The multi-objective Minkowski Sum Problem - Theory and definitions

Mark Lyngesen[a], Sune Lauth Gadegaard[a], Lars Relund Nielsen[a]

[a]*Department of Economics and Business Economics, BSS, Aarhus University, Fuglsangs Allé 4, Aarhus, 8000, Aarhus Municipality, Denmark*

Abstract

Some multi-objective optimization problems can be solved using decomposition methods where a set of smaller independent subproblems are solved. For this class of problems, the global objectives can be described as the sum of local objective values of the subproblems. We present some theoretical results for the nondominated sets of such problems and formulate so-called generator sets. The generator sets consists only of the necessarily nondominated points for each subproblem required for generating the global nondominated set. Using the generator sets we define improved upper bound sets that reduce the search area for nondominated (generator) points in the subproblems. Using these generator upper bound sets we present a method of solving the otherwise independent subproblems in a way which solve the global problem by sharing information between subproblems.

Report submitted to BSS, Aarhus University *7th February 2024*

Contents

1 Introduction **3**

2 Literature review **3**

3 Prerequisites **3**

3.1 General MO 3

4 Minkowski Sum Problem **3**

5 Generator sets and generator upper bound sets **3**

5.1 Theory for two subproblems S = 1,2 3

5.2 Generator sets 3

5.3 Generator upper bound sets 3

6 Empirical study **4**

6.1 Test instances 4

6.2 Empirical study of Generator sets 6

6.3 Empirical study of Generator upper bound sets 6

6.4 Empirical study of information-sharing between subproblems . 6

6.5 Methods for answering research questions 6

6.5.1 Sequential minimum generators 9

1. Introduction

2. Literature review

[Jeg vil foreslå at vi lægger dette i introduktionen. Se Dietz et al. (2020) for eksempel. Start bredt.]

Review of relevant literature

1. Complex systems in generel
2. Uncoupled systems
3. Minkowski sum problems.
4. Kerbérénès phd.
5. filtering problems

3. Prerequisites

3.1. General MO

- Relations (sets and vectors)
- Bound sets.
- Operator notation: $\oplus, \ominus, \bigoplus ... \times$
- Notation $\mathcal{X}, \mathcal{Y}, f, \mathcal{X}_E, \mathcal{Y}_\mathcal{N}$...

4. Minkowski Sum Problem

MSP definition for general p and S

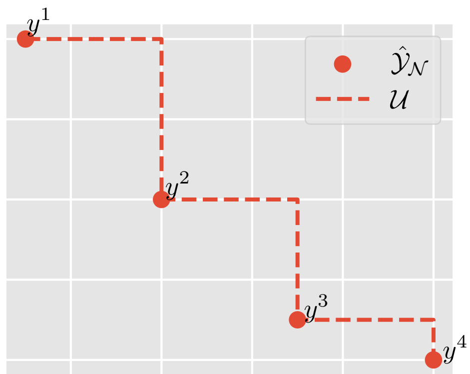
5. Generator sets and generator upper bound sets

5.1. Theory for two subproblems S = 1,2

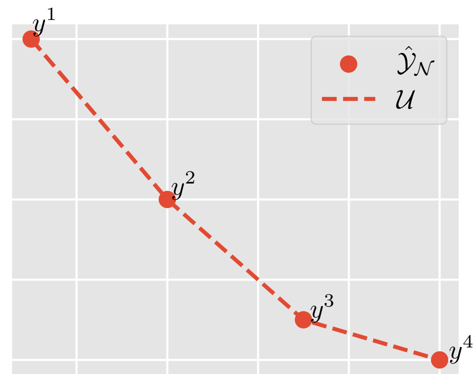
5.2. Generator sets

5.3. Generator upper bound sets

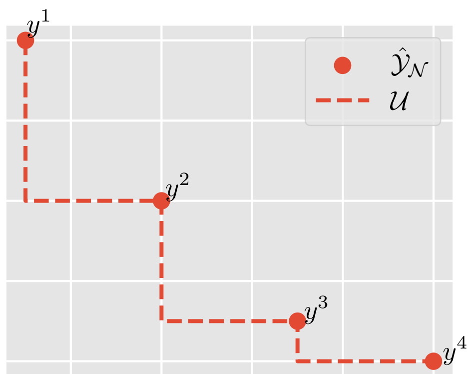
Consider different generator upper bound sets. $\bar{\mathcal{U}}^1$ is generally defined by some known solution $\hat{\mathcal{Y}}^1_\mathcal{N} \subseteq \mathcal{Y}^1_\mathcal{N}$ and $\hat{\mathcal{Y}}^1_\mathcal{N} \subseteq \mathcal{Y}^2_\mathcal{N}$. Consider how $\bar{\mathcal{U}}^1$ differs depending on the assumptions on the set of incumbent solutions, see Subsection 5.3.



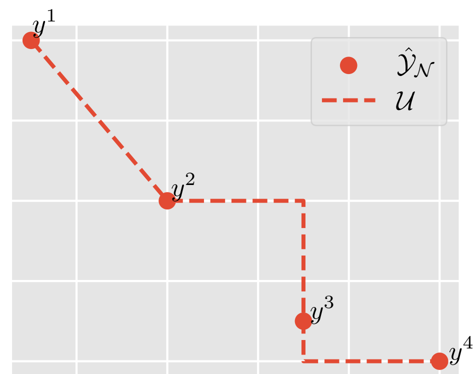
(a) Consecutive



(b) Supported



(c) Not consecutive



(d) mixed

Figure 1: Induced upper bound sets based on varrying assumptions on the set $\hat{\mathcal{Y}}_{\mathcal{N}} \subseteq \mathcal{Y}_{\mathcal{N}}$.

6. Empirical study

The purpose of the computational study is to answer the following questions:

[General intro see e.g. https://www.research.relund.dk/publications/pdf/forget22b.pdf and https://www.research.relund.dk/publications/pdf/forget22a.pdf]

1. How large is $|\mathcal{Y}_{\mathcal{N}}|$ given $|\mathcal{Y}^s_{\mathcal{N}}|, s \in S$ ($\prod_s |\mathcal{Y}^s_{\mathcal{N}}|$)?
2. What is the relative size of the generator sets compared to $|\mathcal{Y}^s_{\mathcal{N}}|, s \in S$?
3. Does the sequence of subproblems impact the generator set size?

6.1. Test instances

Possible setup: In an instance folder, create a subfolder for each instance = problem. Keep on csv file for each subproblem. Even better could be to have no subfolders but instead one json file for each instance. Another solution is to have a json file for each subproblem. This is properly the best option. However, the instances will be correlated. An instance can then be defined as a combination of subproblems.

For each subproblem we

- Test $p = 2, \dots, 5$. [4 options]
- Width is the same for all objectives (Kerberenes (2022) use $w_i = 1000$ in Chap. 2) or width is the same for approx. half of the objectives (e.g. 1000) and 1/4 for the remaining (i.e. 250). [2 options]
- Generate on a sphere: either lower/west part (many supported) or upper/east part (many unsupported). [2 options]
- Generate $10, 50, 100, 150, 200$ points and find the nd points among them for each subproblem. [5 options]
- Classify each nd point (supported, extreme, unsupported)
- Keep statistics: number of points, supported, extreme, unsupported, ratio = supported/unsupported, (range) width w_i = max-min, min and max value for each objective $i = 1, \dots p$.
- 5 instances for each config. [5 options]

In total $4 \cdot 2 \cdot 2 \cdot 2 \cdot 5 \cdot 5 = 800$ files. Naming convention could be `sub-<id>-<p>-<width>-<sphere-method>-<nd-points>-<ratio>-<instance-id>.json`.

Note that given integer points, an upper bound on the width of the master problem with $s = 1, \dots, m$ subproblems is $w_i = \sum_s w_{si}$. That is, if we keep the width of the subproblem fixed, the width of the master grows as m grows.

Note that given integer points and $p = 2$, an upper bound on the number of nd points is $1 + \min_i w_i$. For a problem with m subproblems an upper bound on the nd points is $1 + \min_i \sum_p w_{pi}$

The following instance/problem groups are generated given combinations of (this should maybe be changed for each question):

- $p = 2, \dots, 5$. [4 options]
- $m = 2, \dots 5$ where m is the number of subproblems. [4 options]

- All subproblems have the same sphere config, half/half. [3 options]
- Increasing nd points in subproblems (if m=3 choose 100, 200, 250), all have highest (n=250). [2 options]
- 10 instances for each config. [10 options]

In total $4 \cdot 4 \cdot 3 \cdot 2 \cdot 10 = 960$ files. Naming convention could be `prob-<id>-<p>-<m>-<sphere-method>-<sub-ids>.json`.
A result file in the format MCDMSociety may be created.

6.2. Empirical study of Generator sets

When are generator sets small relative to the non-dominated sets?

6.3. Empirical study of Generator upper bound sets

When are upper bound sets from generators 'good'. (make precise)?

6.4. Empirical study of information-sharing between subproblems

***RQ? Formulate major research questions.

- Generate a new point from a subproblem and update all generator sets and upper bound sets.
- Periodically generate a point for each subproblem where after all generator sets and upper bound sets are updated.
- Sweep: Generate all supported non-dominated points for each subproblem where after all generator sets and upper bound sets are updated.

6.5. Methods for answering research questions

In the following, we describe ways of generating test metrics used to answer the main empirical research questions. Each test is performed on an MO-MSP instance given by a sequence of sets $\{\mathcal{Y}^s\}_{s\in\mathcal{S}}$ for some ordered sequence of subproblems indexed by a set $\mathcal{S}$ (where we assume $\mathcal{Y}^s \subseteq \mathcal{Y}^s_{\mathcal{N}} \forall s \in \mathcal{S}$).

1. How large is $|\mathcal{Y}_\mathcal{N}|$ given $\{|\mathcal{Y}^s_\mathcal{N}|\}_{s\in\mathcal{S}}$? To determine the set $\mathcal{Y}_\mathcal{N}$ we run the Minkowski Sum filtering algorithm *Sequential Pooling algorithm with IF* from (Kerbérénès et al., 2022, Algorithm 1). This algorithm sequentially generates the nondominated Minkowski Sum and uses a nondominance filter as a subroutine. The speed of the nondominance algorithm differs for $p = 2$ and for $p > 2$.
   In each test we record the following measures:
   - $|\mathcal{Y}_\mathcal{N}|$ total nondominated points.
   - $|\mathcal{Y}_{\mathcal{SN}}|$ total supported nondominated points.
   - $|\mathcal{Y}_{\mathcal{EN}}|$ total extreme (supported) nondominated points.
   - $|\mathcal{Y}_\mathcal{N} \setminus \mathcal{Y}_{\mathcal{SN}}|$ total unsupported nondominated points.
2. What is the relative size of the generator sets compared to $|\mathcal{Y}^s|(\forall s \in \mathcal{S})$? We first note that the generator sets are not unique. We consider several different generator sets and compare them. We define a *Minimum generator set sequence* as a sequence of generators $\left\{\bar{\mathcal{Y}}^s\right\}_{s\in\mathcal{S}}$ where $\bar{\mathcal{Y}}^s \subseteq \mathcal{Y}^s$ such that $\bar{M} := \sum_{s\in\mathcal{S}}|\bar{\mathcal{Y}}^s|$ is minimum.
   Let $\bar{M}$ be the size of a minimum generator set sequence. $\bar{M}$ is defined by the following combinatorial optimization problem.

$$\bar{M} = \min \sum_{s\in\mathcal{S}} |\hat{\mathcal{Y}}^s| \tag{1}$$

$$s.t. \quad \bigoplus_{s\in\mathcal{S}} |\hat{\mathcal{Y}}^s| = \mathcal{Y}_\mathcal{N}, \tag{2}$$

$$\hat{\mathcal{Y}}^s \subseteq \mathcal{Y}^s, \quad \forall s \in \mathcal{S} \tag{3}$$

   We will reformulate the combinatorial problem as a general MIP. To do this we introduce the following notation. We order the sets $\mathcal{Y}^s := \{y^s_1, \ldots\} = \{\mathcal{Y}^s_i\}_{s\in\mathcal{I}^s}$ indexed by set $\mathcal{I}^s$ for all subproblems $s \in \mathcal{S}$. Likewise, we order the set $\mathcal{Y}_\mathcal{N} := \{y_j\}_{j\in\mathcal{J}}$ indexed by the set $\mathcal{J}$. We introduce decision variables x and w such that:

$$x^s_i = \begin{cases} 1, & \text{if } y^s_i \in \hat{\mathcal{Y}}^s \\ 0, & \text{otherwise} \end{cases} \qquad \forall i \in \mathcal{I}^s, \forall s \in \mathcal{S} \tag{4}$$

$$w^s_{i,j} = \begin{cases} 1, & \text{if } y^s_i \text{ generates } y_j \in \mathcal{Y}_\mathcal{N} \\ 0, & \text{otherwise} \end{cases} \qquad \forall i \in \mathcal{I}^s, \forall s \in \mathcal{S}, \forall j \in \mathcal{J}$$

To enforce the definitions of the decision variables we add the following constraints.

$$w_{i,j}^s \leq x_i^s, \quad \forall i \in \mathcal{I}^s, \forall s \in \mathcal{S}, \forall j \in \mathcal{J} \tag{5}$$

That is, y_i^s can only be part of generating y_j (for any j) if $y^s \in \hat{\mathcal{Y}}^s$. We now add constraints which ensure that each point in the set $\mathcal{Y}_\mathcal{N}$ is 'generated' by some combination of points of the subproblems:

$$\sum_{s\in\mathcal{S}} \sum_{i\in\mathcal{I}^s} w_{i,j}^s y_i^s = y_j, \quad \forall j \in \mathcal{J} \tag{6}$$

Finally, each point $y \in \mathcal{Y}_\mathcal{N}$ must be a combination of points with exactly one point from each subproblem:

$$\sum_{i\in\mathcal{I}^s} w_{i,j}^s = 1, \quad \forall s \in \mathcal{S}, \forall j \in \mathcal{J} \tag{7}$$

The constraints Equation (5) together with Equation (7) also ensures that at least one solution is chosen to each subproblem $\left(\sum_{i\in I^s} x_i^s, \forall s \in \mathcal{S}\right)$. The objective function is the sum of the chosen generator points: $\bar{M} := \sum_{s\in\mathcal{S}} \sum_{i\in\mathcal{I}^s} x_i^s$. This results in the following MIP:

$$\begin{aligned}
\bar{M} := \min & \sum_{s\in\mathcal{S}} \sum_{i\in\mathcal{I}^s} x_i^s \\
s.t. \quad & w_{i,j}^s \leq x_i^s & \forall i \in \mathcal{I}^s, \forall s \in \mathcal{S}, \forall j \in \mathcal{J} \\
& \sum_{s\in\mathcal{S}} \sum_{i\in\mathcal{I}^s} w_{i,j}^s y_i^s = y_j & \forall j \in \mathcal{J} \\
& \sum_{i\in\mathcal{I}^s} w_{i,j}^s = 1 & \forall s \in \mathcal{S}, \forall j \in \mathcal{J} \\
& x_i^s \in \{0,1\} & \forall i \in \mathcal{I}^s, \forall s \in \mathcal{S} \\
& w_{i,j}^s \in \{0,1\} & \forall i \in \mathcal{I}^s, \forall s \in \mathcal{S}, \forall j \in \mathcal{J}
\end{aligned}$$

Given an optimal solution $(\bar{x}, \bar{w})$ the corresponding generator sets $\bar{\mathcal{Y}}^s$ is gives as $\bar{\mathcal{Y}}^s = \{y_i^s \in \mathcal{Y}^s : \bar{x}_i^s = 1\}$.
For each test we record:

- Total points, supported points, extreme (supported) points, unsupported nondominated points.

- Global $|\mathcal{Y}_{\mathcal{N}}|$, $|\mathcal{Y}_{\mathcal{SN}}|$, $|\mathcal{Y}_{\mathcal{EN}}|$, $|\mathcal{Y}_{\mathcal{N}} \setminus \mathcal{Y}_{\mathcal{SN}}|$.
- Subproblems (already known) $|\mathcal{Y}^s_{\mathcal{N}}|$, $|\mathcal{Y}^s_{\mathcal{SN}}|$, $|\mathcal{Y}^s_{\mathcal{EN}}|$, $|\mathcal{Y}^s_{\mathcal{N}} \setminus \mathcal{Y}^s_{\mathcal{SN}}|$ $\forall s \in \mathcal{S}$.
- Generator sets $|\bar{\mathcal{Y}}^s|$, $|\bar{\mathcal{Y}}^s \cap \mathcal{Y}^s_{\mathcal{SN}}|$, $|\bar{\mathcal{Y}}^s \cap \mathcal{Y}^s_{\mathcal{EN}}|$, $|\bar{\mathcal{Y}}^s \setminus \mathcal{Y}^s_{\mathcal{SN}}|$ $\forall s \in \mathcal{S}$
- $\bar{M}$

6.5.1. Sequential minimum generators

In order to get a second measure for answering questions 2 and also to answer question 3 we seek to define minimum generator set sequence which depend on the index sets $\mathcal{S}$.

For each type of we are interested in generating sets $\bar{\mathcal{Y}}^s$ where we to some extend minimuze $\left(\left|\bar{\mathcal{Y}}^1\right|, \ldots \left|\bar{\mathcal{Y}}^{|\mathcal{S}|}\right|\right)$. For each test we record:

- $\left|\bar{\mathcal{Y}}^s\right|$ for all $s \in \mathcal{S}$
- $\sum_{s\in\mathcal{S}} \left|\bar{\mathcal{Y}}^s\right|$
- Also similar metrics for supported points, extreme supported points and unsupported.

MGS1

- Each generator $\bar{\mathcal{Y}}^{s'}$ depends on: $\bar{\mathcal{Y}}^1, \ldots, \bar{\mathcal{Y}}^{s'-1}, \mathcal{Y}^{s'}$.
- At each step a generator $\bar{\mathcal{Y}}^{s'} := \hat{\mathcal{Y}}^{s'}$ is fixed.
- At each step the combined generators must 'generate' $\left(\mathcal{Y}^1 \oplus \ldots \oplus \mathcal{Y}^{s'}\right)_{\mathcal{N}}$.

We will now define a more restricted sequence of generator sets which we call the *local sequentially minimum sequence of generator sets* (MGS1). The first generator set is simply defined as the set of nondominated points of the first subproblem.

$$\bar{\mathcal{Y}}^1 = \mathcal{Y}^1_{\mathcal{N}} \tag{8}$$

For the second generator we find the smallest set which together with the generator $\bar{\mathcal{Y}}^1$ generates $(\mathcal{Y}^1 \oplus \mathcal{Y}^2)_{\mathcal{N}}$.

$$\begin{aligned} \bar{\mathcal{Y}}^2 = \arg\min&|\hat{\mathcal{Y}}^2| && (9)\\ st. \quad &\left(\bar{\mathcal{Y}}^1 \oplus \hat{\mathcal{Y}}^2\right)_{\mathcal{N}} = \left(\mathcal{Y}^1 \oplus \mathcal{Y}^2\right)_{\mathcal{N}} \\ &\hat{\mathcal{Y}}^2 \subseteq \mathcal{Y}^2_{\mathcal{N}} && (10) \end{aligned}$$

For a general $s' \in \mathcal{S}$

$$\bar{\mathcal{Y}}^{s'} = \arg\min |\hat{\mathcal{Y}}^{s'}| \tag{11}$$

$$st. \quad \left(\left(\bigoplus_{s=1}^{s'-1} \bar{\mathcal{Y}}^{s} \right) + \hat{\mathcal{Y}}^{2} \right)_{\mathcal{N}} = \left(\bigoplus_{s=1}^{s'} \mathcal{Y}^{s} \right)_{\mathcal{N}}$$

$$\hat{\mathcal{Y}}^{s'} \subseteq \mathcal{Y}^{s'}_{\mathcal{N}} \tag{12}$$

MGS2

- Each generator $\bar{\mathcal{Y}}^{s'}$ depends on: $\bar{\mathcal{Y}}^{1}, \ldots, \bar{\mathcal{Y}}^{s'-1}, \mathcal{Y}^{s'}, \ldots, \mathcal{Y}^{|S|}$.
- At each step a generator $\bar{\mathcal{Y}}^{s'} := \hat{\mathcal{Y}}^{s'}$ is fixed.
- At each step the combined generators must 'generate' $\mathcal{Y}_{\mathcal{N}}$.

An alternative *global sequentially minimum sequence of generator sets* is defined as follows:

$$\bar{\mathcal{Y}}^{1} = \arg\min |\hat{\mathcal{Y}}^{1}| \tag{13}$$

$$st. \quad \left(\hat{\mathcal{Y}}^{1} \oplus \bigoplus_{s=2}^{|S|} \mathcal{Y}^{s} \right)_{\mathcal{N}} = \mathcal{Y}_{\mathcal{N}}$$

$$\hat{\mathcal{Y}}^{1} \subseteq \mathcal{Y}^{1}_{\mathcal{N}} \tag{14}$$

For a general $s' \in \mathcal{S}$

$$\bar{\mathcal{Y}}^{s'} = \arg\min |\hat{\mathcal{Y}}^{s'}| \tag{15}$$

$$st. \quad \left(\left(\bigoplus_{s=1}^{s'-1} \hat{\mathcal{Y}}^{s} \right) \oplus \bar{\mathcal{Y}}^{s'} \oplus \left(\bigoplus_{s=2}^{|S|} \right) \right)_{\mathcal{N}} = \mathcal{Y}_{\mathcal{N}}$$

$$\hat{\mathcal{Y}}^{s'} \subseteq \mathcal{Y}^{s'}_{\mathcal{N}} \tag{16}$$

A pitfall of MGS1 and MGS2 is that the metrics they produce are not unique.

References

T. Dietz, K. Klamroth, K. Kraus, S. Ruzika, L. E. Schäfer, B. Schulze, M. Stiglmayr, M. M. Wiecek, Introducing multiobjective complex systems, European Journal of Operational Research 280 (2020) 581–596.

A. Kerberenes, Multiobjective optimization for complex systems, Ph.D. thesis, Université Paris sciences et lettres, 2022.

A. Kerbérénès, D. Vanderpooten, J.-M. Vanpeperstraete, Computing efficiently the nondominated subset of a set sum, International Transactions in Operational Research 30 (2022) 3455–3478. doi:10.1111/itor.13191.