Hypervolume Subset Selection in Two Dimensions: Formulations and Algorithms

Tobias Kuhn[*]   Carlos M. Fonseca[†]   Luís Paquete[†]

Stefan Ruzika[‡]   José Rui Figueira[§]

March 30, 2014

Abstract. The hypervolume subset selection problem consists of finding a subset, with a given cardinality, of a nondominated set of points that maximizes the hypervolume indicator. This problem arises in selection procedures of population-based heuristics for multiobjective optimization, and for which practically efficient algorithms are strongly required. In this article, we provide two new formulations for the two-dimensional variant of this problem. The first is an integer programming formulation that can be solved by solving its linear relaxation. The second formulation is a k-link shortest path formulation on a special digraph with Monge property that can be solved by dynamic programming in $\mathcal{O}(n^2)$ time complexity. This improves upon the existing result of $O(n^3)$ in Bader [4].

1 Introduction

The hypervolume subset selection problem (HSSP) is defined as the problem of finding a subset in a set of nondominated points that has a predefined cardinality and maximizes the hypervolume indicator. This indicator measures the dominated region in the corresponding space bounded by some reference point. For the two-dimensional case the best known algorithm is the dynamic programming (DP) approach described in [4] with a time complexity of $\mathcal{O}(n^3)$. Such a selection procedure can be used by population-based

---

[*]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, University of Lisbon, Portugal

heuristic approaches for multiobjective optimization problems, which aim to maximize the hypervolume indicator (e.g. in [7, 5, 8]). In each iteration a pool of solutions is determined and a subset of these solutions with fixed cardinality is kept for further iterations. Hence, fast subset selection procedures can enhance the running time of such heuristic approaches (see [10, 11] for the case of the ϵ-indicator).

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 from the set of nondominated points. The main result is the formulation of the HSSP as a k-link shortest path problem. In this formulation, the arc costs possess a special property, called Monge property, that allows us to solve the HSSP in $\mathcal{O}(n^2)$ time, improving upon the $\mathcal{O}(n^3)$ complexity bound in Bader [4].

The remainder of this article is organized as follows. In Section 2, we introduce concepts, their definitions, notation and some basic results. In Section 3, we briefly review Bader's dynamic programming algorithm. In Section 4, we explain the crucial preprocessing step for calculating the weights used in the two formulations. In Section 5, we introduce an integer programming formulation for the HSSP and prove the integrality of the polyhedron of its linear programming relaxation. In Section 6, we present the k-link shortest path formulation, which will be used to improve the known complexity bound. Finally, in Section 7 we provide some conclusions and avenues for future research.

2 Terminology and Basic Results

In the following, some concepts, their definitions and the notation used in this article are given. Consider the two vectors z^1, $z^2 \in \mathbb{R}^q$:

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

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*, where no point in N is dominating another point in N.

Definition 2 (Hypervolume Indicator):
Let z^{ref} denote 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 that set and bounded below by a reference point.

Definition 3 (HSSP):
Consider $k \in \{1, \ldots, n\}$. 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 $\{0, 1, -1\}$.

Theorem 5 (Integrality [9]): *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 [3]): *Let $A \in \mathbb{R}^{p \times q}$, $p \geqslant q$, denote a totally monotone matrix. Then the Matrix-Searching Algorithm in [3] finds the minimal entries in all columns in $\mathcal{O}(p)$ time.*

In the following sections, we assume a set of nondominated points $N := \{z_1, \ldots, z_n\} \subseteq \mathbb{R}^2$ with $z_1^i < z_1^j$ for $i < j$, some reference point z^{ref} and a desired cardinality $k \in \{1, \ldots, n\}$.

3 Bader's Dynamic Programming Algorithm

Bader [4] proposes a DP algorithm for the two-dimensional case of the HSSP with $\mathcal{O}(n^3)$-time complexity. This algorithm is based on the principle that for the given nondominated set N the contribution to the hypervolume indicator of the left-most point z^1 only depends on its direct neighbor z^2.

For a given $z^c \in N$ and cardinality $t \in \{1, \ldots, k\}$, let P_c^t denote the HSSP subproblem of finding the subset of $\{z^c, z^{c+1}, \ldots, z^n\}$ with cardinality t that has the maximum hypervolume indicator and contains z^c. Moreover, we denote with $N_c^t \subseteq N$ an optimal subset for the subproblem P_c^t.

Algorithm 1 Bader's DP algorithm for the HSSP

Input: $N \subseteq \mathbb{R}^2$, k
Output: Solution of the HSSP with cardinality k
1: $N_c^1 := \{z^c\}$ for all $c \in \{1, \ldots, n\}$
2: **for** $t = 2, \ldots, k$ **do**
3:     **for** $c = 1, \ldots, n - t + 1$ **do**
4:         $S(N_c^t) := \max\limits_{d > c} S(N_d^{t-1} \cup \{z^c\})$
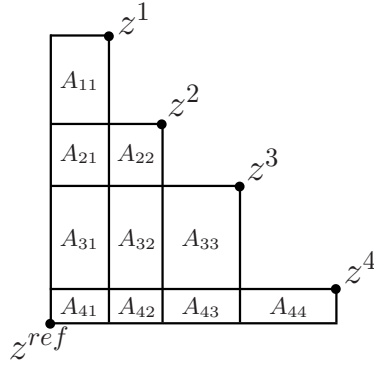   **return** $\max\limits_{c \leqslant n-k+1} S(N_c^k)$

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

The pseudocode of Bader's DP algorithm is described in Algorithm 1. For $t > 1$ and a given index c, the algorithm computes N_c^t as the subset with maximum hypervolume indicator from the sets $N_d^{t-1} \cup \{z^c\}$, for $d = c+1, \ldots, n-t+2$. The final output is then given by the set from N_c^k for all $c \leqslant n - k + 1$ with the largest hypervolume indicator.

Since for each $t = 2, \ldots, k$ all points $z^c \in N$ have to be combined with each set N_d^{t-1}, $d > c$, the overall running time needed to calculate the optimal solution for the HSSP is in $\mathcal{O}(n^3)$.

4 Preprocessing: Decomposition of the Dominated Region

This section is devoted to the preprocessing step, which is the crucial aspect for our new formulations. 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 $\lambda(A_{ij})$ 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.

5 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 relaxation. Following the notation in Section 4, 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:

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

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

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

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

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 constraint (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:

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

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

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

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

where the new variables s_{ij} are *surplus variables*.

If we rearrange the columns in a certain way, first the variables $x_{\ell\ell}$, $\ell = 1, \ldots, n$, and after that 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 (LP_k) is given by

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

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 $\left(\begin{array}{ccc|ccc} 0 & \ldots & 0 & 0 & \ldots & 0 \\ \hline & \text{-}I & & & \text{-}I & \end{array} \right)$.

Observe that D is obviously *totally unimodular* and $\tilde{C}$ has the consecutive ones property [9] 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 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} 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. $\qquad\square$

Corollary 9 (Integrality): *The polyhedron corresponding to (LP_k) is integral.*

Proof:
This follows from theorems 5 and 8 and the following upper bound of the surplus variables:

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

6 A k-link Shortest Path Formulation with Monge Property

In the following, we show that the HSSP can be modeled using a k-link shortest path formulation in an appropriate directed graph (digraph). The corresponding shortest path problem with a cardinality constraint can then be solved using a DP approach. This digraph has a special structure, the Monge property, that allows us to solve the HSSP problem in $O(n^2)$ time-complexity.

We first explain the construction of the corresponding digraph $G = (V, E)$ for the 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 4), 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.

Observation 10:
Each choice in the HSSP of a subset $\{z^{s_1}, \ldots, z^{s_{k-1}}\}$ out 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 $k - 1$.
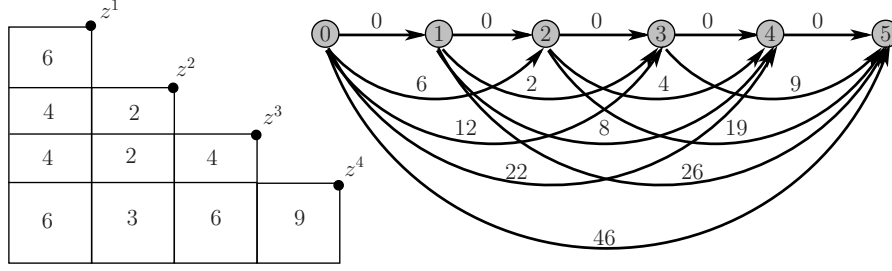
Figure 2: Example for the graph construction

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

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 2). 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 (see Section 3), since finding the minimum in line 4 is in a naïve way done in $\mathcal{O}(n)$, resulting in an overall running time in $\mathcal{O}(n^3)$.

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* [2]:

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

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

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

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

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).
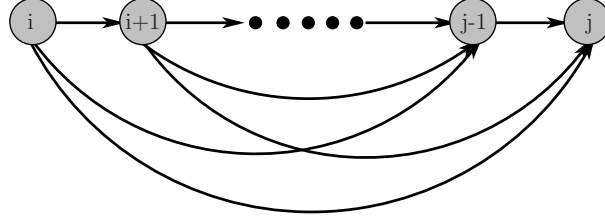
8

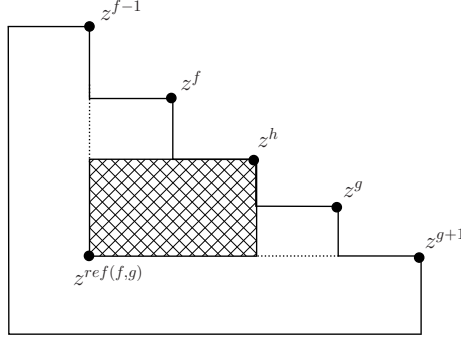Figure 3: Selected arcs in Theorem 11

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

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:

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

With these we can state the following chain:

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

Algorithm 3 Calculation of the costs c_{uv}

Input: Weights w_{ij}, $i \geqslant j$, from preprocessing
Output: The costs c_{uv} for all arcs e_{uv}, $u, v \in \{0, \dots, n+1\}$ with $u < v$.
  1: $c_{u,u+1} := 0$ for all $u \in \{0, \dots, n\}$
  2: $c_{u,u+2} := w_{u+1,u+1}$ for all $u \in \{0, \dots, n-1\}$
  3: **for** $d = 3, \dots, n+1$ **do**
  4:     **for** $u = 0, \dots, n+1-d$ **do**
  5:         $c_{u,u+d} := c_{u,u+(d-1)} + c_{u+1,u+d} - c_{u+1,u+d-1} + w_{u+d-1,u+1}$
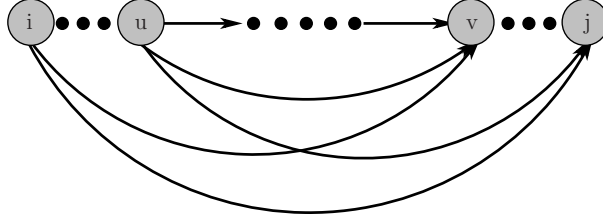---

Figure 5: Selected Arcs in Corollary 13

Observation 12:
From the proof of Theorem 11 we can design an algorithm for calculating the costs c_{uv} in $\mathcal{O}(n^2)$ time. We use formula (7) to calculate the costs in a bottom-up manner (see Algorithm 3). Note that in each iteration all costs used to calculate the cost $c_{u,u+d}$ have been calculated before, since $c_{u+1,u+d} = c_{u+1,(u+1)+(d-1)}$ and $c_{u+1,u+d-1} = c_{u+1,(u+1)+(d-2)}$.

From the Monge property it is easy to follow the following equivalent property [1].

Corollary 13: *Consider the following arcs*

$$e_{i,j}, \ e_{i,v}, \ e_{u,j}, \ e_{u,v}$$

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

$$c_{i,j} > c_{i,v} + c_{u,j} - c_{u,v}$$

Going back to Algorithm 2, to find for a fixed $\ell \leqslant k$ the new entries $D(\ell, v)$, $v = \ell, \dots, n+1$, 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 row $\ell \in \{2, \dots, k\}$*

10

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 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:

$$c_{i_g, j_2} - c_{i_g, j_1} < c_{i_h, j_2} - c_{i_h, j_1} \tag{8}$$

Furthermore, we are now in the situation $i_g < i_h < j_1 < j_2$ and from Corollary 13 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 (8) to a contradiction and we get $\min(j_1) \leqslant \min(j_2)$. $\qquad\square$

Corollary 15: *Using the Matrix-Searching Algorithm from Theorem 7 in the DP approach (see Algorithm 2), the k-link shortest path problem in the constructed digraph can be solved in $\mathcal{O}(nk)$ time.*

Hence, incorporating the time for the preprocessing (see Section 4) we can state the following result.

Corollary 16: *The two-dimensional HSSP can be solved in $\mathcal{O}(n^2)$ time.*

Remark 17:
Note that the algorithm induced from Corollary 16 can obtain the solution of the HSSP for all $k = 1, \ldots, n$ in $\mathcal{O}(n^2)$ time.

7 Conclusion

In this paper, 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 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 problem in $\mathcal{O}(n^2)$ improving the best known complexity bound of $\mathcal{O}(n^3)$. We remark that if a parallel environment is available, the k-link shortest path problem on our special digraph from Section 6 can be solved in $\mathcal{O}(n\sqrt{k}\log(n))$ [2].

It is an open question whether the time complexity of our dynamic programming approach can be reduced by skipping the preprocessing step, i.e. calculating the weights/costs on demand. In principle, an overall time complexity of $O(nk)$ could be achieved by computing the matrix entries during the run of the matrix searching algorithm in constant amount of time (see, e.g., in [6]). However, for our particular problem, it is not clear how to combine each step of the matrix searching algorithm with area computation in constant amount of time.

The developed methods cannot be used for more than two dimensions. The IP formulation (IP_k) from Section 5 can be extended to three dimensions for $N \subseteq \mathbb{R}^3$. However, the corresponding LP will 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 relaxation (w.r.t. $z^{ref} = 0$) has objective value 11.31 and the integer program 11.02. Moreover, also the whole graph construction from Section 6 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.

8 Acknowledgments

References

[1] 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.

[2] 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.

[3] Alok Aggarwal, Maria M. Klawe, Shlomo Moran, Peter Shor, and Robert Wilber. Geometric Applications of a Matrix-Searching Algorithm. *Algorithmica*, 2(1-4):195–208, 1987.

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

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

[6] J. Hershberger and S. Suri. Matrix Searching with the Shortest Path Metric. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of Computing*, pages 485–494. ACM press, 1993.

[7] 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.

[8] Christian Igel, Nikolaus Hansen, and Stefan Roth. Covariance Matrix Adaptation for Multi-objective Optimization. *Evol. Comput.*, 15(1):1–28, March 2007.

[9] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, New York, 1999.

[10] Aníbal Ponte, Luís Paquete, and José R. Figueira. On Beam Search for Multicriteria Combinatorial Optimization Problems. In Nicolas Beldiceanu, Narendra Jussien, and Éric Pinson, editors, *Integration of AI and OR Techniques in Contraint Programming for Combinatorial Optimization Problems*, volume 7298 of *Lecture Notes in Computer Science*, pages 307–321. Springer Berlin Heidelberg, 2012.

[11] Daniel Vaz, Luís Paquete, and Aníbal Ponte. A note on the ϵ-indicator subset selection. *Theoretical Computer Science*, 499:113 – 116, 2013.