 10 different random instances. All experiments were conducted on a computer with Intel Pentium processor, running at 2.0 GHz with 3 GB RAM and 64-bit Linux (ArchLinux, kernel 3.18.5). Both implementations were compiled with OpenJDK version 1.7.0_75.

The plots of Figure 6 show the computational results obtained with both approaches for different values of n and fixed ratio k/n, in a log-log scale. The plots indicate that the running time of both approaches increases in a very similar way as the instance size

[1] Code available at http://docs.theinf.uni-jena.de/code/ssp.zip



(a) $k = n/10$ (b) $k = 4n/10$

(c) $k = 6n/10$ (d) $k = 9n/10$

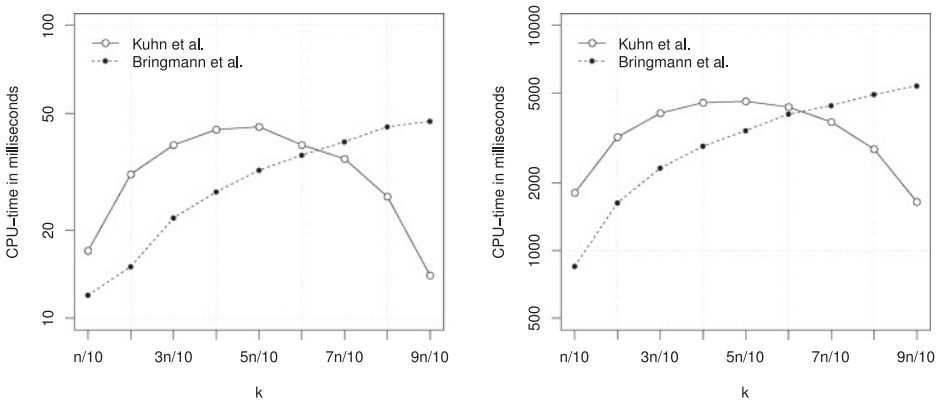
Figure 6: Average CPU time in milliseconds taken by both approaches for fixed ratio k/n.

grows, which is expected because of their similar time complexity. Noteworthy, our approach performs better as k becomes closer to n, as suggested by the time complexity of our approach; see plots (c) and (d). This observation becomes clearer in Figure 7, for different values of k and fixed n, which shows that our approach performs better for $k > 6n/10$.

7 Conclusion

In this article, we considered the two-dimensional hypervolume subset selection problem. Based on the investigation of structural properties of the problem, we proposed a new integer programming formulation, and showed that the polyhedron of its linear programming relaxation is integral. Moreover, we developed a k-link shortest path formulation on a simple, directed acyclic graph. Exploiting the special structure of the arc costs, we stated a dynamic programming approach which solves the HSSP in $\mathcal{O}(k(n-k) + n \log n)$.

It is worth noting that the HSSP IP formulation (IP_k) from Section 4 can be extended to an arbitrary number of dimensions $d > 2$. In this case, any appropriate IP solver can



(a) $n = 1000$ (b) $n = 10000$

Figure 7: Average CPU time in milliseconds taken by both approaches for fixed n.

be used to obtain an exact solution, even if the corresponding LP formulation will not, in general, define an integral polyhedron, as in the following example:

$$z^1 = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \qquad z^2 = \begin{pmatrix} 2 \\ 1 \\ 3.1 \end{pmatrix} \qquad z^3 = \begin{pmatrix} 2.1 \\ 2.1 \\ 2 \end{pmatrix} \qquad z^4 = \begin{pmatrix} 2.2 \\ 3 \\ 1 \end{pmatrix}.$$

Here, the solution of the linear programming relaxation (w.r.t. $z^{ref} = 0$) has objective value 11.31, whereas the solution of the integer program has objective value 11.02. Nevertheless, the linear programming relaxation can still be used to obtain an upper bound on the optimal hypervolume indicator.

Similarly, the graph construction in Section 5 cannot be applied to the three-dimensional case either, since the problem can no longer be easily transformed into a k-link shortest path problem. This can be observed in the following example. Let us consider three points:

$$z^1 = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}, \qquad z^2 = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}, \quad \text{and} \quad z^3 = \begin{pmatrix} 3 \\ 3 \\ 1 \end{pmatrix}.$$

Looking at the corresponding dominated regions/boxes (w.r.t. $z^{ref} = 0$), one can infer that each pair out of the three induced boxes possesses a nonempty intersection belonging only to the two boxes considered. Hence, there is no unique sorting of the points as in the two-dimensional case. Nevertheless, suppose that we assign an arbitrary order to the three nodes in the corresponding digraph with five nodes (including source and target nodes). Clearly, to model the subset selection corresponding to all points except one, the arc jumping over this node must have cost equal to the exclusive volume of the corresponding point. Then, the path corresponding to the subset including only the midpoint (w.r.t. the digraph nodes) will have the wrong weight, since it contains only the two arcs that jump over the second and second to last nodes, respectively. We would miss subtracting the part of the exclusive volume of the two points not selected that is dominated by both. On the other hand, with two objectives, the application of the obtained formulations and algorithms to other indicators such as, for instance, the

generalized hypervolume indicator introduced by Grunert da Fonseca and Fonseca (2012), should follow naturally.

Finally, we remark that the hypervolume indicator can also be used as a measure of the quality of discrete representations of an optimal set of nondominated points (see Sayın, 2000; Ruzika and Wiecek, 2005). Given a large nondominated set in the biobjective case, the algorithm described in this article can be used to find an optimal cardinality-constrained representation with respect to the hypervolume indicator very efficiently (see also Bringmann et al., 2014a).

Acknowledgments

The authors gratefully acknowledge support under the bilateral cooperation research project "Tractability in multiobjective combinatorial optimization," funded by the Deutscher Akademischer Austausch Dienst and Fundação para a Ciência e a Tecnologia. Authors Kuhn and Ruzika acknowledge the support of the Federal Ministry of Education and Research Germany, grant DSS_Evac_Logistic, FKZ 13N12229. Fonseca and Paquete acknowledge the iCIS project (CENTRO-07-ST24-FEDER-002003), which is co-financed by QREN, in the scope of the Mais Centro Program and European Union's FEDER. Fonseca, Kuhn, and Paquete also acknowledge the Lorentz Center Workshop on Set-Oriented and Indicator-Based Multi-Criteria Optimization (SIMCO 2013), where some of the results presented in this work were developed. Finally, the authors acknowledge Karl Bringmann for pointing out the result in Bringmann et al. (2014b) as well as the technique mentioned in Remark 2.

References

Aggarwal, A., Klawe, M. M., Moran, S., Shor, P., and Wilber, R. (1987). Geometric applications of a matrix-searching algorithm. *Algorithmica*, 2(1–4): 195–208.

Aggarwal, A., and Park, J. (1989). Sequential searching in multidimensional monotone arrays. (Technical Report) Massachusetts Institute of Technology, Cambridge, MA.

Aggarwal, A., Schieber, B., and Tokuyama, T. (1994). Finding a minimum-weight *k*-link path in graphs with the concave Monge property and applications. *Discrete and Computational Geometry*, 12(1): 263–280.

Bader, J. (2009). *Hypervolume-based search for multiobjective optimization: Theory and methods*. (Unpublished doctoral dissertation) Eidgenössische Technische Hochschule, ETH, Zürich.

Bader, J., and Zitzler, E. (2011). HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1): 45–76.

Bazaraa, M. S., Jarvis, J. J., and Sherali, H. D. (2011). *Linear programming and network flows*. New York: Wiley.

Beume, N., Naujoks, B., and Emmerich, M. (2007). SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3): 1653–1669.

Bringmann, K., and Friedrich, T. (2009). Approximating the least hypervolume contributor: NP-hard in general, but fast in practice. In *Evolutionary Multi-Criterion Optimization*, pp. 6–20. Lecture Notes in Computer Science, Vol. 5467.

Bringmann, K., and Friedrich, T. (2010). An efficient algorithm for computing hypervolume contributions. *Evolutionary Computation*, 18(3): 383–402.

Bringmann, K., Friedrich, T., and Klitzke, P. (2014a). Generic postprocessing via subset selection for hypervolume and epsilon-indicator. In *Parallel Problem Solving from Nature*, pp. 518–527. Lecture Notes in Computer Science, Vol. 5467.

Bringmann, K., Friedrich, T., and Klitzke, P. (2014b). Two-dimensional subset selection for hypervolume and epsilon-indicator. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pp. 589–596.

Emmerich, M., and Fonseca, C. M. (2011). Computing hypervolume contributions in low dimensions: Asymptotically optimal algorithm and complexity results. In *Evolutionary Multi-Criterion Optimization*, pp. 121–135. Lecture Notes in Computer Science, Vol. 8672.

Fleischer, M. (2003). The measure of Pareto optima: Applications to multi-objective metaheuristics. In *Evolutionary Multi-Criterion Optimization*, pp. 519–533. Lecture Notes in Computer Science, Vol. 2632.

Grunert da Fonseca, V., and Fonseca, C. (2012). The relationship between the covered fraction, completeness and hypervolume indicators. In *Artificial Evolution*, pp. 25–36. Lecture Notes in Computer Science, Vol. 7401.

Huband, S., Hingston, P., While, L., and Barone, L. (2003). An evolution strategy with probabilistic mutation for multi-objective optimisation. In *Proceedings of the Congress on Evolutionary Computation*, pp. 2284–2291.

Igel, C., Hansen, N., and Roth, S. (2007). Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation*, 15(1): 1–28.

Knowles, J., and Corne, D. (2002). On metrics for comparing nondominated sets. In *Proceedings of the Congress on Evolutionary Computation*, pp. 711–716.

Knowles, J., and Corne, D. (2003). Properties of an adaptive archiving algorithm for storing nondominated vectors. *IEEE Transactions on Evolutionary Computation*, 7(2): 100–116.

Knowles, J., Corne, D., and Fleischer, M. (2003). Bounded archiving using the Lebesgue measure. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 2490–2497.

Knowles, J., Corne, D., and Oates, M. (2000). On the assessment of multiobjective approaches to the adaptive distributed database management problem. In *Parallel Problem Solving from Nature*, pp. 869–878. Lecture Notes in Computer Science, Vol. 1917.

Nemhauser, G. L., and Wolsey, L. A. (1999). *Integer and combinatorial optimization*. New York: Wiley.

Ruzika, S., and Wiecek, M. (2005). Approximation methods in multiobjective programming. *Journal of Optimization Theory and Applications*, 126(3): 473–501.

Sayın, S. (2000). Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. *Mathematical Programming*, 87(3): 543–560.

Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2): 173–195.

Zitzler, E., Knowles, J., and Thiele, L. (2008). Quality assessment of Pareto set approximations. In *Multiobjective Optimization, Interactive and Evolutionary Approaches*, pp. 373–404. Lecture Notes in Computer Science, Vol. 5252.

Zitzler, E., and Künzli, S. (2004). Indicator-based selection in multiobjective search. In *Parallel Problem Solving from Nature*, pp. 832–842. Lecture Notes in Computer Science, Vol. 3242.

Zitzler, E., and Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms: A comparative case study. In *Parallel Problem Solving from Nature*, pp. 292–301. Lecture Notes in Computer Science, Vol. 1498.

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