Hypervolume Subset Selection in Two Dimensions: Formulations and Algorithms

Tobias Kuhn* Carlos M. Fonseca† Luís Paquete†
Stefan Ruzika‡ José Rui Figueira§

May 14, 2014

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 $\mathcal{O}(n(k + \log n))$ time. This improves upon the $\mathcal{O}(n^2k)$ result of Bader [4], and matches the recent result of Bringmann et al. [10], which was developed independently from this work using different techniques. Moreover, it is shown that these bounds may be further improved under mild conditions on k.

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 [24]. Due to its properties [15, 25, 22], this indicator has been instrumental both in the assessment of the performance of multiobjective evolutionary algorithms [18, 21] and in the

*Mathematical Institute, University of Kaiserslautern, Germany
†CISUC, Department of Informatics Engineering, University of Coimbra, Portugal
‡Mathematical Institute, University of Koblenz-Landau, Campus Koblenz, Germany
§CEG-IST, Instituto Superior Técnico, Universidade de Lisboa, Portugal

development of multiobjective selection and archiving procedures for such algorithms [12, 17, 16, 23, 7, 5].

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 [5]. 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 [13, 7, 14]; see also approximation results by Bringmann and Friedrich [8] 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 [11].

So far, the tightest time complexity bound for solving the HSSP to optimality for $d > 2$ dimensions and arbitrary k is $\mathcal{O}(n^{d/2} \log n + n^k)$ [9]. For the particular case of $d = 2$, to the best of our knowledge, only two approaches have been proposed. Bader [4] 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 left-most point of a given nondominated subset depends only on its immediate neighbor. Very recently and independently from the present work, Bringmann et al. [10] proposed another approach to the same problem with a time complexity of $\mathcal{O}(n(k + \log n))$. This algorithm computes, in the ℓ-th 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 $\mathcal{O}(n(k + \log n))$ time, which matches the result of Bringmann et al. [10]. Moreover, by taking into account known results from the literature about the k-link shortest path problem, in particular, the algorithms of Aggarwal et al. [3] and Schieber [20], we show that it is even possible to improve this bound under mild conditions on k. In addition, if a parallel environment is available, the HSSP can be solved in $\mathcal{O}(n\sqrt{k \log n})$ [3].

The remainder of this article is organized as follows. In Section 2, we introduce concepts, definitions, notation and some basic results. 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 improved complexity bounds for the HSSP. Finally, in Section 6 we provide some conclusions and avenues for future research.

2 Terminology and Basic Results

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 $\mathbb{R}^q$:

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

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.

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

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 $\{0, 1, -1\}$.

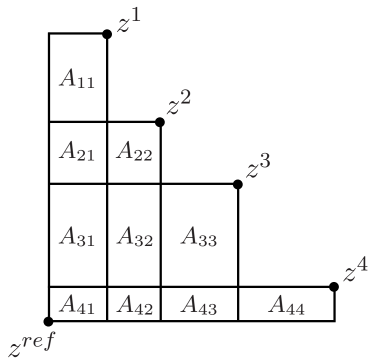


Figure 1: Partition of the dominated region for a given set $N = \{z^1, z^2, z^3, z^4\}$

Theorem 5 (Integrality [19]): *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 non-empty faces contains an integral point.*

Definition 6 (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 7 (Matrix-Searching Algorithm [1]): *Let $A \in \mathbb{R}^{p\times q}$, $p \geqslant q$, denote a totally monotone matrix. Then the Matrix-Searching Algorithm in [1] finds the minimal entries in all columns in $\mathcal{O}(p)$ time.*

In the following sections, we assume a set of nondominated points $N := \{z^1, \dots, z^n\} \subseteq \mathbb{R}^2$, some reference point z^{ref} and a desired cardinality $k \in \{1, \dots, n\}$ to be given. We further assume that the points in set N are sorted in increasing order of the first component, i.e. $z^i_1 < z^j_1$ 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 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 $\{z^j, \dots, z^i\}$ 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 $\lambda(A_{ij})$ of rectangle A_{ij}. If we define $z^0_1 := z^{ref}_1$ and $z^{n+1}_2 := z^{ref}_2$ 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.

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. Following the notation in Section 3, we denote with A_{ij}, $i \geqslant j$, 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:

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

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

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

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

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.

Consider now the linear programming (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:

$$
\begin{aligned}
(LP_k) \qquad & \max \sum_{i=1}^{n}\sum_{j=1}^{i} w_{ij}x_{ij} && && (4)\\
& \text{subject to } \sum_{\ell=1}^{n} x_{\ell\ell} = k && && (5)\\
& \sum_{\ell=j}^{i} x_{\ell\ell} - x_{ij} - s_{ij} = 0 && i=2,\dots,n;\ j=1,\dots,i-1 && (6)\\
& 0 \leqslant x_{ij} \leqslant 1 && i=1,\dots,n;\ j=1,\dots,i &&\\
& s_{ij} \geqslant 0 && i=2,\dots,n;\ j=1,\dots,i-1 &&
\end{aligned}
$$

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

$$
\left(\begin{array}{ccc|ccc|ccc}
1 & \dots & 1 & 0 & \dots & 0 & 0 & \dots & 0 \\
\hline
 & C & & & \text{-}I & & & \text{-}I &
\end{array}\right)
$$

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 $\begin{pmatrix} e \\ C \end{pmatrix}$, where $e \in \mathbb{R}^n$ is the vector of all ones, and by D the submatrix $\left(\begin{array}{ccc|ccc} 0 & \dots & 0 & 0 & \dots & 0 \\ \hline & \text{-}I & & & \text{-}I & \end{array}\right)$.
Observe that D is obviously totally unimodular and $\tilde{C}$ has the *consecutive ones property* [19] and thus is also totally unimodular.

Theorem 8: *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 j-th 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} corresponds also to a squared submatrix of the constraint matrix and 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 9 (Integrality): *The polyhedron corresponding to* (LP_k) *is integral.*

Proof:
This follows from theorems 5 and 8 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

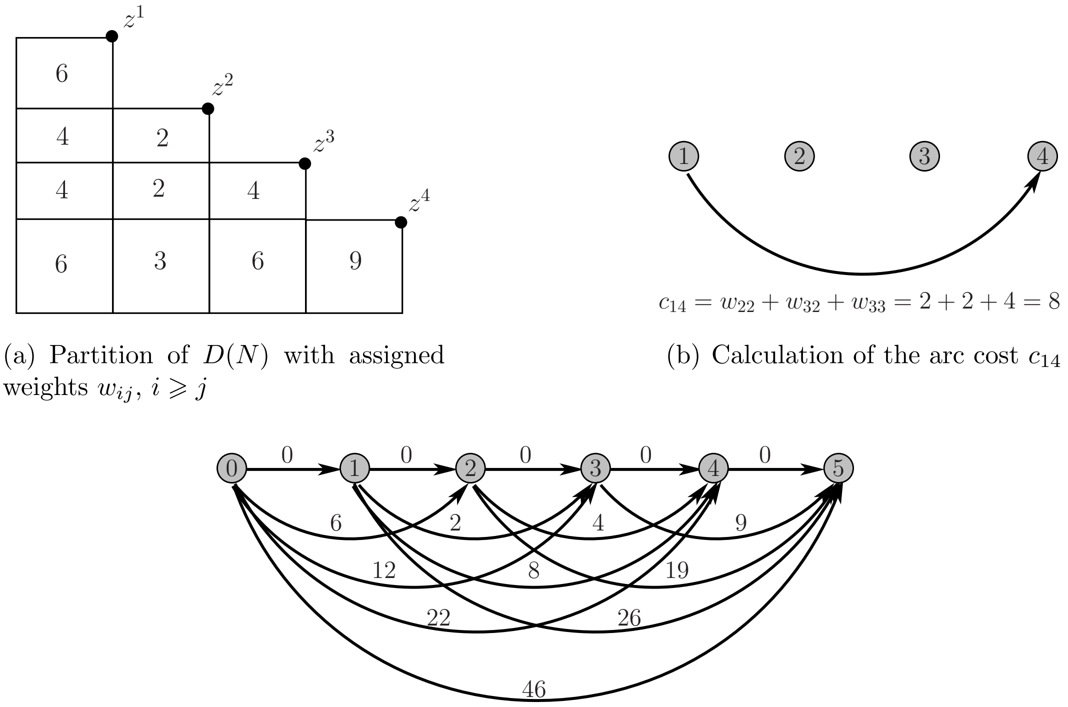
In the following, we show that the HSSP can be modeled using a k-link shortest path formulation in a directed graph (digraph). The corresponding shortest path problem with a cardinality constraint can then be solved using a dynamic programming (DP) approach. This digraph has a special structure, the Monge property, that allows us to solve the HSSP in $\mathcal{O}(n(k + \log n))$ time.

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 every 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 $u \in \{0, \ldots, n\}$.

The cost c_{uv} describes the contribution to the hypervolume indicator of the whole set $\{z^{u+1}, \ldots, z^{v-1}\}$, which will be called 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.



(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

Observation 10:
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 an 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 obviously 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 to a better running time than Bader's DP algorithm, since finding the minimum in line 4 is in a naïve way done in $\mathcal{O}(n)$, resulting in an overall running time in $\mathcal{O}(n^2k)$.

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* [3]:

Theorem 11 (Monge property): *Consider the following arcs*

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

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

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

```
D(1, v) := c_{0v} for all v ∈ {1, ..., n − k + 2}
for ℓ = 2, ..., k do
    for v = ℓ, ..., n + 1 − k + ℓ do
        D(ℓ, v) := min_{u=ℓ−1,...,v−1} {D(ℓ − 1, u) + c_{uv}}
```

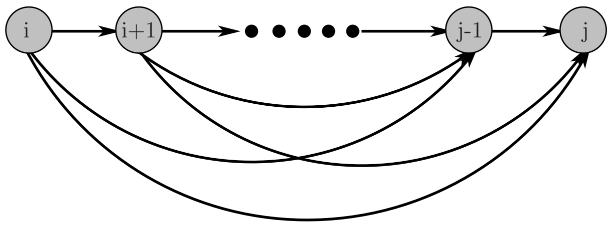


Figure 3: Selected arcs in Theorem 11

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

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

Proof:
For $0 \leqslant f \leqslant h \leqslant g \leqslant n$ 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 $z^{ref(f,g)} := \begin{pmatrix} z_1^{f-1} \\ z_2^{g+1} \end{pmatrix}$ 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:

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

Moreover, we know:

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

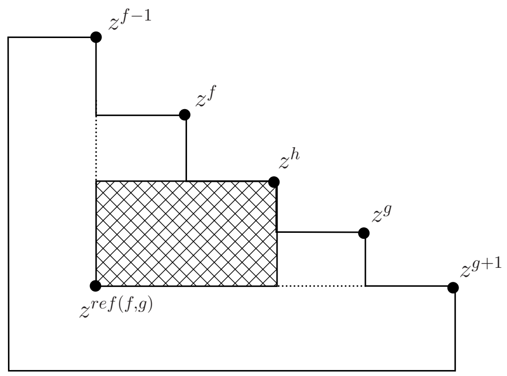


Figure 4: Example for $B^{(f,g)}(z^h)$ (shaded area)

With these we can state the following chain:

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

Adapting a proof of [2], we can state the following equivalent property.

Corollary 12: *Consider the following arcs*

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

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

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

In particular, we get the following formula

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

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 $j > u + 1$,

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

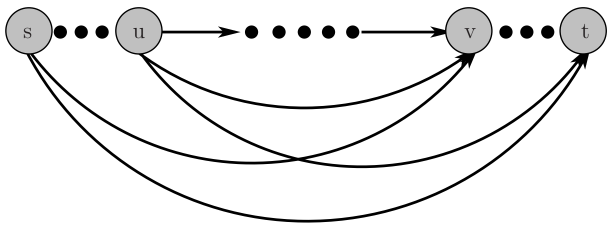


Figure 5: Selected Arcs in Corollary 12

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

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

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

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

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). □

Observation 13:
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 $0 < u < v < n + 1$, which can be done in $\mathcal{O}(n)$ time (see Algorithm 2).

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

Input: $N = \{z^1, \dots, z^n\}$

Output: The costs c_{0v} and $c_{v-1,n+1}$ for all nodes $v \in \{1, \dots, n+1\}$.

$c_{01} := 0$

$c_{n,n+1} := 0$

for $v = 1, \dots, n$ **do**

  $c_{0,v+1} := c_{0v} + (z_1^v - z_1^{ref}) \cdot (z_2^v - z_2^{v+1})$

for $v = n, \dots, 2$ **do**

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

Going back to Algorithm 1, to find for a fixed $\ell \leqslant k$ the new entries $D(\ell, v)$, $v = \ell, \dots, n+1-k+\ell$, we have to find in a matrix M^ℓ, where only the entries $M^\ell(u,v) := M^\ell_{uv} := D(\ell-1, u) + c_{uv}$, $v \in \{\ell, \dots, n+1-k+\ell\}$, $u \in \{\ell-1, \dots, v-1\}$ 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 is a square matrix in $\mathbb{R}^{(n-k+2)\times(n-k+2)}$.

Theorem 14: *M^ℓ is totally monotone for a fixed $\ell \in \{2, \dots, k\}$*

Proof:
Choose some arbitrary submatrix with rows $i_1, i_2, \dots, i_s$, 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 due to 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 $i_g < i_h < j_1 < j_2$ and from Corollary 12 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 (9) to a contradiction and we get $\min(j_1) \leqslant \min(j_2)$. □

Corollary 15: *Using the Matrix-Searching Algorithm from Theorem 7 in the DP approach (see Algorithm 1), the k-link shortest path problem in the constructed digraph can be solved in $\mathcal{O}(nk)$ time.*

Hence, incorporating the idea from Observation 13 we can state the following result.

Theorem 16: *The two-dimensional HSSP can be solved in $\mathcal{O}(n(k + \log n))$ time.*

We remark that, by taking into account known results in the literature about the k-link shortest path problem on our special constructed digraph, the following results can also be stated.

Theorem 17: *The two-dimensional HSSP can be solved in*

1. $\mathcal{O}(n(\log U + \log n))$ *time, if the costs are integral and* U *denotes the largest cost [3].*
2. $n2^{\mathcal{O}(\sqrt{\log k \log \log n})} + \mathcal{O}(n \log n)$ *time, if* $k = \Omega(\log n)$ *[20].*

Note that both results stated in the theorem above are an improvement over the bound stated in Theorem 16 if $k = \Omega(\log U)$ and $k = \Omega(\log n)$, respectively. Finally, we state the following result in case a parallel environment, as described in [3], is available.

Theorem 18: *The two-dimensional HSSP can be solved in $\mathcal{O}(n\sqrt{k \log n})$ [3].*

Observation 19:
Note that a simple modification of the algorithm induced from Theorem 16 can obtain the solution of the HSSP for all $k = 1, \ldots, n$ in $\mathcal{O}(n^2)$ time.

6 Conclusion

In this article, we considered the two-dimensional hypervolume subset selection problem. We have proposed a new integer programming formulation and showed that the polyhedron of its linear programming relaxation is integral. Moreover, we have given 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 $\mathcal{O}(n(k + \log n))$. Further improvements can be achieved under some particular conditions.

The developed methods cannot be used for more than two dimensions. The IP formulation (IP_k) from Section 4 can be extended to more than two dimensions, in particular to the three-dimensional case for $N \subseteq \mathbb{R}^3$. However, the corresponding LP will, in general, not define an integral polyhedron and can therefore not be used to solve the IP. This can be observed for example with the following four points

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

Here, the linear programming relaxation (w.r.t. $z^{ref} = 0$) has objective value 11.31 and the integer program 11.02. Nevertheless, the linear programming relaxation can be used to obtain an upper bound on the optimal hypervolume indicator. Moreover, also the whole graph construction from Section 5 cannot be applied to the three-dimensional case, since there the problem cannot be easily transformed to a k-link shortest path problem, which can be observed in the following example. Let us consider the following 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 imply that each pair out of the three induced boxes possesses a non-empty intersection only belonging to both considered boxes. Hence, there is no unique sorting of the points as in the two-dimensional case. Nevertheless, suppose that we assign an arbitrary sorting 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 selection by choosing only the mid-point (w.r.t. the digraph nodes) will have the wrong value, since we only need two arcs, for jumping only over the second and second last node, respectively. We would miss to subtract the volume of the exclusive intersection of the two not selected points.

7 Acknowledgments

The authors gratefully acknowledge the support on the bilateral cooperation project “RepSys - Representation systems with quality guarantees for multi-objective optimization problems” founded by the Deutscher Akademischer Austausch Dienst and Conselho de Reitores das Universidades Portuguesas. The authors Kuhn and Ruzika acknowledge the support of the Federal Ministry of Education and Research Germany, grant DSS_Evac_Logistic, FKZ 13N12229. 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 [10] as well as the technique mentioned in Observation 13.

References

[1] A. Aggarwal, M. M. Klawe, S. Moran, P. Shor, and R. Wilber. Geometric Applications of a Matrix-Searching Algorithm. *Algorithmica*, 2(1-4):195–208, 1987.

[2] A. Aggarwal and J. Park. Notes on Searching in Multidimensional Monotone Arrays. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, pages 497–512, 1988.

[3] A. Aggarwal, B. Schieber, and T. Tokuyama. Finding a Minimum-Weight k-link Path in Graphs with the Concave Monge Property and Applications. *Discrete & Computational Geometry*, 12(1):263–280, 1994.

[4] J. Bader. *Hypervolume-Based Search for Multiobjective Optimization: Theory and Methods*. PhD thesis, Eidgenössische Technische Hochschule ETH Zürich, 2009.

[5] J. Bader and E. Zitzler. HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. *Evolutionary Computation*, 19(1):45–76, 2011.

[6] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali. *Linear Programming and Network Flows*. Wiley, New York, 2011.

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

[8] K. Bringmann and T. Friedrich. Approximating the Least Hypervolume Contributor: NP-Hard in General, But Fast in Practice. In M. Ehrgott, C. M. Fonseca, X. Gandibleux, J.-K. Hao, and M. Sevaux, editors, *Evolutionary Multi-Criterion Optimization*, volume 5467 of *Lecture Notes in Computer Science*, pages 6–20. Springer Berlin Heidelberg, 2009.

[9] K. Bringmann and T. Friedrich. An Efficient Algorithm for Computing Hypervolume Contributions. *Evolutionary Computation*, 18(3):383–402, 2010.

[10] K. Bringmann, T. Friedrich, and P. Klitzke. Two-dimensional Subset Selection for Hypervolume and Epsilon-Indicator. In *Proceedings of the ACM Genetic and Evolutionary Computation Conference (GECCO)*. ACM press (to appear), in press, 2014.

[11] M. Emmerich and C. M. Fonseca. Computing Hypervolume Contributions in Low Dimensions: Asymptotically Optimal Algorithm and Complexity Results. In R. H. C. Takahashi, K. Deb, E. F. Wanner, and S. Greco, editors, *Evolutionary Multi-Criterion Optimization*, volume 6576 of *Lecture Notes in Computer Science*, pages 121–135. Springer Berlin Heidelberg, 2011.

[12] M. Fleischer. The Measure of Pareto Optima: Applications to Multi-objective Metaheuristics. In C. M. Fonseca, P. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *Evolutionary Multi-criterion Optimization (EMO 2003)*, volume 2632 of *Lecture Notes in Computer Science*, pages 519–533, Berlin, Germany, 2003. Springer-Verlag.

[13] S. Huband, P. Hingston, L. While, and L. Barone. An Evolution Strategy with Probabilistic Mutation for Multi-Objective Optimisation. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC '03)*, volume 4, pages 2284–2291, Dec 2003.

[14] C. Igel, N. Hansen, and S. Roth. Covariance Matrix Adaptation for Multi-objective Optimization. *Evolutionary Computation*, 15(1):1–28, March 2007.

[15] J. Knowles and D. Corne. On Metrics for Comparing Nondominated Sets. In *CEC '02. Proceedings of the 2002 Congress on Evolutionary Computation, 2002.*, pages 711–716. IEEE, 2002.

[16] J. Knowles and D. Corne. Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors. *IEEE Transactions on Evolutionary Computation*, 7(2):100–116, April 2003.

[17] J. Knowles, D. Corne, and M. Fleischer. Bounded Archiving using the Lebesgue Measure. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC2003)*, volume 4, pages 2490–2497, New York, NY, USA, 2003. IEEE Press.

[18] J. Knowles, D. Corne, and M. Oates. On the Assessment of Multiobjective Approaches to the Adaptive Distributed Database Management Problem. In M. Schoenauer et al., editors, *Parallel Problem Solving from Nature (PPSN VI)*, volume 1917 of *Lecture Notes in Computer Science*, pages 869–878. Springer-Verlag, Berlin, Germany, 2000.

[19] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, New York, 1999.

[20] B. Schieber. Computing a Minimum Weight k-Link Path in Graphs with the Concave Monge Property. *Journal of Algorithms*, 29:204–222, 1999.

[21] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, 2000.

[22] E. Zitzler, J. Knowles, and L. Thiele. Quality Assessment of Pareto Set Approximations. In J. Branke et al., editors, *Multiobjective Optimization, Interactive and Evolutionary Approaches*, volume 5252 of *Lecture Notes in Computer Science*, pages 373–404. Springer-Verlag, Berlin, Germany, 2008.

[23] E. Zitzler and S. Künzli. Indicator-Based Selection in Multiobjective Search. In X. Yao et al., editors, *Parallel Problem Solving from Nature (PPSN VIII)*, volume 3242 of *Lecture Notes in Computer Science*, pages 832–842. Springer-Verlag, Berlin, Germany, 2004.

[24] E. Zitzler and L. Thiele. Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In A. E. Eiben, T. Bäck, M. Schoenauer, and

H.-P. Schwefel, editors, *Conference on Parallel Problem Solving from Nature (PPSN V)*, volume 1498 of *Lecture Notes in Computer Science*, pages 292–301. Springer Berlin Heidelberg, 1998.

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