Can we approximate a weight set decomposition?

Stefan Ruzika*, Pascal Halffmann*

RAMOO Workshop, 15th November 2018

* Technische Universität Kaiserslautern

These results have been acquired during the

vOpt Project

Exact Efficient Solution of Mixed-Integer Programming Problems with Multiple Objective Functions

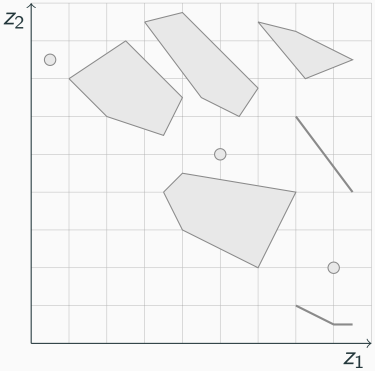
https://vopt-anr-dfg.univ-nantes.fr/

DFG grant RU 1524/4-1

I. Introduction

Nondominance and Weighted Sum

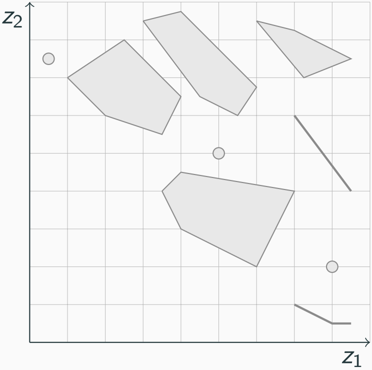
$min_{x \in X}\ z(x) = Cx$ with $Y = z(X)$.



Nondominance and Weighted Sum

$y^* \in Y$ is nondominated, if $\nexists\, y \in Y$ such that $y_i \leq y_i^*$, $\forall i$ and $y \neq y^*$. Y_N is the set of nondominated images.

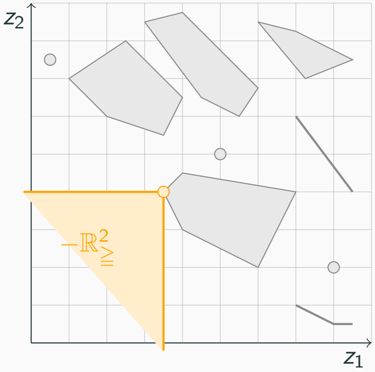
$$min_{x \in X}\ z(x) = Cx \text{ with } Y = z(X).$$



Nondominance and Weighted Sum

$y^* \in Y$ is nondominated, if $\nexists\, y \in Y$ such that $y_i \leq y_i^*,\ \forall i$ and $y \neq y^*$. Y_N is the set of nondominated images.

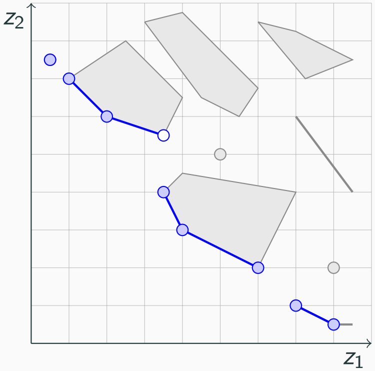
$min_{x \in X}\ z(x) = Cx$ with $Y = z(X)$.



Nondominance and Weighted Sum

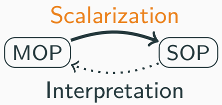
$y^* \in Y$ is nondominated, if $\nexists\, y \in Y$ such that $y_i \leq y_i^*$, $\forall i$ and $y \neq y^*$. Y_N is the set of nondominated images.

$min_{x \in X}\ z(x) = Cx$ with $Y = z(X)$.

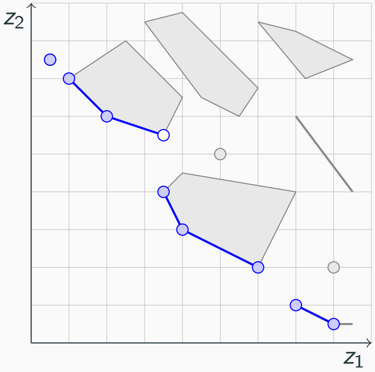


Nondominance and Weighted Sum

$y^* \in Y$ is nondominated, if $\nexists\, y \in Y$ such that $y_i \leq y_i^*$, $\forall i$ and $y \neq y^*$. Y_N is the set of nondominated images.

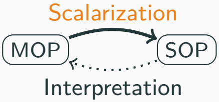


$min_{x \in X}\, z(x) = Cx$ with $Y = z(X)$.



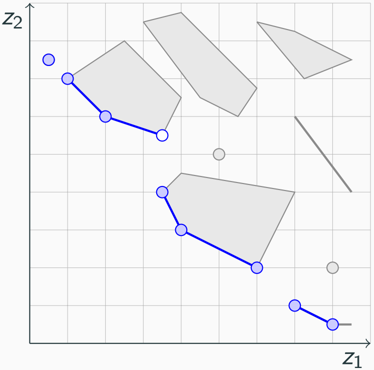
Nondominance and Weighted Sum

$y^* \in Y$ is nondominated, if $\nexists\, y \in Y$ such that $y_i \leq y^*_i$, $\forall i$ and $y \neq y^*$. Y_N is the set of nondominated images.



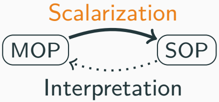
Weighted Sum:
$y^* \in Y$ is supported, if $\exists \lambda \in \mathbb{R}^p_>$ such that $\lambda^T y^* = \min_{y \in Y} \lambda^T y$.
Y_{SN} set of supported images.
Y_{ESN} set of extreme supported images.

$$min_{x \in X}\, z(x) = Cx \text{ with } Y = z(X).$$



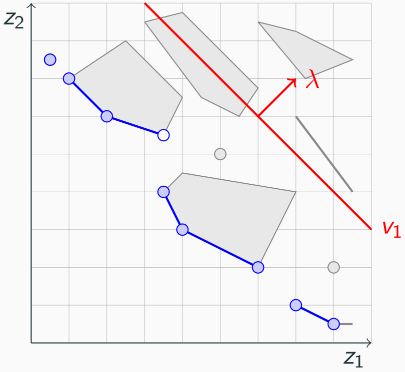
Nondominance and Weighted Sum

$y^* \in Y$ is nondominated, if $\nexists\, y \in Y$ such that $y_i \leq y_i^*$, $\forall i$ and $y \neq y^*$. Y_N is the set of nondominated images.



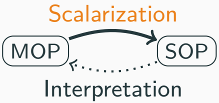
Weighted Sum:
$y^* \in Y$ is supported, if $\exists \lambda \in \mathbb{R}^p_>$ such that $\lambda^T y^* = \min_{y \in Y} \lambda^T y$.
Y_{SN} set of supported images.
Y_{ESN} set of extreme supported images.

$$min_{x \in X}\, z(x) = Cx \text{ with } Y = z(X).$$



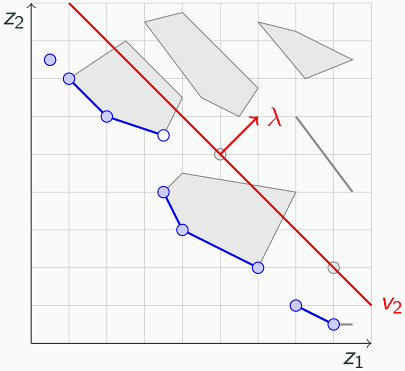
Nondominance and Weighted Sum

$y^* \in Y$ is nondominated, if $\nexists\, y \in Y$ such that $y_i \leq y_i^*$, $\forall i$ and $y \neq y^*$. Y_N is the set of nondominated images.



Weighted Sum:
$y^* \in Y$ is supported, if $\exists \lambda \in \mathbb{R}^p_>$ such that $\lambda^T y^* = \min_{y \in Y} \lambda^T y$.
Y_{SN} set of supported images.
Y_{ESN} set of extreme supported images.

$min_{x \in X}\, z(x) = Cx$ with $Y = z(X)$.

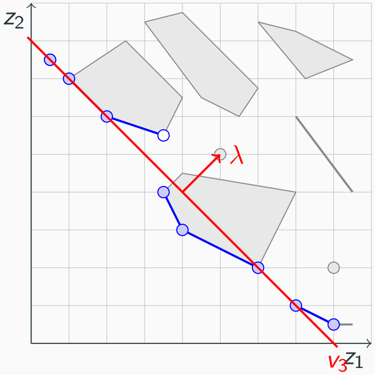


Nondominance and Weighted Sum

$y^* \in Y$ is nondominated, if $\nexists\, y \in Y$ such that $y_i \leq y_i^*$, $\forall i$ and $y \neq y^*$. Y_N is the set of nondominated images.

$$min_{x \in X}\; z(x) = Cx \text{ with } Y = z(X).$$

$$\text{MOP} \underset{\text{Interpretation}}{\overset{\text{Scalarization}}{\rightleftarrows}} \text{SOP}$$



Weighted Sum:

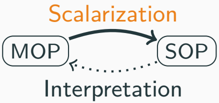
$y^* \in Y$ is supported, if $\exists \lambda \in \mathbb{R}^p_>$ such that $\lambda^T y^* = \min_{y \in Y} \lambda^T y$.

Y_{SN} set of supported images.

Y_{ESN} set of extreme supported images.

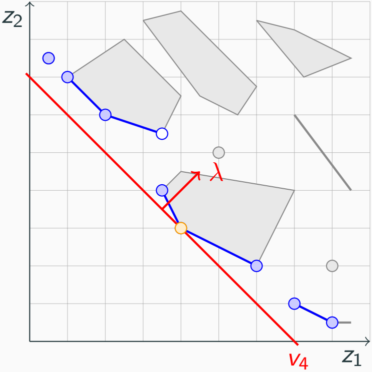
Nondominance and Weighted Sum

$y^* \in Y$ is nondominated, if $\nexists\, y \in Y$ such that $y_i \leq y_i^*$, $\forall i$ and $y \neq y^*$. Y_N is the set of nondominated images.



Weighted Sum:
$y^* \in Y$ is supported, if $\exists \lambda \in \mathbb{R}^p_{>}$ such that $\lambda^T y^* = \min_{y \in Y} \lambda^T y$.
Y_{SN} set of supported images.
Y_{ESN} set of extreme supported images.

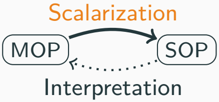
$$min_{x \in X}\, z(x) = Cx \text{ with } Y = z(X).$$

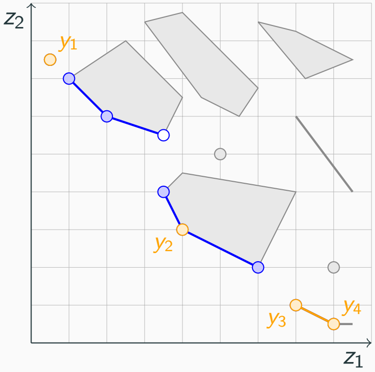


Nondominance and Weighted Sum

$y^* \in Y$ is nondominated, if $\nexists\, y \in Y$ such that $y_i \leq y_i^*$, $\forall i$ and $y \neq y^*$. Y_N is the set of nondominated images.

$$min_{x \in X}\ z(x) = Cx \text{ with } Y = z(X).$$

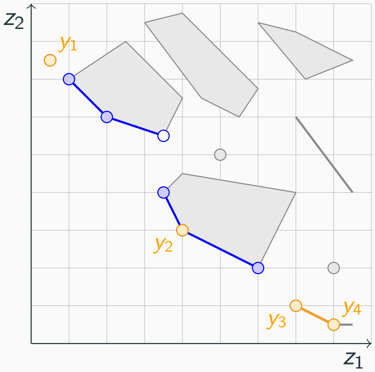




Weighted Sum:
$y^* \in Y$ is supported, if $\exists \lambda \in \mathbb{R}^p_>$ such that $\lambda^T y^* = \min_{y \in Y} \lambda^T y$.
Y_{SN} set of supported images.
Y_{ESN} set of extreme supported images.

The Weight Set Decomposition

$min_{x \in X}\ z(x) = Cx$ with $Y = z(X)$.

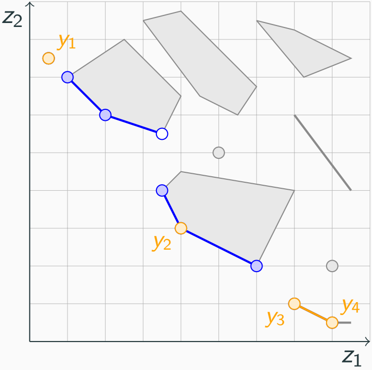


The Weight Set Decomposition

$min_{x \in X}\ z(x) = Cx$ with $Y = z(X)$.

Normalized Weight Set:

$$\Lambda := \left\{ \lambda \in \mathbb{R}^p_{\geqq} : \sum_{k=1}^{p} \lambda_k = 1 \right\} \subseteq \mathbb{R}^{p-1},$$

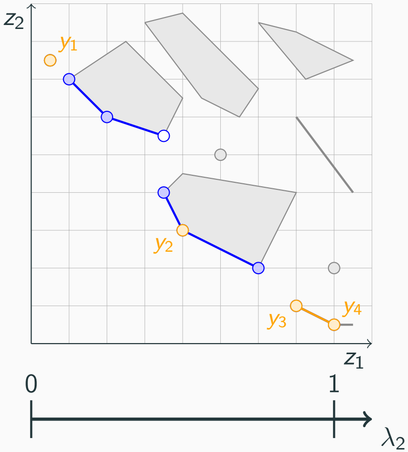


The Weight Set Decomposition

$min_{x \in X}\ z(x) = Cx$ with $Y = z(X)$.

Normalized Weight Set:

$$\Lambda := \left\{ \lambda \in \mathbb{R}^p_{\geqq} : \sum_{k=1}^{p} \lambda_k = 1 \right\} \subseteq \mathbb{R}^{p-1},$$



The Weight Set Decomposition

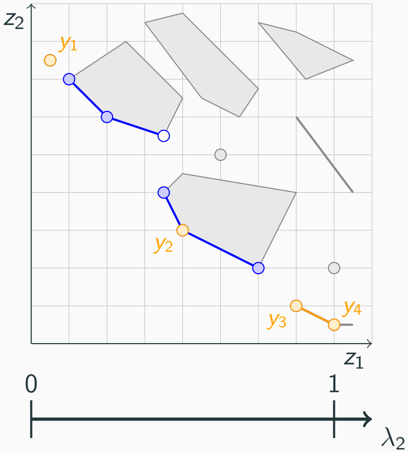
$min_{x \in X}\ z(x) = Cx$ with $Y = z(X)$.

Normalized Weight Set:

$$\Lambda := \left\{ \lambda \in \mathbb{R}^p_{\geqq} : \sum_{k=1}^{p} \lambda_k = 1 \right\} \subseteq \mathbb{R}^{p-1},$$

Weight Set Component:

$$\Lambda(y^*) := \left\{ \lambda \in \Lambda : \lambda^T y^* = \min_{y \in Y} \lambda^T y \right\} \subseteq \Lambda.$$



The Weight Set Decomposition

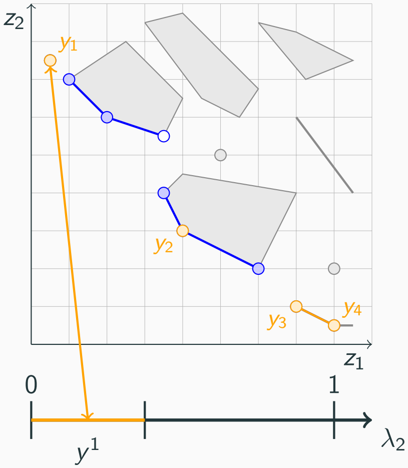
$min_{x \in X}\ z(x) = Cx$ with $Y = z(X)$.

Normalized Weight Set:

$$\Lambda := \left\{ \lambda \in \mathbb{R}^p_{\geqq} : \sum_{k=1}^{p} \lambda_k = 1 \right\} \subseteq \mathbb{R}^{p-1},$$

Weight Set Component:

$$\Lambda(y^*) := \left\{ \lambda \in \Lambda : \lambda^T y^* = \min_{y \in Y} \lambda^T y \right\} \subseteq \Lambda.$$



The Weight Set Decomposition

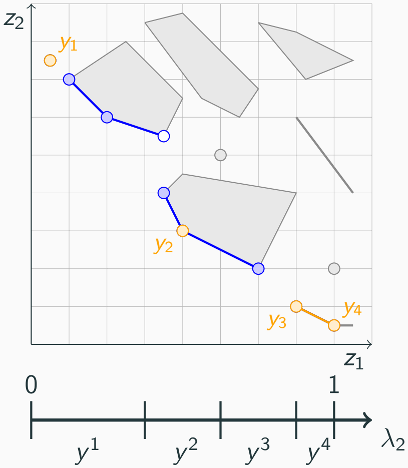
$min_{x \in X}\ z(x) = Cx$ with $Y = z(X)$.

Normalized Weight Set:

$$\Lambda := \left\{ \lambda \in \mathbb{R}^p_{\geqq} : \sum_{k=1}^{p} \lambda_k = 1 \right\} \subseteq \mathbb{R}^{p-1},$$

Weight Set Component:

$$\Lambda(y^*) := \left\{ \lambda \in \Lambda : \lambda^T y^* = \min_{y \in Y} \lambda^T y \right\} \subseteq \Lambda.$$



The Weight Set Decomposition

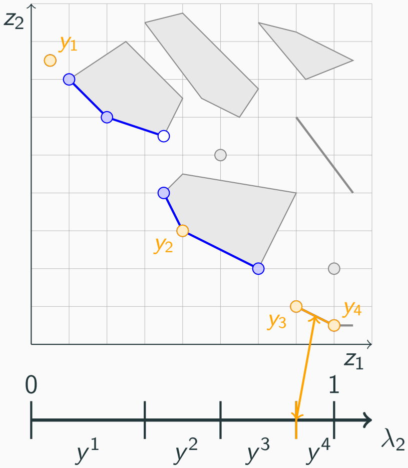
$min_{x \in X}\, z(x) = Cx$ with $Y = z(X)$.

Normalized Weight Set:

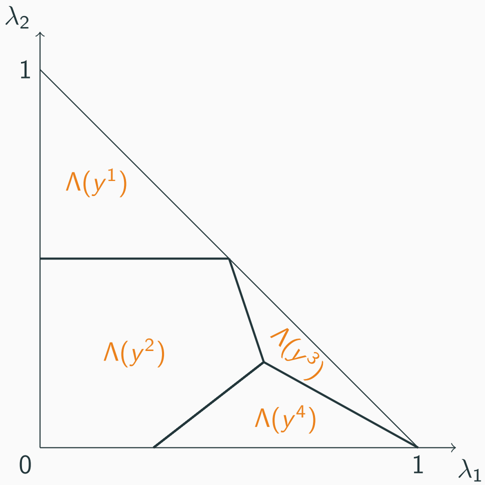
$$\Lambda := \left\{ \lambda \in \mathbb{R}^p_{\geqq} : \sum_{k=1}^{p} \lambda_k = 1 \right\} \subseteq \mathbb{R}^{p-1},$$

Weight Set Component:

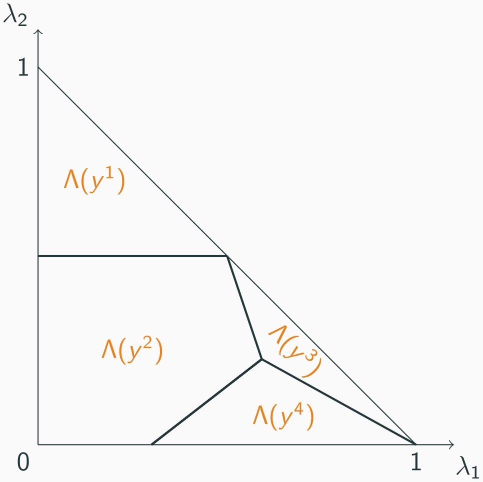
$$\Lambda(y^*) := \left\{ \lambda \in \Lambda : \lambda^T y^* = \min_{y \in Y} \lambda^T y \right\} \subseteq \Lambda.$$



Weight Set for Tri Objective Problems

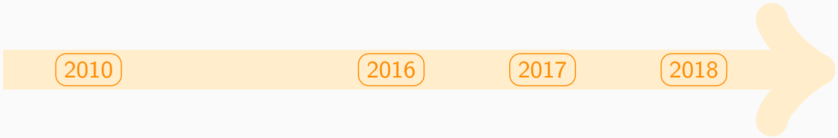


Weight Set for Tri Objective Problems



Weight set components are convex polytopes and the components of the ESN images decompose the weight set.

Recent Work



Recent Work

2010

Przybylski et al. First exact algorithm for three objectives, with recursion usable for > 3 objectives.

Özpeynirci and Köksalan First non-recursive exact algorithm for more than three objectives.

2016

Alves and Costa Exact algorithm for three objectives.

2017

Bökler and Mutzel First output sensitive exact algorithm for finding ESN images for combinatorial problems.

2018

Halffmann et al. Output sensitive graphical exact algorithm for the weight set decomposition using Dichotomic Search (in progress).

II. Approximation of a Weight Set Decomposition

Motivation

- Even though output sensitive, algorithms are not necessary polynomial time algorithms.

Motivation

- Even though output sensitive, algorithms are not necessary polynomial time algorithms.
- Also true, if the weighted sum problem can be solved in polynomial time.

Motivation

- Even though output sensitive, algorithms are not necessary polynomial time algorithms.
- Also true, if the weighted sum problem can be solved in polynomial time.
- Sometimes computing an exact weight set component is too time consuming or simply not necessary.

Motivation

- Even though output sensitive, algorithms are not necessary polynomial time algorithms.
- Also true, if the weighted sum problem can be solved in polynomial time.
- Sometimes computing an exact weight set component is too time consuming or simply not necessary.

What about approximation algorithms or heuristics? This has not been done before!

How to Approximate the Weight Set Decomposition?

Different Variants:

How to Approximate the Weight Set Decomposition?

Different Variants:

- Approximate the weight set decomposition as a whole.

How to Approximate the Weight Set Decomposition?

Different Variants:

- Approximate the weight set decomposition as a whole.
- Approximate the weight set components.

How to Approximate the Weight Set Decomposition?

Different Variants:

- Approximate the weight set decomposition as a whole.
- Approximate the weight set components.

How to Approximate the Weight Set Decomposition?

Different Variants:

- Approximate the weight set decomposition as a whole.
- Approximate the weight set components.

Challenges:

How to Approximate the Weight Set Decomposition?

Different Variants:

- Approximate the weight set decomposition as a whole.
- Approximate the weight set components.

Challenges:

- For a weight set component $\Lambda(y)$ a set $L(y) \subseteq \Lambda$ has to be returned that meets certain requirements.

How to Approximate the Weight Set Decomposition?

Different Variants:

- Approximate the weight set decomposition as a whole.
- Approximate the weight set components.

Challenges:

- For a weight set component $\Lambda(y)$ a set $L(y) \subseteq \Lambda$ has to be returned that meets certain requirements.
- An appropriate quality measure of the approximation $L(y)$ of $\Lambda(y)$.

How to Approximate the Weight Set Decomposition?

Different Variants:

- Approximate the weight set decomposition as a whole.
- Approximate the weight set components.

Challenges:

- For a weight set component $\Lambda(y)$ a set $L(y) \subseteq \Lambda$ has to be returned that meets certain requirements.
- An appropriate quality measure of the approximation $L(y)$ of $\Lambda(y)$.
- An approximation factor or bound and additional requirements.

Quality Measures

Quality Measures

Ratio of the Hypervolume

Quality Measures

Ratio of the Hypervolume

- $\frac{V(L(y))}{V(\Lambda(y))}$.
- Easy to comprehend and visualize.
- Hard to compute, at least for the weight set component.

Quality Measures

Ratio of the Hypervolume

- $\frac{V(L(y))}{V(\Lambda(y))}$.
- Easy to comprehend and visualize.
- Hard to compute, at least for the weight set component.

Hausdorff Distance

Quality Measures

Ratio of the Hypervolume

- $\frac{V(L(y))}{V(\Lambda(y))}$.
- Easy to comprehend and visualize.
- Hard to compute, at least for the weight set component.

Hausdorff Distance

- $\delta(L(y), \Lambda(y)) := \max\{\max\{D(\lambda, L(y)), \lambda \in \Lambda\} \max\{D(l, \Lambda(y)), l \in L(y)\}\}$,
  with $D(f, G) := \min\{d(f, g), g \in G\}$ and $d(.,.)$ is the euclidean metric.
- Use as measure $1 - \frac{\delta(L(y),\Lambda(y))}{\max_{a,b\in\Lambda(y)} d(a,b)}$.
- Easier to compute but harder to comprehend.

Approximation of a Weight Set Component

Inner Approximation

$\underline{L}(y)$ is an inner approximation if $\underline{L}(y) \subseteq \Lambda(y)$. It is an inner approximation component if $\underline{L}(y)$ is a convex polytope.

Approximation of a Weight Set Component

Inner Approximation

$\underline{L}(y)$ is an inner approximation if $\underline{L}(y) \subseteq \Lambda(y)$. It is an inner approximation component if $\underline{L}(y)$ is a convex polytope.

Outer Approximation

$\overline{L}(y)$ is an outer approximation if $\Lambda(y) \subseteq \overline{L}(y)$. It is an outer approximation component if $\overline{L}(y)$ is a convex polytope.

Approximation of a Weight Set Component

Inner Approximation

$\underline{L}(y)$ is an inner approximation if $\underline{L}(y) \subseteq \Lambda(y)$. It is an inner approximation component if $\underline{L}(y)$ is a convex polytope.

Outer Approximation

$\overline{L}(y)$ is an outer approximation if $\Lambda(y) \subseteq \overline{L}(y)$. It is an outer approximation component if $\overline{L}(y)$ is a convex polytope.

Instead of $L(y)$ and $\Lambda(y)$, use $\underline{L}(y)$ and $\overline{L}(y)$, simplifies the computation of the quality measure.

Basic Definitions

Weight Set Decomposition Approximation Algorithm

Given an instance of a multiobjective problem, an α-Weight Set Decomposition Approximation Algorithm with $0 < \alpha \leq 1$ returns for a subset $\emptyset \neq S \subseteq Y_{ESN}$ of the extreme supported nondominated images sets $L(y) \subseteq \Lambda, y \in S$ such that each $L(y)$ approximates $\Lambda(y)$ by at least α. Further, the algorithm is a polynomially α-Weight Set Decomposition Approximation Algorithm if it additionally runs in polynomial time.

Basic Definitions

Weight Set Decomposition Approximation Algorithm

Given an instance of a multiobjective problem, an α-Weight Set Decomposition Approximation Algorithm with $0 < \alpha \leq 1$ returns for a subset $\emptyset \neq S \subseteq Y_{ESN}$ of the extreme supported nondominated images sets $L(y) \subseteq \Lambda, y \in S$ such that each $L(y)$ approximates $\Lambda(y)$ by at least α. Further, the algorithm is a polynomially α-Weight Set Decomposition Approximation Algorithm if it additionally runs in polynomial time.

Weight Set Decomposition Approximation Heuristic

Each iteration i the algorithm returns a subset S^i and an α^i-approximation of the weight set component of each $y \in S^i$. We require that $\lim_{i\to\infty} S^i = Y_{ESN}$ and $\lim_{i\to\infty} \alpha^i = 1$ and each iteration can be done in polynomial time.

Basic Definitions

Weight Set Decomposition Approximation Algorithm

Given an instance of a multiobjective problem, an α-Weight Set Decomposition Approximation Algorithm with $0 < \alpha \leq 1$ returns for a subset $\emptyset \neq S \subseteq Y_{ESN}$ of the extreme supported nondominated images sets $L(y) \subseteq \Lambda, y \in S$ such that each $L(y)$ approximates $\Lambda(y)$ by at least α. Further, the algorithm is a polynomially α-Weight Set Decomposition Approximation Algorithm if it additionally runs in polynomial time.

Weight Set Decomposition Approximation Heuristic

Each iteration i the algorithm returns a subset S^i and an α^i-approximation of the weight set component of each $y \in S^i$. We require that $\lim_{i\to\infty} S^i = Y_{ESN}$ and $\lim_{i\to\infty} \alpha^i = 1$ and each iteration can be done in polynomial time.

Convergence Rate

Let $a^i(y)$ the approximation factor for $\Lambda(y)$ in iteration i. We measure the convergence rate for one component of the heuristic by

$$\limsup_{i\to\infty} \frac{a^{i+1}(y)}{a^i(y)}.$$

III. Approximation Algorithms and Heuristics

Overview of the Methods

Overview of the Methods

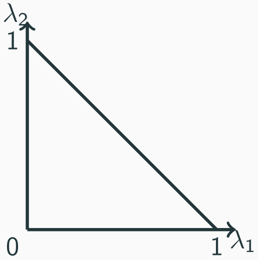
Point Search

Subdivide the weight set into easy computable full-dimensional polytopes.
Evaluate the vertices:
If one solution is optimal for all vertices, then this polytope belongs to the component of this solution, otherwise subdivide again.

Overview of the Methods

Point Search

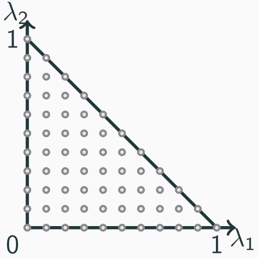
Subdivide the weight set into easy computable full-dimensional polytopes.
Evaluate the vertices:
If one solution is optimal for all vertices, then this polytope belongs to the component of this solution, otherwise subdivide again.



Overview of the Methods

Point Search

Subdivide the weight set into easy computable full-dimensional polytopes.
Evaluate the vertices:
If one solution is optimal for all vertices, then this polytope belongs to the component of this solution, otherwise subdivide again.



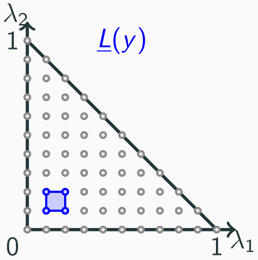
Overview of the Methods

Point Search

Subdivide the weight set into easy computable full-dimensional polytopes.

Evaluate the vertices:

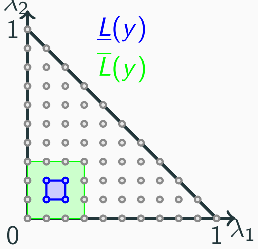
If one solution is optimal for all vertices, then this polytope belongs to the component of this solution, otherwise subdivide again.



Overview of the Methods

Point Search

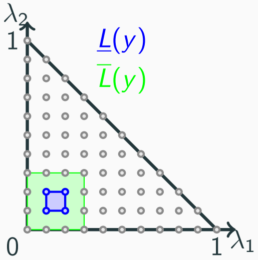
Subdivide the weight set into easy computable full-dimensional polytopes.
Evaluate the vertices:
If one solution is optimal for all vertices, then this polytope belongs to the component of this solution, otherwise subdivide again.



Overview of the Methods

Point Search

Subdivide the weight set into easy computable full-dimensional polytopes.
Evaluate the vertices:
If one solution is optimal for all vertices, then this polytope belongs to the component of this solution, otherwise subdivide again.



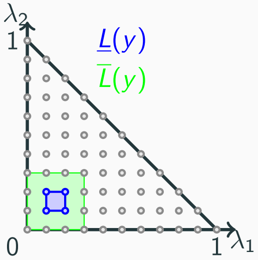
Line Search

Evaluate lines passing through the weight set by Dichotomic Search. Take the convex hull of breakpoints belonging to the same optimal solution.

Overview of the Methods

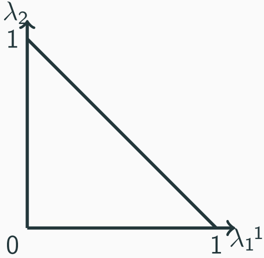
Point Search

Subdivide the weight set into easy computable full-dimensional polytopes.
Evaluate the vertices:
If one solution is optimal for all vertices, then this polytope belongs to the component of this solution, otherwise subdivide again.



Line Search

Evaluate lines passing through the weight set by Dichotomic Search. Take the convex hull of breakpoints belonging to the same optimal solution.



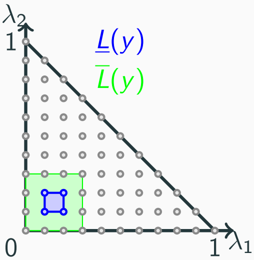
Overview of the Methods

Point Search

Subdivide the weight set into easy computable full-dimensional polytopes.

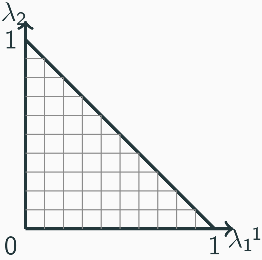
Evaluate the vertices:

If one solution is optimal for all vertices, then this polytope belongs to the component of this solution, otherwise subdivide again.



Line Search

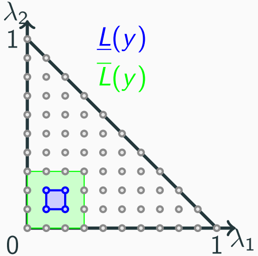
Evaluate lines passing through the weight set by Dichotomic Search. Take the convex hull of breakpoints belonging to the same optimal solution.



Overview of the Methods

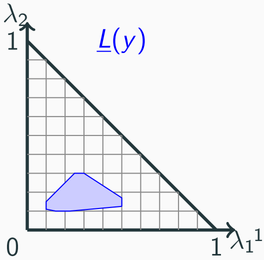
Point Search

Subdivide the weight set into easy computable full-dimensional polytopes.
Evaluate the vertices:
If one solution is optimal for all vertices, then this polytope belongs to the component of this solution, otherwise subdivide again.



Line Search

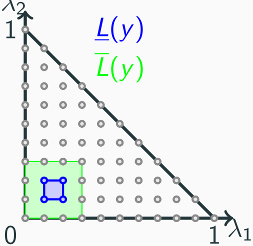
Evaluate lines passing through the weight set by Dichotomic Search. Take the convex hull of breakpoints belonging to the same optimal solution.



Overview of the Methods

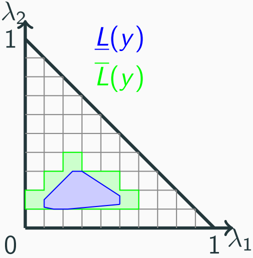
Point Search

Subdivide the weight set into easy computable full-dimensional polytopes.
Evaluate the vertices:
If one solution is optimal for all vertices, then this polytope belongs to the component of this solution, otherwise subdivide again.

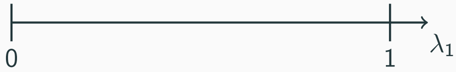


Line Search

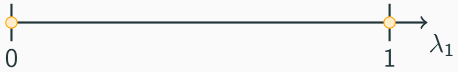
Evaluate lines passing through the weight set by Dichotomic Search. Take the convex hull of breakpoints belonging to the same optimal solution.



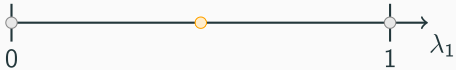
Point Search: The Biobjective Case



Point Search: The Biobjective Case

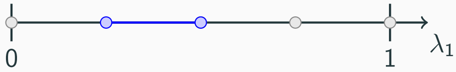


Point Search: The Biobjective Case

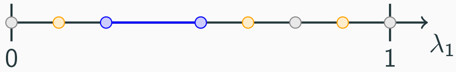


Point Search: The Biobjective Case

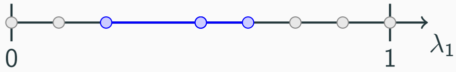
Point Search: The Biobjective Case



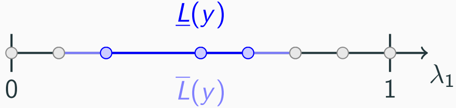
Point Search: The Biobjective Case



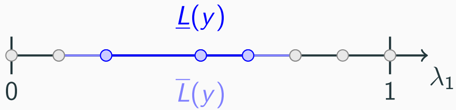
Point Search: The Biobjective Case



Point Search: The Biobjective Case



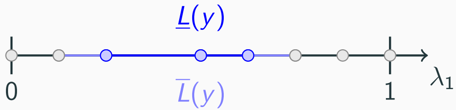
Point Search: The Biobjective Case



Result

Given $\epsilon > 0$ the Point Search gives an $\frac{1}{3}$-approximation (Hypervolume) of all components that are found in $\mathcal{O}(\frac{1}{\epsilon} T_{WS})$. In order to definitely return at least one weight set component, we require $\frac{1}{\epsilon} > |Y_{ESN}|$. This method has a convergence rate of $\frac{1}{2}$.

Point Search: The Biobjective Case



Result

Given $\epsilon > 0$ the Point Search gives an $\frac{1}{3}$-approximation (Hypervolume) of all components that are found in $\mathcal{O}(\frac{1}{\epsilon} T_{WS})$. In order to definitely return at least one weight set component, we require $\frac{1}{\epsilon} > |Y_{ESN}|$. This method has a convergence rate of $\frac{1}{2}$.

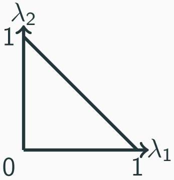
Due to the choice of ϵ this can be only used as an heuristic.

Point Search: The Multiobjective Case

Different Subdivisions:

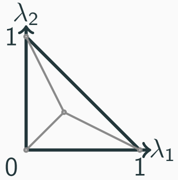
Point Search: The Multiobjective Case

Different Subdivisions:



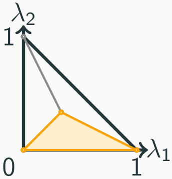
Point Search: The Multiobjective Case

Different Subdivisions:



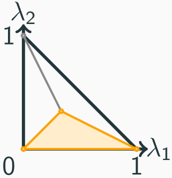
Point Search: The Multiobjective Case

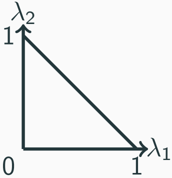
Different Subdivisions:



Point Search: The Multiobjective Case

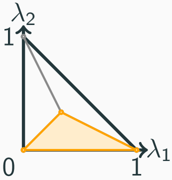
Different Subdivisions:

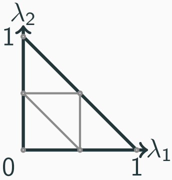




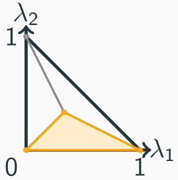
Point Search: The Multiobjective Case

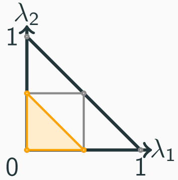
Different Subdivisions:





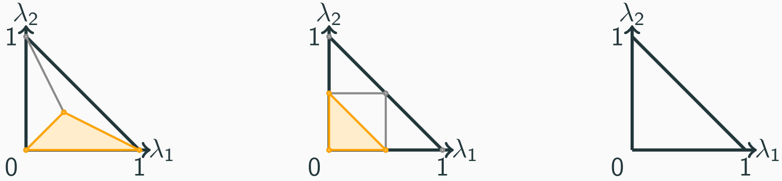
Different Subdivisions:





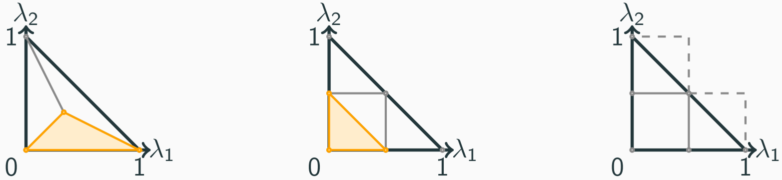
Point Search: The Multiobjective Case

Different Subdivisions:



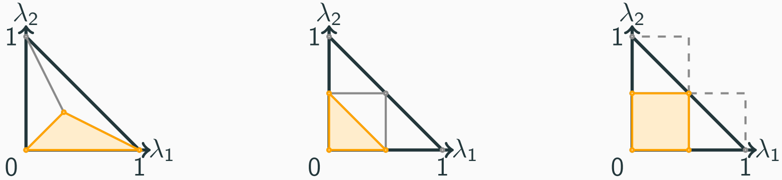
Point Search: The Multiobjective Case

Different Subdivisions:



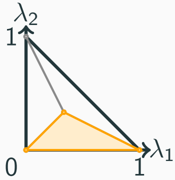
Point Search: The Multiobjective Case

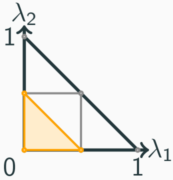
Different Subdivisions:

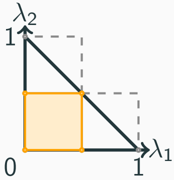


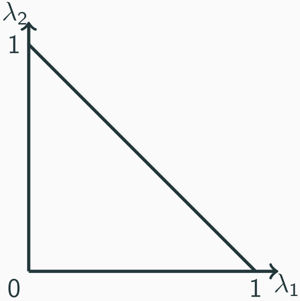
Point Search: The Multiobjective Case

Different Subdivisions:



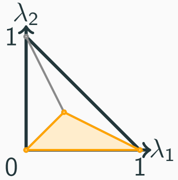


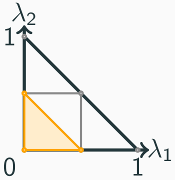


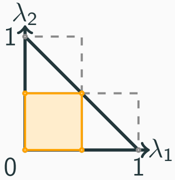


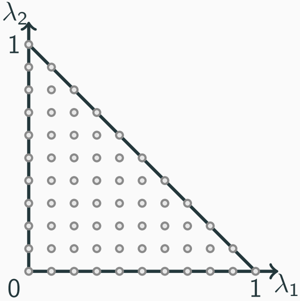
Point Search: The Multiobjective Case

Different Subdivisions:



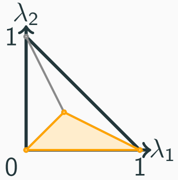


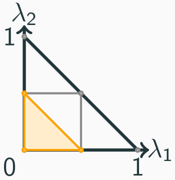


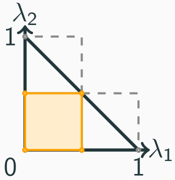


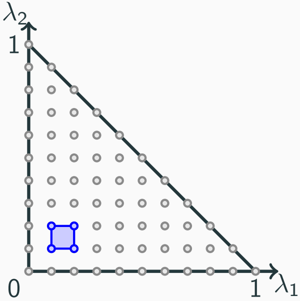
Point Search: The Multiobjective Case

Different Subdivisions:

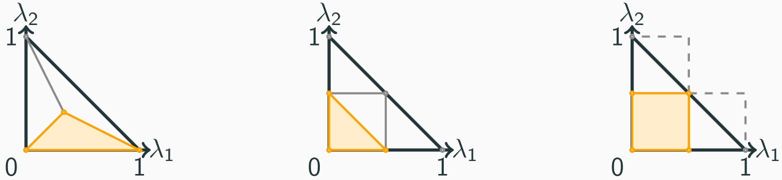


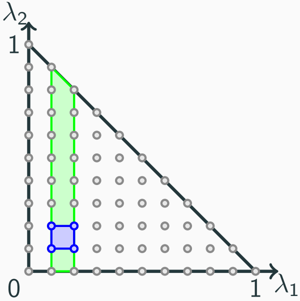




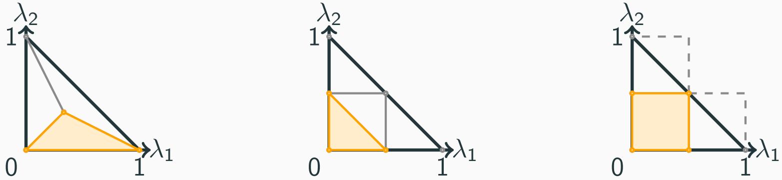


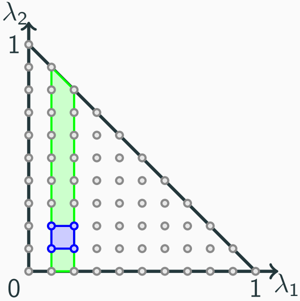
Different Subdivisions:





Different Subdivisions:



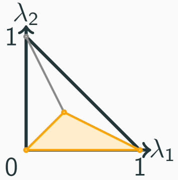


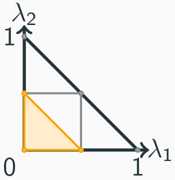
Result

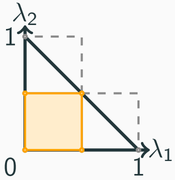
Given $\epsilon > 0$ the Point Search gives an ϵ-approximation (Hausdorff) of all components that are found in $\mathcal{O}(\frac{1}{\epsilon} T_{WS})$.

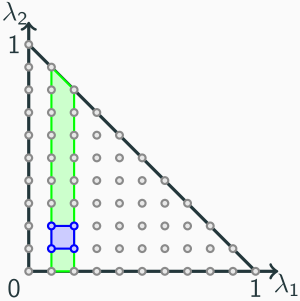
Point Search: The Multiobjective Case

Different Subdivisions:









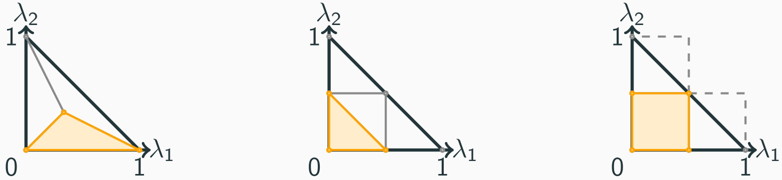
Result

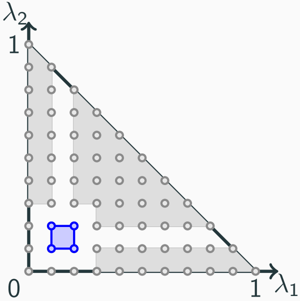
Given $\epsilon > 0$ the Point Search gives an ϵ-approximation (Hausdorff) of all components that are found in $\mathcal{O}(\frac{1}{\epsilon} T_{WS})$.

Hard to find an ϵ such that at least one component is found.

Point Search: The Multiobjective Case

Different Subdivisions:





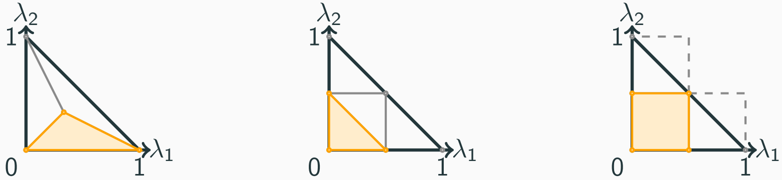
Result

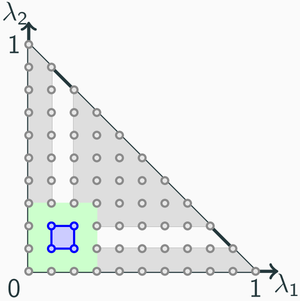
Given $\epsilon > 0$ the Point Search gives an ϵ-approximation (Hausdorff) of all components that are found in $\mathcal{O}(\frac{1}{\epsilon} T_{WS})$.

Hard to find an ϵ such that at least one component is found.

Point Search: The Multiobjective Case

Different Subdivisions:





Result

Given $\epsilon > 0$ the Point Search gives an ϵ-approximation (Hausdorff) of all components that are found in $\mathcal{O}(\frac{1}{\epsilon} T_{WS})$.

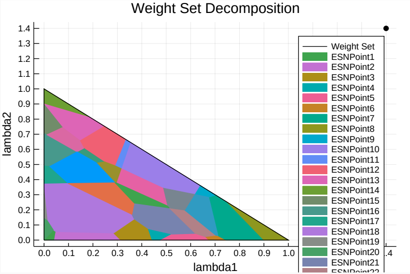
Hard to find an ϵ such that at least one component is found.

Point Search: Example

Assignment Problem with 10 items:

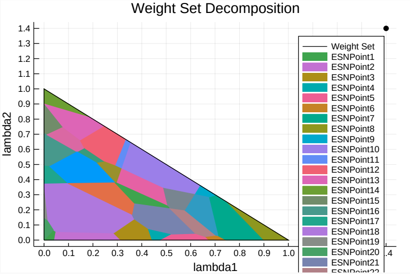
Point Search: Example

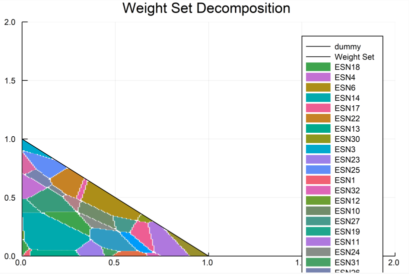
Assignment Problem with 10 items:



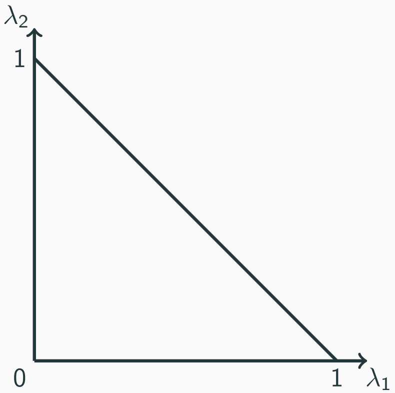
Point Search: Example

Assignment Problem with 10 items:

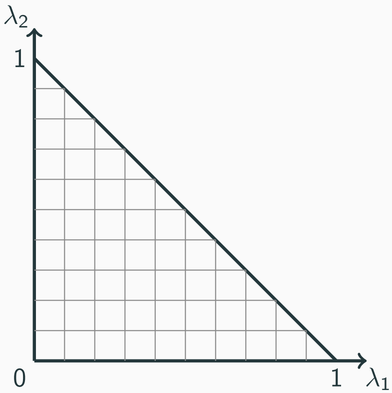




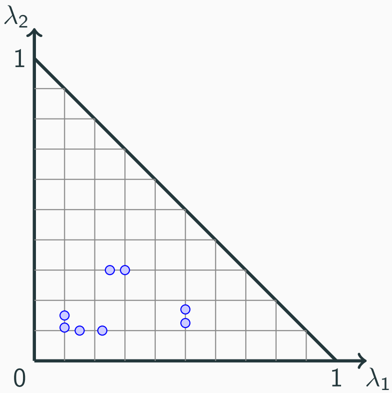
Line Search



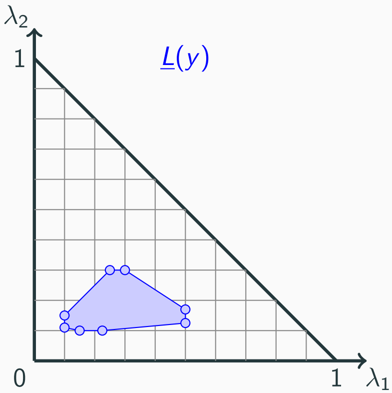
Line Search



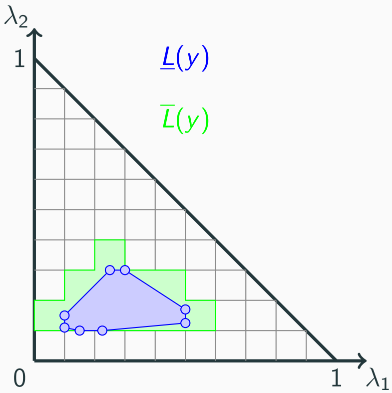
Line Search



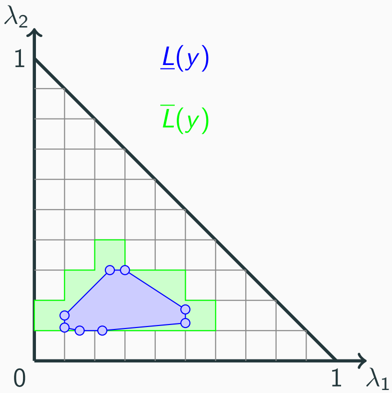
Line Search



Line Search



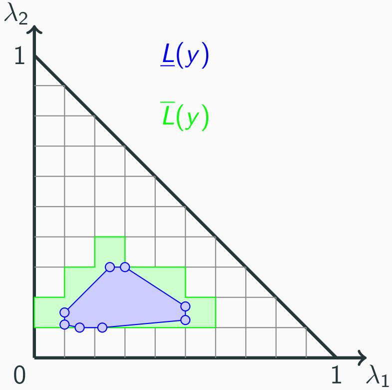
Line Search



Result

Given $\epsilon > 0$ the Line Search gives an $1 - \epsilon$-approximation (Hausdorff) of all components that are found in $\mathcal{O}(\frac{1}{\epsilon}|Y_{ESN}|T_{WS})$.

Line Search



Result

Given $\epsilon > 0$ the Line Search gives an $1-\epsilon$-approximation (Hausdorff) of all components that are found in $\mathcal{O}(\frac{1}{\epsilon}|Y_{ESN}|T_{WS})$.

Not polynomial, hard to find an ϵ such that at least one component is found. Same problem as for the Point Search for $\dim \Lambda \geq 2$.

IV. Open Question and Outlook

Open Question

Approximate Weight Set Component Problem

Let $\alpha > 0$. Given an instance of a multiobjective problem where the weighted sum problem is solvable in polynomial time. Can we, in terms of our definition, approximate at least one weight set component by factor α in polynomial time?

Open Question

Approximate Weight Set Component Problem

Let $\alpha > 0$. Given an instance of a multiobjective problem where the weighted sum problem is solvable in polynomial time. Can we, in terms of our definition, approximate at least one weight set component by factor α in polynomial time?

Conjecture: This problem is *APX-hard*, however proving this is hard by itself, as it is a mixture of an optimization and a search problem.

Outlook

- Progress on the approximation quality and convergence rate.

Outlook

- Progress on the approximation quality and convergence rate.
- Answer the general approximability of the weight set decomposition.

Outlook

- Progress on the approximation quality and convergence rate.
- Answer the general approximability of the weight set decomposition.
- What happens, if we use approximation algorithms to solve the weighted sum problem?

Thank you for your attention!

References

Maria J. Alves and João P. Costa. "Graphical exploration of the weight space in three-objective mixed integer linear programs". In: *European Journal of Operational Research* 248.1 (2016), pp. 72–83. URL: http://www.sciencedirect.com/science/article/pii/S0377221715006268

Fritz Bökler and Petra Mutzel. "Output-Sensitive Algorithms for Enumerating the Extreme Nondominated Points of Multiobjective Combinatorial Optimization Problems". In: *Algorithms - ESA 2015: 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*. Ed. by Nikhil Bansal and Irene Finocchi. Springer Berlin Heidelberg, 2015, pp. 288–299. URL: http://dx.doi.org/10.1007/978-3-662-48350-3_25

Matthias Ehrgott. *Multicriteria optimization*. Springer Science & Business Media, 2005

Özgür Özpeynirci and Murat Köksalan. "An Exact Algorithm for Finding Extreme Supported Nondominated Points of Multiobjective Mixed Integer Programs". In: *Management Science* 56.12 (2010), pp. 2302–2315. URL: http://dx.doi.org/10.1287/mnsc.1100.1248

Anthony Przybylski, Xavier Gandibleux, and Matthias Ehrgott. "A Recursive Algorithm for Finding All Nondominated Extreme Points in the Outcome Set of a Multiobjective Integer Programme". In: *INFORMS Journal on Computing* 22.3 (2010), pp. 371–386. URL: http://dx.doi.org/10.1287/ijoc.1090.0342