A linear bound on the number of scalarizations needed to solve discrete tricriteria optimization problems

Kerstin Dächert

joint work with Kathrin Klamroth

Department of Mathematics and Natural Sciences
Universität Wuppertal, Germany

12/09/2014

Recent advances in multi-objective optimization, Vienna

Outline

1. Introduction
2. Generic decomposition of the 'search region'
3. Redundancy for problems with more than two criteria ($m > 2$)
4. Improved decomposition for $m = 3$
5. Linear bound on the number of subproblems for $m = 3$
6. Conclusion

Notation

General multicriteria problem:

$$\begin{aligned} \min \quad & f(x) = [f_1(x), \ldots, f_m(x)]^\top \\ \text{s.t.} \quad & x \in X \end{aligned}$$

Representation in outcome space:

$$\begin{aligned} \min \quad & z = [z_1, \ldots, z_m]^\top \\ \text{s.t.} \quad & z \in Z \end{aligned}$$

Z discrete, finite set

Goal: Determine the entire nondominated set Z_N

Approach based on scalarizations

- Solve sequence of scalarizations with different parameter choices (‘subproblems’)
- Basic algorithm

Determine initial search region; $s = 1$;
while there is some unexplored region **do**
    Choose region, solve subproblem ‘therein’
    if subproblem infeasible **then**
        remove explored region
    else
        save new nondominated point z^s;
        update search region based on z^s;
    end if
    $s := s + 1$;
end while
Output: Set of nondominated points Z_N

Bound on number of subproblems

Bicriteria case:

- $\mathcal{O}(N)$, at most $2N - 1$, see Aneja & Nair (1979), Chalmet et al. (1986), Ralphs et al. (2006)

Multicriteria case:

- $\mathcal{O}(N^{m-1})$, see Laumanns et al. (2006), Özlen and Azizoğlu (2009), Lokman and Köksalan (2013), Ozlen et al. (2014), Kirlik and Sayın (2014)
  $\Rightarrow \mathcal{O}(N^2)$ for $m = 3$

We show $\mathcal{O}(N)$ for $m = 3$:

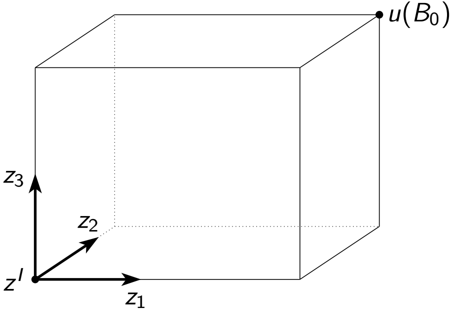
- $3N - 2$ for arbitrary scalarization
- $2N - 1$ for ε-constraint method

Decomposition of search region for $m \geq 2$

Initial search region (box)

$$B_0 := \{z \in \mathbb{R}^m : z^I \leqq z < u\}$$

with $u_i := \max_{x \in X}\{f_i(x)\} + \delta,\ i = 1, \ldots, m,\ \delta > 0$

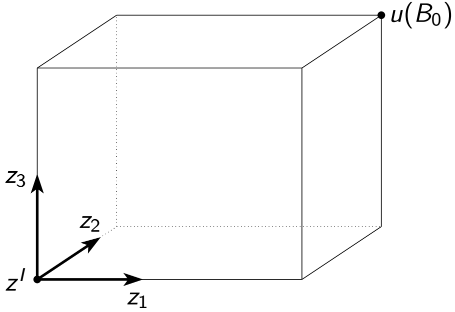


Decomposition of search region for $m \geq 2$

Initial search region (box)

$$B_0 := \{z \in \mathbb{R}^m \,:\, z^I \leqq z < u\}$$

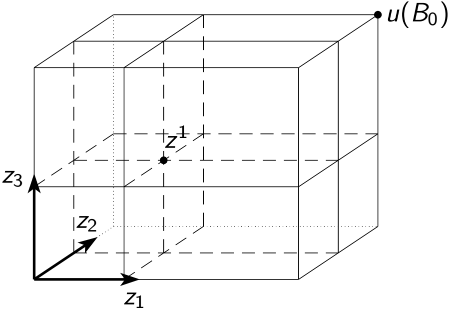
$\rightsquigarrow$ Note: Every box B characterized by $u(B)$



Decomposition of search region for $m \geq 2$

Solve subproblem in $B_0 \rightsquigarrow z^1 \in Z_N \cap B_0$

Insertion of z^1 into B_0

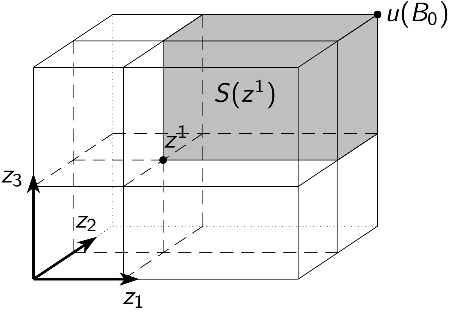


Decomposition of search region for $m \geq 2$

By definition of nondominance:

$$Z_N \cap S(z^1) = \{z^1\} \quad \text{mit} \quad S(z^1) := \{z \in B_0 : z \geqq z^1\}$$

$\Rightarrow$ All $z \in Z_N \setminus \{z^1\}$ contained in $B_0 \setminus S(z^1)$

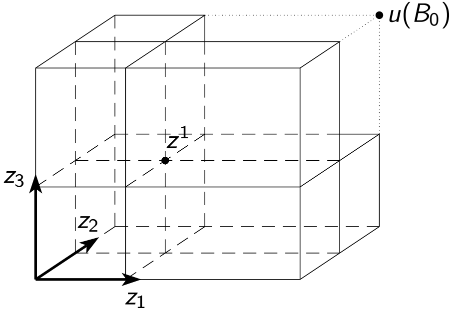


Decomposition of search region for $m \geq 2$

By definition of nondominance:

$$Z_N \cap S(z^1) = \{z^1\} \quad \text{mit} \quad S(z^1) := \{z \in B_0 : z \geqq z^1\}$$

$\Rightarrow$ All $z \in Z_N \setminus \{z^1\}$ contained in $B_0 \setminus S(z^1)$

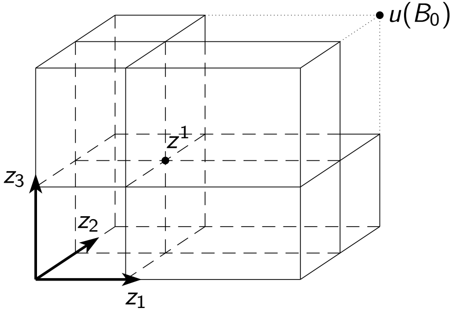


Decomposition of search region for $m \geq 2$

Representation of $B_0 \backslash S(z^1)$ by $\bigcup_{i=1}^{m} B_{1,i}$ with

$$B_{1,i} := \{z \in B_0 : z_i < z_i^1\}, \quad i = 1, \ldots, m,$$

i.e. $u_i(B_{1,i}) := z_i^1$, $u_j(B_{1,i}) := u_j(B_0) \ \forall j \neq i$

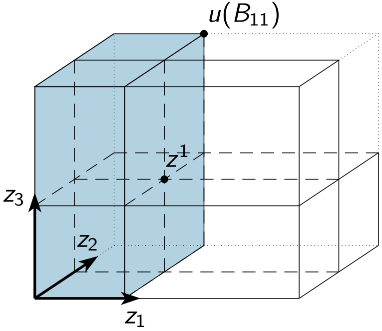


Decomposition of search region for $m \geq 2$

Representation of $B_0 \backslash S(z^1)$ by $\bigcup_{i=1}^m B_{1,i}$ with

$$B_{1,i} := \{z \in B_0 : z_i < z_i^1\}, \quad i = 1, \ldots, m,$$

i.e. $u_i(B_{1,i}) := z_i^1$, $u_j(B_{1,i}) := u_j(B_0) \;\; \forall\, j \neq i$

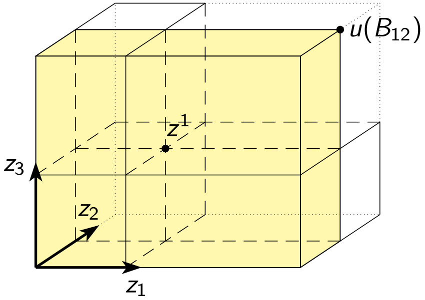


Decomposition of search region for $m \geq 2$

Representation of $B_0 \backslash S(z^1)$ by $\bigcup_{i=1}^{m} B_{1,i}$ with

$$B_{1,i} := \{z \in B_0 : z_i < z_i^1\}, \quad i = 1, \ldots, m,$$

i.e. $u_i(B_{1,i}) := z_i^1$, $u_j(B_{1,i}) := u_j(B_0)\ \ \forall j \neq i$

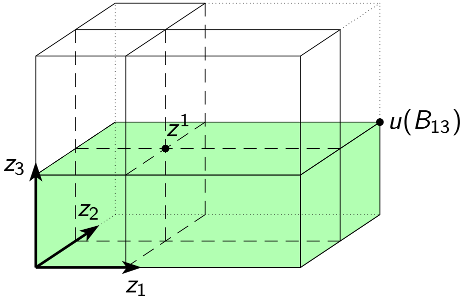


Decomposition of search region for $m \geq 2$

Representation of $B_0 \backslash S(z^1)$ by $\bigcup_{i=1}^{m} B_{1,i}$ with

$$B_{1,i} := \{z \in B_0 : z_i < z_i^1\}, \quad i = 1, \dots, m,$$

i.e. $u_i(B_{1,i}) := z_i^1$, $u_j(B_{1,i}) := u_j(B_0)\ \forall j \neq i$

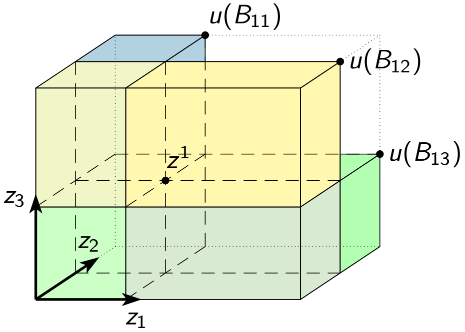


Decomposition of search region for $m \geq 2$

Representation of $B_0 \backslash S(z^1)$ by $\bigcup_{i=1}^{m} B_{1,i}$ with

$$B_{1,i} := \{z \in B_0 : z_i < z_i^1\}, \quad i = 1, \ldots, m,$$

$\Rightarrow$ Decomposition of $B_0 \backslash S(z^1)$ into m (non-disjoint) subboxes



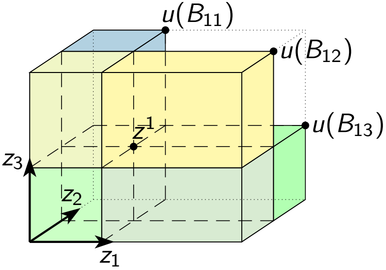
Full m-split

Full m-split

Replace box B wrt. $z^s \in B \cap Z_N$ into m new boxes

$$B_{s,i} := \{z \in B \,:\, z_i < z_i^s\} \quad \forall\, i = 1, \ldots, m$$

(see Dhaenens et al. (2010), Przybylski et al. (2010))



Note: No decomposition into $2^m - 1$ disjoint boxes (cf. Tenfelde-Podehl (2003)) in order to keep number of subproblems small

Full m-split

1. Using full m-split in iterative algorithm:
   In every iteration s, we split every box, in which current nondominated point z^s is contained, into m new boxes.

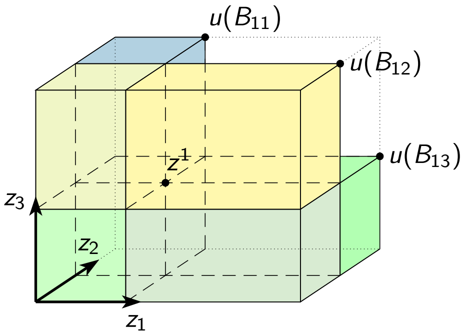
2. Correctness:
   In iteration s, only

   $$S_2(z^s) := \{z \in B_0 : z \geqq z^s\}$$

   is excluded from search region

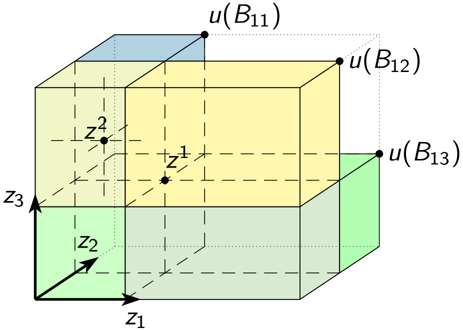
3. Disadvantage of full m-split: Redundant boxes appear for $m \geq 3$

Full m-split: Redundancy for $m \geq 3$



$$\mathcal{B}_2 = \{B_{11}, B_{12}, B_{13}\}$$

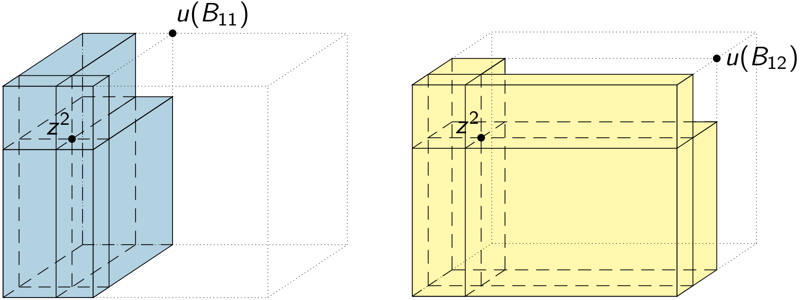
Full m-split: Redundancy for $m \geq 3$



Let $z^2 \in (B_{11} \cap B_{12})$

$\Rightarrow$ Split B_{11} and B_{12} into 3 new boxes, resp.

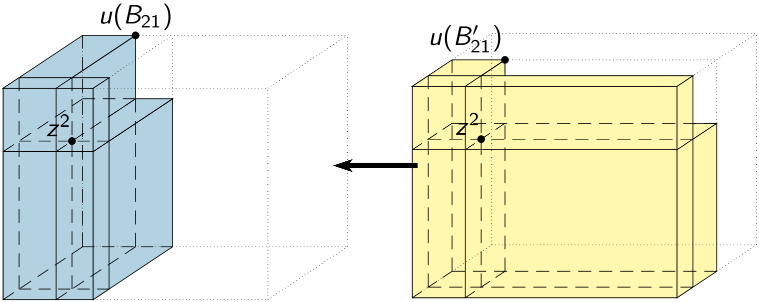
Full m-split: Redundancy for $m \geq 3$



B_{21}, B_{22}, B_{23} $B'_{21}, B'_{22}, B'_{23}$

Full 3-split of B_{11} and B_{12} wrt. z^2

Full m-split: Redundancy for $m \geq 3$

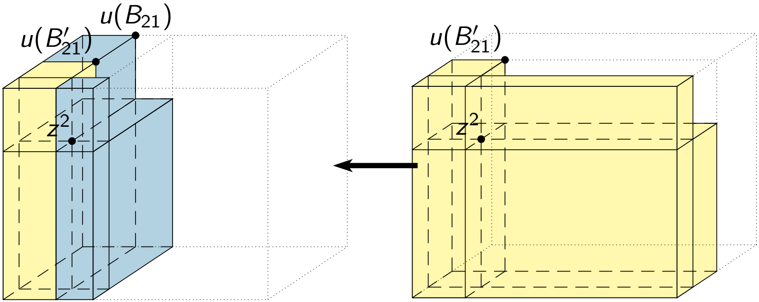


$\underline{\mathbf{B_{21}}}, B_{22}, B_{23}$ $\underline{\mathbf{B'_{21}}}, B'_{22}, B'_{23}$

Split wrt. $i = 1$:

$B'_{21} \subseteq B_{21} \iff u(B'_{21}) \leqq u(B_{21})$

Full m-split: Redundancy for $m \geq 3$

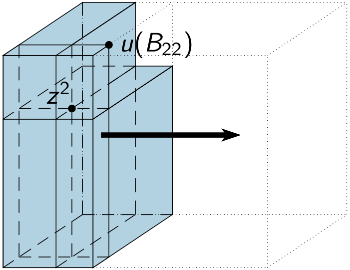


$\underline{\mathbf{B_{21}}}, B_{22}, B_{23}$ $\underline{\mathbf{B'_{21}}}, B'_{22}, B'_{23}$

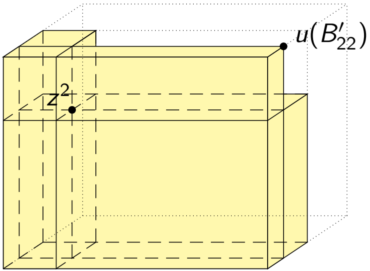
Split wrt. $i = 1$:

$$B'_{21} \subseteq B_{21} \iff u(B'_{21}) \leqq u(B_{21})$$

Full m-split: Redundancy for $m \geq 3$



B_{21}, $\underline{\mathbf{B_{22}}}$, B_{23}

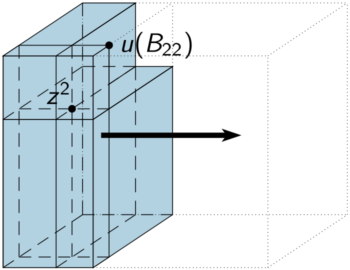


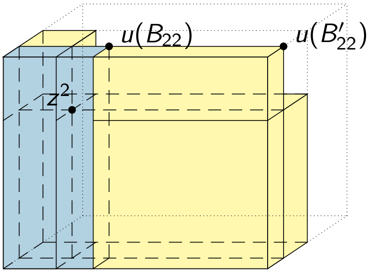
B'_{21}, $\underline{\mathbf{B'_{22}}}$, B'_{23}

Split wrt. $i = 2$:

$$B_{22} \subseteq B'_{22} \iff u(B_{22}) \leqq u(B'_{22})$$

Full m-split: Redundancy for $m \geq 3$



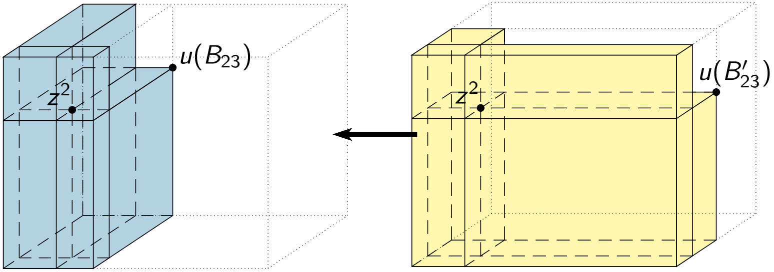


$B_{21}, \underline{\mathbf{B_{22}}}, B_{23}$ $B'_{21}, \underline{\mathbf{B'_{22}}}, B'_{23}$

Split wrt. $i = 2$:

$$B_{22} \subseteq B'_{22} \iff u(B_{22}) \leqq u(B'_{22})$$

Full m-split: Redundancy for $m \geq 3$

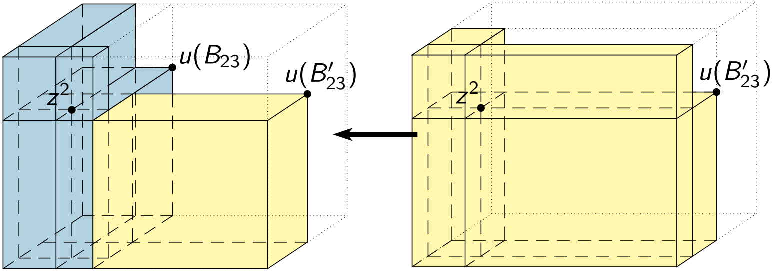


$B_{21}, B_{22}, \underline{\mathbf{B_{23}}}$ $B'_{21}, B'_{22}, \underline{\mathbf{B'_{23}}}$

Split wrt. $i = 3$:

no redundancy

Full m-split: Redundancy for $m \geq 3$



$B_{21}, B_{22}, \underline{\mathbf{B_{23}}}$

$B'_{21}, B'_{22}, \underline{\mathbf{B'_{23}}}$

Split wrt. $i = 3$:
no redundancy

Redundancy

Full m-split produces redundant boxes

- Example: already in 2nd iteration, two of the six new boxes redundant
- if redundant boxes are kept in decomposition
  - additional, unnecessary subproblems are solved
  - running time of algorithm increased

$\Rightarrow$ avoid generation of redundant boxes

Identifying redundant boxes:

1. compare upper bounds $u(B)$ pairwise, remove redundant ones (Przybylski et al. (2010))
2. detect redundant boxes before their creation, i.e. only split box if resulting box non-redundant

Individual subsets

Observation:

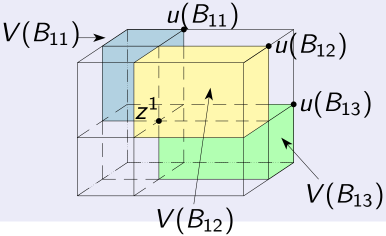
B non-redundant $\iff$ B contains non-empty subset which is not part of any other box of the decomposition

Definition (Individual subsets)

For every $\bar{B} \in \mathcal{B}_s$, the set

$$V(\bar{B}) := \bar{B} \setminus \left(\bigcup_{B \in \mathcal{B}_s \setminus \{\bar{B}\}} B \right)$$

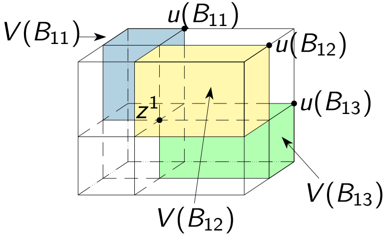
is called individual subset of $\bar{B}$.



Individual subsets

- Maintaining non-redundant boxes $\iff$ maintaining boxes with non-empty individual subset
- Split box B wrt. $i \iff V(B_i) \neq \emptyset$
- Goal: Explicit representation of $V(B)$

Simplifying assumption:
General position: For all $z, \bar{z} \in Z_N, z \neq \bar{z}$, let $z_i \neq \bar{z}_i$ for all $i = 1, \ldots, m$.



Representation of $V(B)$

- Consider box $\bar{B}$ of (non-redundant) decomposition $\mathcal{B}_s$ and component $i \in \{1, \dots, m\}$.
- A box $\hat{B} \in \mathcal{B}_s$ with

$$u_i(\hat{B}) < u_i(\bar{B})$$
$$u_j(\hat{B}) \geq u_j(\bar{B}) \quad \forall j \neq i$$

and

$$u_i(\hat{B}) = \max\{u_i(B) : B \in \mathcal{B}_s, u_i(B) < u_i(\bar{B})\}$$

limits $V(\bar{B})$ wrt. i.

- We call $\hat{B}$ neighbor of $\bar{B}$ wrt. component i in iteration s (Notation: $\hat{B} = B_i^s(\bar{B})$).

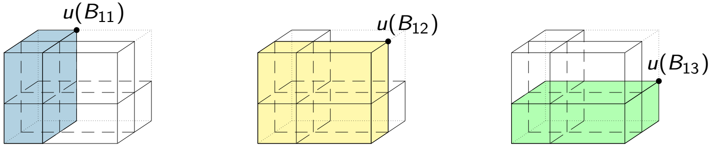
Lemma (Neighbor of a box for $m = 3$)

For every $\bar{B} \in \mathcal{B}_s$ *and every* $i \in \{1, 2, 3\}$ *with* $u_i(\bar{B}) > \min_{B \in \mathcal{B}_s}\{u_i(B)\}$ $\exists!\, \hat{B} \in \mathcal{B}_s$ *such that*

$$u_i(\hat{B}) < u_i(\bar{B})$$
$$u_j(\hat{B}) > u_j(\bar{B}) \quad \text{for some } j \neq i$$
$$u_k(\hat{B}) = u_k(\bar{B}) \quad \text{for } k \neq i, j$$

and

$$u_i(\hat{B}) = \max\{u_i(B) : B \in \mathcal{B}_s, u_i(B) < u_i(\bar{B})\}$$



Representation of $V(B)$ for $m = 3$

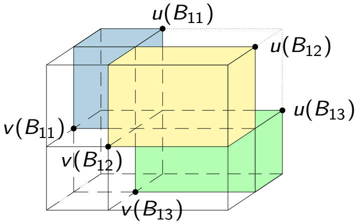
The individual subsets $V(B), B \in \mathcal{B}_s$ can be represented as

$$V(B) = \{z \in B_0 : v(B) \leqq z < u(B)\}$$

with

$$v_i(B) := \begin{cases} u_i(B_i^s(B)), & \text{if } B_i^s(B) \neq \emptyset \\ z_i^I, & \text{otherwise} \end{cases}, \quad i \in \{1, 2, 3\}$$

and $B_i^s(B)$ denoting the neighbor of B wrt. i in iteration s,



e.g.

$$v(B_{12}) = \begin{pmatrix} u_1(B_{11}) \\ z_2^I \\ u_3(B_{13}) \end{pmatrix}$$

The v-split

$$\boxed{\text{Recall: Split box } B \text{ wrt } i \iff V(B_i) \neq \emptyset}$$

Lemma

Let $z^s \in B$, and let B_i be box obtained from B by a split wrt. component $i \in \{1, 2, 3\}$.

$$\textit{Then } B_i \textit{ is non-redundant} \iff z_i^s \geq v_i(B).$$

Definition (v-split)

Let $z^s \in B$. We call split of $B \in \mathcal{B}_s$ wrt. components $i \in \{1, 2, 3\}$, for which

$$z_i^s \geq v_i(B)$$

holds, v-split of B.

The v-split algorithm

Changes in comparison to generic algorithm:

1. Save $v(B)$
2. Apply v-split: For every $B \in \mathcal{B}_s$ with $z^s \in B$ and every $i \in \{1, 2, 3\}$ check whether
$$z_i^s \geq v_i(B)$$
3. Update of $v(B)$:

   For all new $B_i, i \in \{1, 2, 3\}$:
   - sort $u(B_i)'s$ increasingly wrt. $j \neq i$, decreasingly wrt. $k \neq i, j$:
   $$z_j^s < \dots \leq u_j(\hat{B}_i) \leq u_j(\bar{B}_i) \leq u_j(\tilde{B}_i) \leq \dots$$
   $$\dots \geq u_k(\hat{B}_i) \geq u_k(\bar{B}_i) \geq u_k(\tilde{B}_i) \geq \dots > z_k^s$$
   - Set $v_j(\bar{B}_i) := u_j(\hat{B}_i)$, $v_k(\bar{B}_i) := v_k(\tilde{B}_i)$ and so on

The v-split algorithm

Changes in comparison to generic algorithm:

1. Save $v(B)$
2. Apply v-split: For every $B \in \mathcal{B}_s$ with $z^s \in B$ and every $i \in \{1, 2, 3\}$ check whether
$$z_i^s \geq v_i(B)$$
3. Update of $v(B)$:

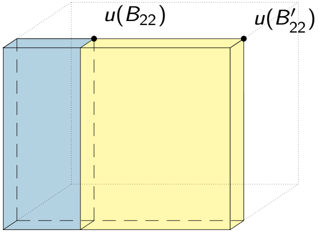
   For all new $B_i, i \in \{1, 2, 3\}$:

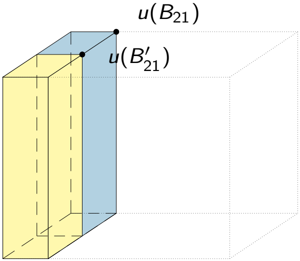
   - sort $u(B_i)'s$ increasingly wrt. $j \neq i$, decreasingly wrt. $k \neq i, j$:
$$z_j^s < \dots \leq u_j(\hat{B}_i) \leq u_j(\bar{B}_i) \leq u_j(\tilde{B}_i) \leq \dots$$
$$\dots \geq u_k(\hat{B}_i) \geq u_k(\bar{B}_i) \geq u_k(\tilde{B}_i) \geq \dots > z_k^s$$
   - Set $v_j(\bar{B}_i) := u_j(\hat{B}_i)$, $v_k(\bar{B}_i) := v_k(\tilde{B}_i)$ and so on

Example revisited

Full m-split: Two redundant boxes

- B_{22} (Split of B_{11} wrt. $i = 2$)
- B'_{21} (Split of B_{12} wrt. $i = 1$)





Example revisited

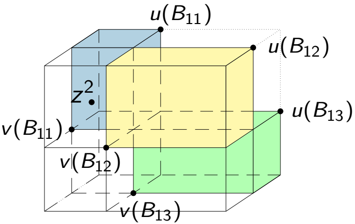
Full *m*-split: Two redundant boxes

- B_{22} (Split of B_{11} wrt. $i = 2$)
- B'_{21} (Split of B_{12} wrt. $i = 1$)

v-Split in B_{11}:

$$\begin{array}{lll} z_1^2 & > & v_1(B_{11}) \quad \checkmark \\ z_2^2 & < & v_2(B_{11}) \\ z_3^2 & > & v_3(B_{11}) \quad \checkmark \end{array}$$

$\Rightarrow$ Split B_{11} wrt. $i = 1$ and $i = 3$
(Redundant box B_{22} not generated!)



Example revisited

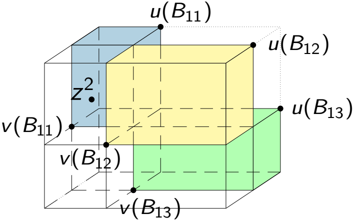
Full m-split: Two redundant boxes

- B_{22} (Split of B_{11} wrt. $i = 2$)
- B'_{21} (Split of B_{12} wrt. $i = 1$)

v-Split in B_{12}:

$$
\begin{array}{lll l}
z_1^2 & < & v_1(B_{12}) & \\
z_2^2 & > & v_2(B_{12}) & \checkmark \\
z_3^2 & > & v_3(B_{12}) & \checkmark
\end{array}
$$

$\Rightarrow$ Split B_{12} wrt. $i = 2$ and $i = 3$
(Redundant box B'_{21} not generated!)



Linear bound on the number of subproblems

Lemma

In every iteration $s \geq 1$ of the v-split algorithm, in which a new nondominated point z^s is found, the number of boxes in the decomposition increases by at most two.

Proof.

Case 1: one box is split $\Rightarrow$ 3 boxes replace one $(-1 + 3 = 2 \checkmark)$

Case 2: more than one box split:

- every box split wrt. at most 2 components
- no pair of boxes split wrt. to the same 2 components
  $\Rightarrow$ at most $2 \cdot 3 - 3 = 3$ additional boxes
- if 3 boxes split wrt. 2 components
  $\Rightarrow$ exists box which is split wrt. no component $(3 - 1 = 2 \checkmark)$

$\square$

Theorem

For Z_N finite ($N = |Z_N|$) and given appropriate initial search region with $lb = z^I$, the v-split algorithm requires at most $3N - 2$ subproblems in order to generate the entire nondominated set.

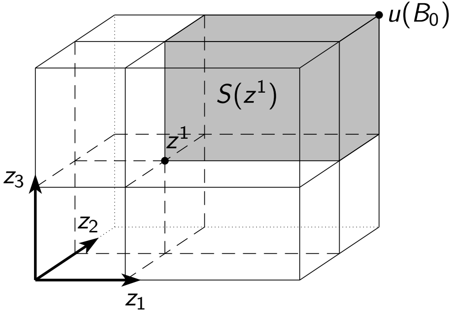
Proof.

- in every iteration one subproblem solved
  $\Rightarrow$ number of subproblems equals number of iterations
- for every nondominated point generated
  $\Rightarrow$ number of boxes increases by at most two (previous Lemma)
- every nondominated point is generated exactly once,
  every empty box is investigated exactly once
  $\Rightarrow$ at most $3N$ boxes explored
- plus initial box $\Rightarrow 3N + 1$
- if $z_i^s = z_i^I$ for $i \in \{1, 2, 3\}$, no box created wrt. $i \Rightarrow 3N - 2$

$\square$

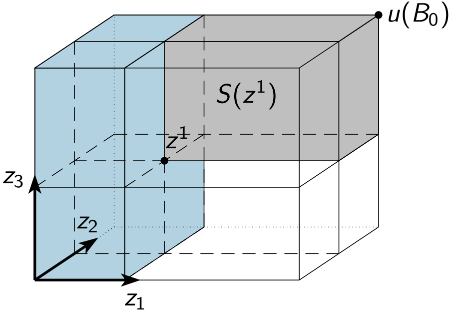
Remarks

1. v-split algorithm and linear bound also valid if nondominated points not in general position
   $\rightsquigarrow$ degenerated individual subsets
2. v-split algorithm independent of particular scalarization
   $\rightsquigarrow$ improved bound $2N - 1$ for ε-constraint method



Remarks

1. v-split algorithm and linear bound also valid if nondominated points not in general position
   $\rightsquigarrow$ degenerated individual subsets
2. v-split algorithm independent of particular scalarization
   $\rightsquigarrow$ improved bound $2N - 1$ for ε-constraint method



Implementation

Matlab-Implementation of the v-split-algorithm with 3D-visualization

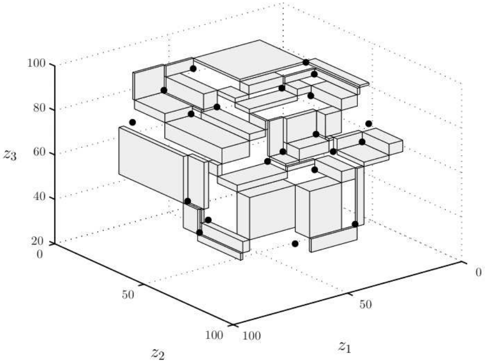


Figure: Example with 21 nondominated points
(Illustration of the individual subsets of all boxes at the end of the algorithm)

Computational experiments

- Multidimensional, tricriteria knapsack problem
- Test problem from Laumanns et al. (2006)
- Original data
- Scalarizations: Weighted Tchebycheff (WT) und ε-Constraint (EC), both in variants Two-stage (TS) und Augmented (A)
- IBM ILOG CPLEX Optimization Studio Version 12.5
- CPLEX called without parallelizing
- MATLAB R2013a
- 4x Intel Xeon E7540 CPUs (2.0 GHz), 128 GB memory

Computational experiments (1)

Validation of theoretical upper bounds $3N - 2$ (WT) and $2N - 1$ (EC)

n	N		WT CPU	WT #SP	EC CPU	EC #SP
10	9	TS	10.03	25	7.97	17
		A	7.81		6.09	
20	61	TS	56.42	181	43.29	121
		A	42.72		30.02	
30	195	TS	213.31	583	163.15	389
		A	163.29		114.39	
40	389	TS	464.47	1165	361.74	777
		A	361.01		257.64	
50	1048	TS	1552.56	3142	1369.89	2095
		A	1174.90		1012.15	

$\Rightarrow$ Bounds met precisely in all instances

Computational experiments (2)

Comparison with algorithms of

1. Lokman & Köksalan (2013)
2. Kirlik & Sayın (2014)
3. Özlen, Burton & MacRae (2014)

Recall:

- $\mathcal{O}(N^2)$ subproblems for $m = 3$
- ε-Constraint Scalarization

$\Rightarrow$ Use (EC) for v-split-algorithm

- Note: All algorithms reimplemented in Matlab and tested with both variants (TS) and (A)

Computational experiments (2)

1. Comparison with Lokman & Köksalan (2013):

n	N		v-Split EC CPU	v-Split EC #SP	LK CPU	LK #SP
10	9	TS	7.97	17	9.48	20
		A	6.09		6.67	
20	61	TS	43.29	121	53.04	127
		A	30.02		31.76	128
30	195	TS	163.15	389	267.88	468
		A	114.39		159.12	464
40	389	TS	361.74	777	657.58	852
		A	257.64		445.07	
50	1048	TS	1369.89	2095	4772.89	2193
		A	1012.15		4129.47	2200

Observation: v-Split-Algorithm requires

- less subproblems,
- less CPU time than LK

Computational experiments (2)

2. Comparison with Kirlik & Sayın (2014):

n	N		v-Split EC CPU	v-Split EC #SP	KS CPU	KS #SP
10	9	TS	7.97	17	8.50	17
		A	6.09		6.07	
20	61	TS	43.29	121	50.08	115
		A	30.02		30.26	
30	195	TS	163.15	389	242.42	373
		A	114.39		155.89	372
40	389	TS	361.74	777	701.95	739
		A	257.64		516.15	738
50	1048	TS	1369.89	2095	4174.48	1913
		A	1012.15		3603.67	1914

Observation: v-split-algorithm requires

- more subproblems in general,
- but less CPU time than KS

Computational experiments (2)

3. Comparison with Özlen, Burton & MacRae (2014):

n	N		v-Split EC CPU	v-Split EC #SP	OBM CPU	OBM #SP
10	9	TS	7.97	17	8.85	19
		A	6.09		6.46	
20	61	TS	43.29	121	48.50	117
		A	30.02		28.83	
30	195	TS	163.15	389	197.05	375
		A	114.39		110.33	374
40	389	TS	361.74	777	430.84	741
		A	257.64		246.68	740
50	1048	TS	1369.89	2095	1533.93	1915
		A	1012.15		945.35	1916

Observation: v-split-algorithm requires

- more subproblems in general,
- partially less, partially more CPU time than OBM

Summary

Conclusion:

1. new split for tricriteria problems avoids redundant boxes
2. linear bound on number of subproblems:
   - $3N - 2$ for arbitrary scalarization
   - $2N - 1$ for ε-constraint method
3. competes with state-of-the-art algorithms

Ongoing research:

1. explicit use of neighborhood structure in algorithm
2. generalization to $m \geq 4$
3. representative subsets / application to continuous problems

Thank you! Questions?