TILBURG UNIVERSITY

Tilburg University

A Renewed Take on Weighted Sum in Sandwich Algorithms

Koenen, Melissa; Balvert, Marleen; Fleuren, HA

Publication date:
2023

Document Version
Early version, also known as pre-print

Link to publication in Tilburg University Research Portal

Citation for published version (APA):
Koenen, M., Balvert, M., & Fleuren, HA. (2023). *A Renewed Take on Weighted Sum in Sandwich Algorithms: Modification of the Criterion Space*. (Center Discussion Paper; Vol. 2023-012). CentER, Center for Economic Research.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 12. Nov. 2023

CentER

Discussion Paper

No. 2023-012

A RENEWED TAKE ON WEIGHTED SUM IN SANDWICH ALGORITHMS: MODIFICATION OF THE CRITERION SPACE

By

Melissa F. Koenen, Marleen Balvert, Hein Fleuren

24 May 2023

ISSN 0924-7815
ISSN 2213-9532

Tilburg University

A renewed take on weighted sum in sandwich algorithms: modification of the criterion space

Melissa F. Koenen [1,2,*] Marleen Balvert [1,2] and Hein Fleuren [1,2]
[1] Tilburg School of Economics and Management, Zero Hunger Lab,
Tilburg University, Tilburg, The Netherlands
[2] Tilburg School of Economics and Management,
Department of Econometrics and Operations Research,
Tilburg University, Tilburg, The Netherlands
Correspondence*: m.f.koenen@tilburguniversity.edu

May 23, 2023

Abstract

Sandwich algorithms are commonly used to approximate the Pareto front of a multi-objective (MO) convex problem by enclosing it between an inner and outer approximation. By iteratively improving the approximations, the distance between them is minimized which gives an estimate of how well the Pareto front is approximated. A well-explainable type of sandwich algorithm is based on weighted sum scalarization (WSS), where the next set of weights is determined by the most promising inner normal of the inner approximation. As these normals can contain negative values, not every optimization will result in finding an efficient point. In order to reduce the number of searches towards the dominated part, we propose an elegant modification of the criterion space which is an advancement on the formulation of Solanki et al. In addition to being well-explainable and easy to integrate within an existing optimization procedure, this modification is theoretically able to obtain all nondominated points of an MO linear programming problem in a finite number of expansions of the inner approximation. Furthermore, we propose two heuristic approaches to determine the distance between the inner and outer approximation that can be used as an alternative for the distance calculation of Solanki et al. These heuristics incorporate the ideas of Solanki et al. and Craft et al. to obtain straightforward and faster methods. (218 words)

Keywords:

Convex multi-objective, sandwich algorithm, weighted sum scalarization, multi-objective linear programming

1 Introduction

Multi-objective optimization (MO) concerns the optimization of multiple conflicting goals, where $p \geq 2$ denotes the number of objectives. In general, one can formulate an MO problem as follows:

$$\begin{aligned} &\min \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_p(\boldsymbol{x})) \\ &\text{s.t. } \boldsymbol{x} \in \mathcal{X}, \end{aligned}$$

where f_i denotes a function that belongs to objective i, with $i = 1, \ldots, p$, and $\mathcal{X}$ denotes the decision space in $\mathbb{R}^n$. The criterion space $\mathcal{Y}$ is given by $\{\boldsymbol{y} \in \mathbb{R}^p \mid \boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) \text{ for some } \boldsymbol{x} \in \mathcal{X}\}$ and represents all possible combinations of objective values. Here, we refer to a solution when it belongs to the decision space and to a point when it belongs to the criterion space.

There is in general not one solution that minimizes all objectives simultaneously; we assume the ideal point is not attainable. This inevitably translates to trade-offs between the objectives, and to understand whether a solution is the 'best' among a set of 'similar solutions' a concept is needed which defines a 'good' solution. This concept is known as Pareto efficiency, where a solution $\bar{\boldsymbol{x}} \in \mathcal{X}$ is efficient if there is no other $\boldsymbol{x} \in \mathcal{X}$ such that $\boldsymbol{f}(\boldsymbol{x}) \leq \boldsymbol{f}(\bar{\boldsymbol{x}})$ [1]. That is, there exists no other solution which performs better in one of the objectives, without worsening the other objective(s). Furthermore, a solution $\bar{\boldsymbol{x}}$ is called weakly efficient if there is no other $\boldsymbol{x} \in \mathcal{X}$ such that $\boldsymbol{f}(\boldsymbol{x}) < \boldsymbol{f}(\bar{\boldsymbol{x}})$, i.e. there is no other solution that improves on all objectives simultaneously.

The sets of all efficient and weakly efficient solutions are denoted by $\mathcal{X}_E$ and $\mathcal{X}_{wE}$, respectively, where $\mathcal{X}_E \subseteq \mathcal{X}_{wE} \subseteq \mathcal{X}$. The set of all efficient points, or non-dominated points, are denoted by $\mathcal{Y}_N$. The weakly non-dominated points are denoted with $\mathcal{Y}_{wN}$.

Generally, one is not interested in finding (a part of) $\mathcal{X}_E$ as often a vast number of efficient solutions with the same objective function values exist, which is unnecessarily overwhelming for the decision maker (DM) as well as computationally challenging to obtain [2]. Instead one is interested in finding $\mathcal{Y}_N$, as this is often more manageable in size. This leads to both a more comprehensible overview of the trade-offs for the DM as well as a reduced computational effort.

In this paper we focus on approximating $\mathcal{Y}_N$ of a convex MO, that is, when the set $\mathcal{X}$ and all objective functions f_i are convex. This implies that $\mathcal{Y}_N$ is connected and that $\mathcal{Y} + \mathbb{R}^p_{++}$ and $\mathcal{Y}_N + \mathbb{R}^p_{++}$ are convex as well [3], with $+$ denoting the Minkowski sum. Besides general convex MO, we are interested in the specific case of obtaining the exact representation of $\mathcal{Y}_N$ of a particular instance of convex MO where the objectives and the constraint set are linear, known as multi-objective linear programming (MOLP).

Sandwich algorithms belong to the class of polyhedral approximation techniques [4] and are commonly used to approximate $\mathcal{Y}_N$ of a convex MO by sandwiching the non-dominated set between an inner and an outer approximation. Here, the inner approximation is a convex hull encompassed in $\mathcal{Y}$ formed by a set of efficient points and the outer approximation is formed by supporting halfspaces of those efficient points. Sandwich algorithms iteratively improve both the inner and outer approximation to minimize the distance between the approximations. The advantages for the DM are that 1) an upper bound on the approximation error is directly provided and 2) no interference of the DM is needed as the algorithm itself is capable of shifting its attention to a part of the non-dominated set that is currently not approximated well.

In the literature there are four types of sandwich algorithms. These are either based on weighted sum scalarization (WSS) [5, 6, 7], polyhedral gauge expansion [8], Benson's outer

approximation [9, 10, 11, 12] or 'dual' vertex enumeration [13]. In this paper we focus on WSS for the following reasons: 1) the idea behind WSS is explainable and intuitive to DMs; 2) the WSS algorithm is easily integrated within existing (convex) optimization procedures [6]. A caveat of the WSS sandwich approach, however, is that — unlike Benson's outer approximation [9] — it is not formally proven to be able to solve MOLPs to optimality. That is, although it is stated that WSS can in principle be used to find the MOLP [4], to the best of our knowledge, there does not exist any formal proof that a sandwich algorithm using WSS is able to find all efficient points in a finite number of iterations. We address this matter in our contributions.

WSS is a common technique to find efficient points, where each of the objectives receives a weight that indicates its importance (see e.g. [1]). It is formulated as follows:

$$\begin{aligned} &\min \boldsymbol{w}^\intercal \boldsymbol{f}(\boldsymbol{x}) = \sum_{i=1}^{p} w_i f_i(\boldsymbol{x}) \\ &\text{s.t. } \boldsymbol{x} \in \mathcal{X}, \end{aligned}$$

where $\boldsymbol{w}$ is a vector containing all weights w_i, and w_i denotes the weight belonging to objective i. In case all weights are non-negative, i.e. $\boldsymbol{w} \geq \mathbf{0}$, then the solution is guaranteed to be weakly efficient. In case all weights are strictly positive, i.e. $\boldsymbol{w} > \mathbf{0}$, the obtained point is even guaranteed to be efficient.

A straightforward way to obtain efficient points of the convex MO is to sample various weights and solve the corresponding weighted sum problem. However, an even spread of weights generally does not result in efficient points spread evenly across $\mathcal{Y}_N$ [14]. Additionally there are infinitely many weights that result in the same efficient point. A more resourceful selection of weights is commonly achieved by integrating the WSS in a sandwich algorithm, as it picks a weight based on the current inner approximation at a point that is the furthest from the outer approximation.

Solanki et al. [5] are one of the first to approximate the nondominated set of a convex MO using WSS with a sandwich algorithm. They make use of the polyhedral structure of the inner approximation to obtain the inner normals of so far generated halfspaces. In their formulation the inner normals for $p = 2$ are always positive, and using them as weights is therefore guaranteed to find an efficient point. Either an existing efficient point is found or a new efficient point is found that expands the inner approximation. In both cases the outer approximation is tightened, as we cannot improve further using this inner normal. At each iteration one needs to decide which face — and corresponding normal — of the polyhedral inner approximation is used to obtain the next point. Solanki et al. [5] pick the face that has maximum distance between the inner and the outer approximation, which they determine exactly for each normal of the current inner approximation using a linear program (LP).

In case $p \geq 3$ the obtained normals of the inner approximation might contain negative values [5]. Using these normals as weights does not guarantee to find an efficient point or may not result in a solution if $\mathcal{Y}$ is unbounded. Solanki et al. [5] address the latter problem by placing an upper bound on the objectives and argue that using negative normals can nonetheless help in finding new positive normals as the inner approximation is expanded.

Craft et al. [6] circumvent the negative normals altogether: in case a halfspace of the inner approximation has a negative normal, they determine a weighted combination of previously found weights that belong to the vertices spanning the plane. This weighted combination is found by solving a small non-linear problem that finds a weight that is maximally different from these previously found weights. As the distance approach of

Solanki et al. [5] can be time consuming, Craft et al. [6] also propose a heuristic approach to determine the distance between the inner approximation and the outer approximation which does not rely on solving an LP.

Rennen et al. [7] enhance the above procedures by including so-called dummy points in the inner approximation. By extending the inner approximation, the relevant planes do not have negative normals anymore, which helps to find efficient points. In their paper, they compare the performances of Solanki et al. [5], Craft et al. [6] and Klamroth et al. [8] with the approach of Solanki et al. [5] including dummy points. The overall performance of the dummy technique outperformed the others in terms of the number of optimizations that needs to be performed. Here, the lack of performance of Solanki et al. [5] is mainly attributed to the algorithm describing the complete criterion space including the dominated part, hence many unnecessary LPs are solved.

This work has several contributions. First, in order to reduce the search towards the dominated part of the criterion space, we will propose an elegant modification of the criterion space. This modification has some useful properties: (1) it allows to find all efficient points for the MOLP in a finite number of iterations, and (2) its performance in terms of the maximum number of WSS LPs that needs to be solved is, unlike [5], not dependent on the chosen upper bound as long as this upper bound is larger than the nadir point. Besides its theoretical properties and explainability, the modification can be easily incorporated within an existing optimization procedure. As a second contribution, we propose two heuristic approaches to determine the distance between the inner and outer approximation and address whether it is always beneficial to use a distance metric within a WSS sandwich algorithm. The two proposed heuristic approaches combine the simplicity of the distance approach of Solanki et al. [5] with the insights of Craft et al. [6] to obtain faster approaches that are straightforward to implement.

2 Notation

In general, let $\operatorname{conv}\{X\}$ denote the convex hull of X, $|X|$ denote the cardinality of X and $\|\boldsymbol{x}\|$ denote the Euclidean norm of x. Let $\boldsymbol{e}^i$ denote the standard unit vector, where the i-th element is 1 and all others are zero. In case $+$ is used in context of polytopes, it denotes the Minkowski sum. Furthermore, let $\bigoplus$ denote the sum of Minkowski sums.

In line with the notation of Rennen et al. [7], let IPS denote the inner approximation, and let OPS denote the outer approximation, where it holds that the non-dominated criterion space $\mathcal{Y}_N$ is sandwiched between IPS and OPS. That is, $\text{IPS} \subseteq \mathcal{Y}_N + \mathbb{R}^p_{++}$ and $\mathcal{Y}_N \subseteq \text{OPS} + \mathbb{R}^p_{++}$. Furthermore, let $\mathcal{H}(\boldsymbol{w}, b) = \{\boldsymbol{y} \mid \boldsymbol{w}^\intercal \boldsymbol{y} = b\}$ be a hyperplane where $\boldsymbol{w}$ is the normal vector and b the offset, and let $\mathcal{H}^{\geq}(\boldsymbol{w}, b) = \{\boldsymbol{y} \mid \boldsymbol{w}^\intercal \boldsymbol{y} \geq b\}$ denote a halfspace where $\boldsymbol{w}$ is the inner normal. In case $\|\boldsymbol{w}\| = 1$, $\boldsymbol{w}$ is an inner unit normal.

Let $\boldsymbol{y}^{\mathrm{A}_i}$ be the anchor point of objective $i = 1, \ldots, p$, that is, $\boldsymbol{y}^{\mathrm{A}_i}$ is the point that minimizes objective i individually such that $\boldsymbol{y}^{\mathrm{A}_i} = \arg\min\{y_i \mid \boldsymbol{y} \in \mathcal{Y}\}$. Let $\boldsymbol{y}^{\mathrm{I}} = \left(y_1^{\mathrm{I}}, \ldots, y_p^{\mathrm{I}}\right)$ be the ideal point that is constructed using the the minimum of all objectives, where $y_i^{\mathrm{I}} = y_i^{\mathrm{A}_i}$. Here, we assume the ideal point to be unattainable. Let $\boldsymbol{y}^{\mathrm{UB}}$ denote an upper bound on the objective function values. An insightful choice for $\boldsymbol{y}^{\mathrm{UB}}$ is the nadir point $\boldsymbol{y}^{\mathrm{N}} = \left(y_1^{\mathrm{N}}, \ldots, y_p^{\mathrm{N}}\right)$ which is the maximum of each objective over $\mathcal{Y}_N$. That is, $y_i^{\mathrm{N}} = \max\{y_i \mid \boldsymbol{y} \in \mathcal{Y}_N\}$ for $i = 1, \ldots, p$. However, as $\mathcal{Y}_N$ is unknown beforehand it is difficult to determine this when $p > 2$ [1]. A practical choice of $\boldsymbol{y}^{\mathrm{UB}}$ is usually determined by either using the expertise of a DM or by finding the pseudonadir point $\boldsymbol{y}^{\mathrm{pN}} = \left(y_1^{\mathrm{pN}}, \ldots, y_p^{\mathrm{pN}}\right)$, where $y_i^{\mathrm{pN}} = \max\left\{y_i^{\mathrm{A}_j} \mid j = 1, \ldots, p\right\}$ [15]. The drawback of these choices for $\boldsymbol{y}^{\mathrm{UB}}$ is that

they do not guarantee to find the complete $\mathcal{Y}_N$ front.

3 WSS sandwich algorithm

Section 3.1 describes the general steps of a WSS sandwich algorithm and Section 3.2 explains the decisions to be made within the algorithm.

3.1 General steps of a WSS sandwich algorithm

The general procedure of a WSS sandwich algorithm is:

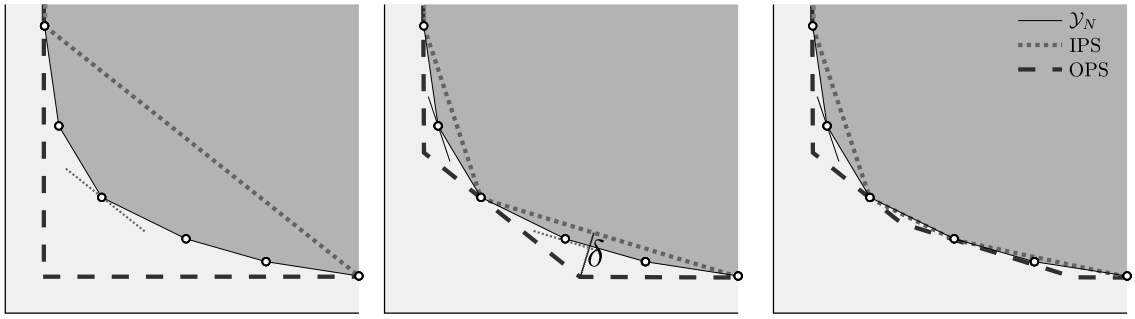
0. **Initialize IPS and OPS.**
   Find the anchor points $\boldsymbol{y}^{\mathrm{A}_i}, \ldots, \boldsymbol{y}^{\mathrm{A}_p}$ and obtain an upper bound value $\boldsymbol{y}^{\mathrm{UB}}$ to construct the convex hull of the IPS. That is, IPS $= \operatorname{conv}\left\{\boldsymbol{y}^{\mathrm{A}_i}, \ldots, \boldsymbol{y}^{\mathrm{A}_p}, \boldsymbol{y}^{\mathrm{UB}}\right\}$. The OPS is constructed as OPS $= \left\{\boldsymbol{y} \mid y_i^{\mathrm{A}_i} \leq y_i \leq y_i^{\mathrm{UB}}\right\}$. Obtain all relevant halfspaces that belong to the IPS.

1. **Expand the IPS based on its distance to the OPS.**
   For each relevant halfspace determine the distance to the OPS and find the halfspace $\mathcal{H}^{\geq}\left(\bar{\boldsymbol{w}}, \bar{b}\right)$ for which the current distance δ from the IPS to the OPS is maximal. Solve the weighted sum method using $\bar{\boldsymbol{w}}$, and obtain $\bar{\boldsymbol{y}} = \arg\min\{\bar{\boldsymbol{w}}^{\intercal}\boldsymbol{y} \mid \boldsymbol{y} \in \mathcal{Y}\}$.

2. **Update IPS and OPS.**
   Add $\bar{\boldsymbol{y}}$ to the IPS, such that IPS $= \operatorname{conv}\{\mathrm{IPS}, \bar{\boldsymbol{y}}\}$. Add $\bar{\boldsymbol{w}}^{\intercal}\boldsymbol{y} \geq \bar{\boldsymbol{w}}^{\intercal}\bar{\boldsymbol{y}}$ to the OPS. Update the relevant halfspaces.

3. **Check stopping criterion.**
   Repeat the above procedure from step 1 onwards until a stopping criterion is met (e.g. ε-distance between IPS and OPS, number of iterations or time).

In Figure 1 a few iterations of the WSS sandwich algorithm are shown for an example with $p = 2$, where the IPS and OPS are iteratively improved.



(a) (b) (c)

Figure 1: Illustration of the WSS sandwich algorithm. In (a) the initial approximations are based on the anchor points. In (b) the approximations are improved after one iteration. Here, δ denotes the maximum distance between the inner and the outer approximation. In (c) the approximations are further improved in the second iteration.

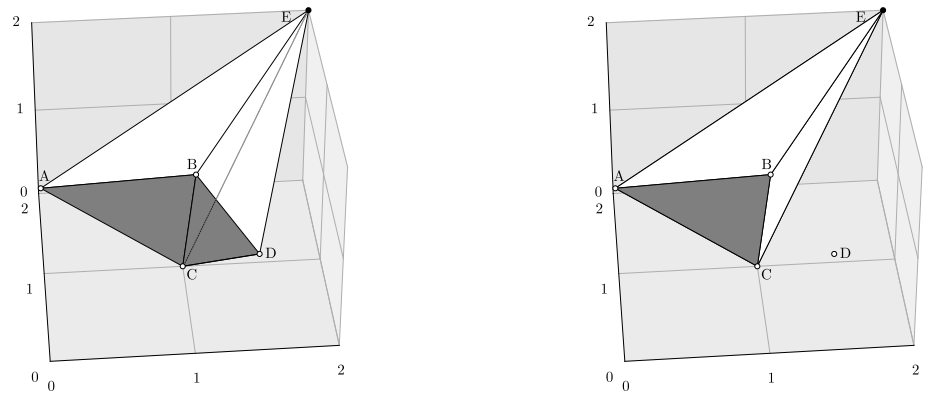
3.2 Decisions within WSS sandwich algorithms

In the general procedure there are several decisions that can impact the algorithm's performance. The most significant ones are how to deal with negative or mixed normals, i.e. halfspaces of the inner approximation that contain negative values, and how to determine the distance between the IPS and the OPS. In this section we discuss the current literature's view on these topics.

3.2.1 Handling mixed and negative normals

As indicated before, the inner normals of the inner approximation can contain negative values when $p \geq 3$. We refer to an inner normal containing some negative values as a mixed normal, and to an inner normal with solely negative values as a negative normal. Because of the efficiency property of the WSS, it holds that an inner normal containing only strictly positive elements will result in finding an efficient solution. As we do not have this guarantee for the mixed or negative normal, the question is: is such a normal still worth exploring? To answer this question, we provide an example, which is a simplification of an example in [5], to show the relevance of a normal that contains negative values.

Consider a small example with $p = 3$ objectives that has extreme points A $= (0, 1, 1)$, B $= (1, 0, 2)$, C $= (1, 1, 0)$, D $= (1.5, 0.55, 0.55)$ and E $= (2, 2, 2)$, where $\mathcal{Y} =$ conv {A,B,C,D,E}. This is represented in Figure 2a. The set of nondominated points $\mathcal{Y}_N$ is given by faces ABC and BCD. Note that of the extreme points only E is not nondominated. To initialize the IPS, we need at least $p + 1 = 4$ vertices to create a convex hull. We initialize our algorithm with the anchor points and our upper bound E. Thus, the current IPS is found by conv {A,B,C,E}, see Figure 2b. Here only the inner normal of face ABC is positive. Using this normal as weight vector will not generate a new efficient point. So, in case one only considers positive normals, vertex D will not be found.



(a) (b)

Figure 2: Mixed and negative normals help to find more nondominated points. Efficient faces are indicated with gray. In (a) $\mathcal{Y}$ is represented. In (b) the IPS is shown after initialization. Only using the single positive inner normal of face ABC does not result in finding efficient vertex D. The mixed normal of face BCE does find D.

Since each optimization is expensive, Craft et al. [6] do not optimize over mixed or negative normals to ensure that each optimization contributes to the IPS by finding an efficient point. Instead of disregarding those normals, they construct a new weight based

on the weights that were used to find the vertices of the corresponding face. In the example of Figure 2b this concerns face BCE. The goal is to still find efficient points that can not be found with the current set of positive normals, i.e. to find point D. The new weights are constructed such that they are maximally different from the weights corresponding to the vertices of the considered face, given that the new weights are positive and are a convex combination of the weights corresponding to the vertices of the considered face. In the initialization of Craft et al. [6], point E is found using the outer normal of face ABC. Combining this normal with the normals used to find B and C, gives the maximally different weight $\left[0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right]$. Optimizing the WSS with this weight results in point C. Despite the newly constructed weight, the approach is not able to find point D.

Solanki et al. [5] allow for the optimization over normals with negative values, as they advocate that mixed and negative normals can help to find efficient points and to improve the overall distance between the IPS and OPS as more points and therefore more normals are available. In our example, using the mixed normal of face BCE results in finding extreme point D. After this, the IPS is updated and face BCE is replaced by the faces BCD, BDE and CDE. Although using a mixed normal or negative normal as a weight does not necessarily result in finding a nondominated point, it indeed helps to find efficient points which are not found if they are disregarded or altered using the method of Craft et al. [6].

In conclusion, using mixed and negative normals as weights are helpful to find efficient points that are otherwise not found. Using them, however, might result in many more optimizations that might not find new efficient points.

3.2.2 Determining the next face

In step 1 of the general WSS sandwich algorithm all of the unexplored relevant faces are of interest. However, optimizing over all these weights simultaneously can result in an exponential growth of the number of faces. As some weights may tighten the gap between the IPS and $\mathcal{Y}_N$ more than others, it could be beneficial to find the face that has the best potential to improve the current IPS. Generally, the most promising face is the one that has the largest distance to $\mathcal{Y}_N$. As it is usually not possible to determine the distance from the IPS to $\mathcal{Y}_N$, the OPS can be used as a proxy for $\mathcal{Y}_N$. Then, using the weight of the most promising face, the IPS will be expanded and the OPS tightened maximally, making the gap smaller.

Solanki et al. [5] select the most promising face, and thus weight, by determining which halfspace of the faces of the IPS has the largest distance to the OPS. Using $\|\boldsymbol{w}\| = 1$, they define the distance for each halfspace $\mathcal{H}^{\geq}(\bar{\boldsymbol{w}}, \bar{b})$ of the IPS to the OPS as follows:

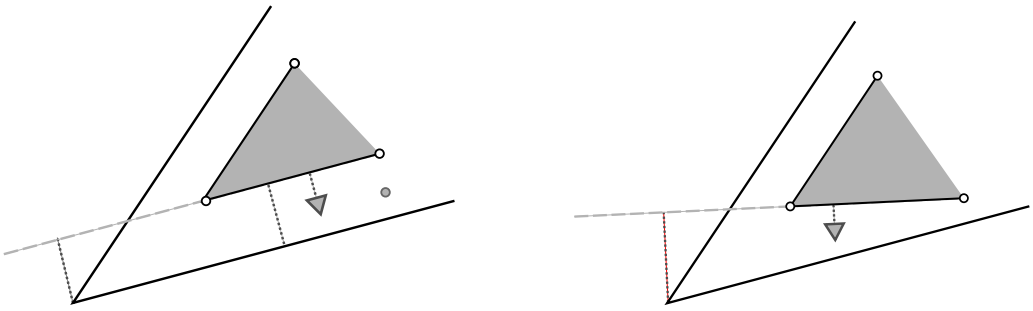
$$\delta(\bar{\boldsymbol{w}}, \bar{b}) = \bar{b} - \min\left\{\bar{\boldsymbol{w}}^{\intercal}\boldsymbol{y} \mid \boldsymbol{y} \in \text{OPS}\right\}, \tag{1}$$

where the OPS is a convex polyhedron by construction. Then, the inner normal of the halfspace for which the distance to the OPS is the largest, is used as weight in the next iteration. This distance is then easily determined by solving an LP for each halfspace of the IPS.

Based on the newly found point, the distance from the OPS to the IPS might change, even for faces of the former IPS which are still part of the updated IPS. Thus, in [5] the distance from each halfspace of the IPS to the OPS is (re)calculated in each iteration. In addition to an exponential growth in the number of distance calculations due to the growing number of faces, the size of the LPs to be solved will grow as well as in each iteration a constraint is added to the OPS.

To reduce the size of the problem, Craft et al. [6] determine the distance between a face of the IPS and the OPS based on previously used inner normals. That is, for each vertex of the IPS that lies on our halfspace there is a corresponding halfspace of the OPS. To obtain the distance between the IPS and the OPS, one needs to determine the maximum distance between a face of the IPS and the intersection point between the halfspaces corresponding to the vertices of that specific IPS face, which is a subset of the OPS. Craft et al. [6] do so by solving a system of inequalities using these halfspaces. However, there are many scenarios in which these halfspaces intersect non-uniquely, do not intersect at all or in which the intersection overestimates the distance measure introduced in [5]. For the latter we have provided an example in Appendix A. In case the halfspaces intersect non-uniquely or do not intersect, the closed form expression cannot be determined. To resolve this, an angle between the halfspaces is calculated corresponding to the vertices of the face of the IPS, such that the one with the smallest angle is chosen as the next facet [6]. Computing the angles makes the approach complex and computationally expensive.

A caveat to this definition of distance, not mentioned in [5] and [6], is that the maximum distance determined with (1) is not necessarily decreasing over the iterations. We conjecture that this effect does not occur in 2 dimensions, due to its geometrical properties. For ease of explanation, however, we show a small 2D example that shows why (1) is not exact. In Figure 3a the maximum distance of the IPS to the OPS, according to (1), is indicated with a dotted line from the face of the IPS. Expanding this face results in finding a new point, and therefore the IPS is expanded as well. Assuming that our relevant part of the OPS was not affected, the distance of the adjusted face to the OPS has decreased but the maximum distance of the halfspace to the OPS has increased. Note that in 2 dimensions the relevant part of the OPS will change, and thus this point would have been cut off, however in more than 2 dimensions this is not necessarily the case.



(a) (b)

Figure 3: Using the distance definition used in [5] the maximum distance can increase. In (a) the maximum found distance from the IPS to the OPS is indicated with a dotted line from the halfspace. The inner normal of this halfspace is used to find a new point. In (b) the found point expands the IPS, where the adjusted face is now closer to the OPS. This face corresponds to the halfspace which has the largest distance to the OPS, which increased compared to the maximum distance in (a).

4 Methodology

We propose an alternative modification of the dominated part of the criterion space, which we refer to as the modified criteron space (MCS). This means that only $\mathcal{Y} \setminus \mathcal{Y}_N$ is modified, whereas $\mathcal{Y}_N$ and $\mathcal{X}$ remain unchanged. In Section 4.1 we first explain the formulation of

Solanki et al. [5] and its drawbacks, and then we present our modification and explain how it differs from the former formulation. In Section 4.2 we suggest two alternative approaches to approximate the distance between the IPS and the OPS combining the ideas of Solanki et al. [5] and Craft et al. [6] to obtain simpler procedures that need fewer recalculations. Then, in Section 4.3 we present an algorithm that contains the above ideas about the MCS, where a distance approach can be incorporated. Finally, in Section 4.4 we show several properties of the MCS.

4.1 Modified criterion space (MCS)

Solanki et al. [5] modify the criterion space by including upper bounds on the objectives, which prevents ending up in an unbounded part of the criterion space while optimizing over a negative normal. The resulting convex MO is formulated as:

$$\begin{aligned} \min\ & \boldsymbol{w}^\intercal \boldsymbol{f}(\boldsymbol{x}) = \sum_{i=1}^{p} w_i f_i(\boldsymbol{x}) \\ \text{s.t. } & \boldsymbol{x} \in \mathcal{X} \\ & \boldsymbol{f}(\boldsymbol{x}) \leq \boldsymbol{y}^{\mathrm{UB}}. \end{aligned}$$

A drawback of this approach is that if the dominated part of the criterion space has a complicated description, many different dominated points will be found while searching with negative normals. That way, one describes the criterion space almost completely instead of focusing on the Pareto front. This is also apparent from the results of Rennen et al. [7][1], where the method of Solanki et al. [5] finds many dominated points.

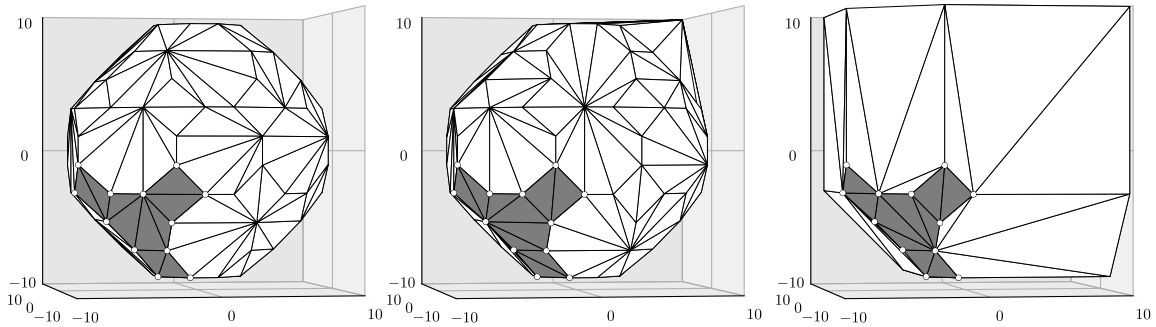
The algorithm spends a lot of computations on describing the dominated part particularly when the chosen upper bound $\boldsymbol{y}^{\mathrm{UB}}$ is larger than the nadir point. To show this we consider an MOLP with three objectives and $\varepsilon = 0$ based on an instance of `Bensolvehedron`[2], which is described in more detail in Section 5. Figure 4a shows the representation of $\mathcal{Y}$, where the gray area represents $\mathcal{Y}_N$. For this example it holds that the nadir point is $\boldsymbol{y}^{\mathrm{N}} = (-1, -1, -1)$.

Figure 4b shows the IPS found with the modification method of Solanki et al. [5] with $\boldsymbol{y}^{\mathrm{UB}} = (10, 10, 10)$, which is larger than the nadir point $\boldsymbol{y}^{\mathrm{N}}$. Although the method is able to find the complete $\mathcal{Y}_N$ front, it also describes a large part of $\mathcal{Y} \setminus \mathcal{Y}_N$. As we are only interested in finding $\mathcal{Y}_N$, many uninformative LPs are solved.

In our suggested modification we still want to use mixed and negative normals as they help to construct the IPS and OPS and ensure that we find the complete $\mathcal{Y}_N$ front, but we want to reduce unnecessary searches towards the dominated part. Therefore, we modify the criterion space by extending and flattening the dominated part of $\mathcal{Y}$ by means of conification. That is, we end up with $\left(\mathcal{Y} + \mathbb{R}^p_{++}\right) \bigcap \left\{\boldsymbol{y} \mid y_i \leq y_i^{\mathrm{UB}}\right\}$, which has the same Pareto front as $\mathcal{Y}$. This conification can be incorporated within the WSS by formulating the problem as follows:

[1] Here, we refer to figure 15 in [7].

[2] Instance is $p = 3$ and $m = 0$.



(a) Representation of $\mathcal{Y}$. (b) Modified $\mathcal{Y}$ with [5]. (c) Modified $\mathcal{Y}$ with MCS.

Figure 4: Complete and modified descriptions of $\mathcal{Y}$. The gray area represents $\mathcal{Y}_N$ with $\boldsymbol{y}^{\mathrm{N}} = (-1, -1, -1)$. The modified descriptions use $\boldsymbol{y}^{\mathrm{UB}} = (10, 10, 10)$.

$$
\begin{aligned}
\text{(MCS)} \quad & \min \boldsymbol{w}^{\intercal} \left(\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{u} \right) = \sum_{i=1}^{p} w_i \left(f_i(\boldsymbol{x}) + u_i \right) \\
\text{s.t.} \quad & \boldsymbol{x} \in \mathcal{X} \\
& \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{u} \leq \boldsymbol{y}^{\mathrm{UB}} \\
& \boldsymbol{u} \geq \boldsymbol{0},
\end{aligned}
$$

where $\boldsymbol{u}$ is a $p \times 1$ vector containing additional decision variables. Here, u_i measures the maximum allowed deviation of objective i. As previously mentioned, we refer to this formulation as the MCS. Note that the two formulations are similar in size, and will therefore have similar performance for a given weight.

Figure 4c illustrates the approximated area of $\mathcal{Y}$ of our previous example using the MCS with the same $\boldsymbol{y}^{\mathrm{UB}}$. In this case the description of the dominated part is less detailed than for the modification method of Solanki et al. [5], whereas the complete $\mathcal{Y}_N$ front is still obtained. In this case fewer uninformative LPs are solved.

4.2 Selecting the next face

As explained in Section 3.2, a distance measure could help to guide the algorithm towards selecting IPS faces, and thus weights, that have the largest potential to decrease the gap between the IPS and the OPS.

In this section we continue to work with the definition of distance as used in [5] and [6], as the concept is well-explainable and comprehensible to DMs, and the distance is solved with a relatively simple LP. For readability we refer to the distance calculation in [5], where in each iteration the distance to the OPS for each halfspace of the IPS is (re)calculated, as SOL.

We propose two alternatives to determine the maximum distance, inspired by the approach of Craft et al. [6]. For the first alternative, the distance is calculated using a relevant subset of the OPS inequalities instead of all OPS inequalities. Only when the relevant subset of a face is altered based on a newly found point, the distance is recalculated. The second alternative uses the current set of OPS inequalities to determine the distance of new halfspaces of the IPS. Once the distance is calculated, it is not recalculated even when the OPS changes. Both alternatives focus in varying degrees on the reduction of the size of the

LP in terms of number of constraints as well as the number of LPs to be solved. We refer to the first alternative and the second alternative as SUB and CUR, respectively.

The first alternative, SUB, is formally described as follows: let $\boldsymbol{w}$ be a weight and let $\boldsymbol{v}$ be the corresponding found point in $\mathcal{Y}$ with objective value β. That is,

$$\beta = \min \left\{ \boldsymbol{w}^\intercal \left(\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{u} \right) \mid \boldsymbol{x} \in \mathcal{X},\, \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{u} \leq \boldsymbol{y}^{\mathrm{UB}},\, \boldsymbol{u} \geq \boldsymbol{0} \right\}.$$

Then, the OPS inequality corresponding to this weight is $\boldsymbol{w}^\intercal \boldsymbol{y} \geq \beta$. Now, let $\mathcal{H}^{\geq}(\boldsymbol{w}, b)$ be the halfspace of the IPS for which we want to determine the distance to the OPS and let $V = \left\{ \boldsymbol{v}^1, \ldots, \boldsymbol{v}^k \right\}$ contain the vertices of the IPS that are on the hyperplane $\mathcal{H}(\boldsymbol{w}, b)$. As just described, these vertices were found before using WSS where $\boldsymbol{w}^1, \ldots, \boldsymbol{w}^k$ are the weights used to find these vertices and $\beta_1, \ldots, \beta_k$ the respective objective values. Then, the considered halfspaces of the OPS — in line with the idea of Craft et al. [6] — are $\left\{ \left(\boldsymbol{w}^i \right)^\intercal \boldsymbol{y} \geq \beta_i \mid \boldsymbol{v}^i \in V \right\}$. Additionally, the initial bounding box of the OPS is included to reduce the overestimation of the distance as explained in Appendix A. Together these sets form a subset of halfspaces of the OPS and they are used to solve (1).

Using this approach, we only need to recalculate the distance in case the vertices of the face have changed, i.e. when new vertices are part of the same halfspace. This is different from SOL where in each iteration of the WSS all distances have to be recalculated. Furthermore, as we use a subset of the OPS halfspaces, the LPs to be solved will consist of a relatively constant number of inequalities in comparison to the size of the OPS, which grows in each iteration.

As briefly explained, the second alternative, CUR, determines the distance only for halfspaces of the IPS which have not been considered before and for which the distance has not yet been determined. So, in case the distance for $\mathcal{H}^{\geq}(\boldsymbol{w}, b)$ has not yet been determined, CUR uses the current OPS to solve the LP corresponding to (1).

For CUR it holds that the size of the solved LP will be larger than for SUB and equal to that of SOL. Its main distinction is that it does not apply any recalculation as SOL and SUB do. Note that both approaches can overestimate the distance definition of SOL, as they do not take the complete OPS into account. Table 1 provides a brief overview of the different features of the three different distance measures.

Table 1: Differences in features of the three distance approaches.

	SUB	CUR	SOL
Recalculation	Some faces	No	All faces
Nr. of OPS inequalities used	Stable	Increases	Increases
Dist. relative to (1)	Overestimation	Overestimation	Exact

Abbreviations: nr. for number and dist. for distance.

4.3 MCS algorithm

Algorithm 1 describes how to obtain the MCS to ε-precision. Note that the numbers of the provided comments correspond to the general steps of a WSS sandwich algorithm as explained in Section 3.1. In line 11 the halfspace is determined which has the largest distance to the OPS. Here, one could use SOL, SUB, CUR or a different distance measure depending on the preference of the user. We refer to Appendix B for implementation choices and details of the algorithm.

Algorithm 1 Modified criterion space (MCS)

Result: Obtain the modified criterion space, and thus $\mathcal{Y}_N$, to ε-precision.
Input: $\boldsymbol{y}^{\mathrm{UB}}$, ε

▷ `/* 0. Initialize IPS and OPS. /*`
IPS $\leftarrow \left\{\boldsymbol{y}^{\mathrm{UB}}\right\}$
OPS $\leftarrow \left\{\left(\boldsymbol{e}^1\right)^\intercal \boldsymbol{y} \leq y_1^{\mathrm{UB}}, \ldots, \left(\boldsymbol{e}^p\right)^\intercal \boldsymbol{y} \leq y_p^{\mathrm{UB}}\right\}$
for $i \leftarrow 1$ to p **do**
  $\tilde{\boldsymbol{x}} \leftarrow \arg\min \{f_i(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{X}\}$
  $\tilde{\beta} \leftarrow f_i(\tilde{\boldsymbol{x}})$
  Add $\left(\boldsymbol{e}^i\right)^\intercal \boldsymbol{y} \geq \tilde{\beta}$ to OPS
  IPS $\leftarrow \operatorname{conv}\{\text{IPS}, \boldsymbol{f}(\tilde{\boldsymbol{x}})\}$
end for
$H \leftarrow$ Set of all halfspaces of IPS
while $H \neq \emptyset$ **do**
▷ `/*1. Expand the IPS based on its distance to the OPS./*`
  Find halfspace $\mathcal{H}^{\leq}(\boldsymbol{w}, b) \in H$ of IPS with largest distance δ to the OPS
  ▷ `/*3. Check stopping criterion./*`
  if $\delta > \varepsilon$ **then**
    $\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}} \leftarrow \arg\min \left\{\boldsymbol{w}^\intercal \left(\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{u}\right) \mid \boldsymbol{x} \in \mathcal{X},\ \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{u} \leq \boldsymbol{y}^{\mathrm{UB}},\ \boldsymbol{u} \geq \boldsymbol{0}\right\}$
    $\bar{\beta} \leftarrow \boldsymbol{w}^\intercal \left(\boldsymbol{f}(\bar{\boldsymbol{x}}) + \bar{\boldsymbol{u}}\right)$
    ▷ `/*2. Update IPS and OPS/*`
    Add $\boldsymbol{w}^\intercal \boldsymbol{y} \geq \bar{\beta}$ to OPS
    IPS $\leftarrow \operatorname{conv}\{\text{IPS}, \boldsymbol{f}(\bar{\boldsymbol{x}})\}$
  else
    break
  end if
  $H \leftarrow$ Set of all halfspaces of IPS not visited before
end while
return IPS

4.4 Properties of WSS sandwich and MCS

In this section we prove that under certain conditions a WSS sandwich algorithm is able to describe all Pareto points of an MOLP in a finite number of iterations. Furthermore, we show that for the MCS, the maximum number of WSS LPs of the MOLP is not affected by $\boldsymbol{y}^{\mathrm{UB}}$, as long as $\boldsymbol{y}^{\mathrm{UB}} > \boldsymbol{y}^{\mathrm{N}}$. Finally, for the MCS we provide an upper bound on the number of WSS LPs needed to find all efficient points of an MOLP.

4.4.1 WSS sandwich can find all Pareto points for MOLP

Although not proven in [5] and, to the best of our knowledge, not proven in the literature so far, one can prove that all Pareto points of an MOLP can be found in a finite number of expansions of the IPS using WSS in a sandwich algorithm. For this the considered criterion space has to be polyhedral, bounded and contain all Pareto points where $\mathcal{Y}_N$ is bounded. In case the considered criterion space is unbounded, one can consider a modification method such as the MCS.

In order to prove this statement, we first need some additional notation. Let $\mathcal{Y}$ denote the original criterion space for which $\mathcal{Y}_N$ is bounded and let $\mathcal{Y}_{MCS}$ denote the modified criterion space. For simplicity, we use the MCS to ensure that $\mathcal{Y}_{MCS}$ is bounded and

polyhedral by construction, that is, $\mathcal{Y}_{MCS} = \left(\mathcal{Y} + \mathbb{R}^p_{++}\right) \bigcap \left\{\boldsymbol{y} \mid y_i \leq y_i^{\mathrm{UB}}\right\}$. Here, we assume that $\boldsymbol{y}^{\mathrm{UB}} \geq \boldsymbol{y}^{\mathrm{N}}$ such that $\mathcal{Y}_N \subseteq \mathcal{Y}_{MCS}$. Note that it is also possible to use the method of [5] to obtain a criterion space that adheres to these assumptions. Then, let $V = \{\boldsymbol{v}^1, \ldots, \boldsymbol{v}^q\}$ be the set of q vertices of $\mathcal{Y}_{MCS}$. As $\mathcal{Y}_{MCS}$ is convex, each point $\boldsymbol{y} \in \mathcal{Y}_{MCS}$ can be written as a convex combination of the vertices in V. That is, $\boldsymbol{y} = a_1\boldsymbol{v}^1 + \cdots + a_q\boldsymbol{v}^q$ where $a_1, \ldots, a_q$ are the coefficients of this combination. Here, we let $V^+(\boldsymbol{a}) \subseteq V$ denote the set of vertices that have a strictly positive coefficient for this particular convex combination. Finally, let IPS denote the current IPS such that $\mathrm{IPS} = \{\boldsymbol{W}\boldsymbol{y} \geq \boldsymbol{d} \mid \boldsymbol{y} \in \mathcal{Y}\}$, where $\boldsymbol{W} = (\boldsymbol{w}^1, \ldots, \boldsymbol{w}^l)$ consists of l inner normals and $\boldsymbol{d} = (d_1, \ldots, d_l)$ contains the corresponding offsets.

Lemma 4.1. *IPS $\neq \mathcal{Y}_{MCS} \iff$ there exists an inner normal of the IPS that will expand the IPS.*

Proof. If IPS $\neq \mathcal{Y}_{MCS}$, there exists at least one point $\boldsymbol{y} \in \mathcal{Y}_{MCS}$ such that $\boldsymbol{y} \notin$ IPS. For this point it must hold that $\exists i$ such that $\left(\boldsymbol{w}^i\right)^\intercal \boldsymbol{y} < d_i$. Thus, using weight $\boldsymbol{w}^i$ in the WSS will result in finding a new point not yet in IPS, hence expanding the IPS. The reverse of the proof is trivial. □

Note that in Lemma 4.1 we do not necessarily find $\boldsymbol{y}$ as there might be other points such as $\boldsymbol{y}' \in \mathcal{Y}_{MCS}$ for which $\left(\boldsymbol{w}^i\right)^\intercal \boldsymbol{y}' \leq \left(\boldsymbol{w}^i\right)^\intercal \boldsymbol{y}$.

Lemma 4.2. *If $\boldsymbol{w}^i$ is a weight that expands the current IPS where $\boldsymbol{y} = a_1\boldsymbol{v}^1 + \cdots + a_q\boldsymbol{v}^q$ is its corresponding found point, then this specific combination of vertices $\boldsymbol{v} \in V^+(\boldsymbol{a})$ has not been considered before while expanding the IPS.*

Proof. As the current IPS can be expanded, according to Lemma 4.1, it holds that IPS $\neq \mathcal{Y}_{MCS}$ and that $\left(\boldsymbol{w}^i\right)^\intercal \boldsymbol{y} = \beta < d_i$. For all $\boldsymbol{v} \in V^+(\boldsymbol{a})$ it also holds that $\left(\boldsymbol{w}^i\right)^\intercal \boldsymbol{v} = \beta$. That is, $\left(\boldsymbol{w}^i\right)^\intercal \boldsymbol{v} \geq \beta$ as $\boldsymbol{y}$ is optimal for (MCS). Combining the previous statement with $\left(\boldsymbol{w}^i\right)^\intercal \boldsymbol{y} = \left(\boldsymbol{w}^i\right)^\intercal \left(a_1\boldsymbol{v}^1 + \cdots + a_q\boldsymbol{v}^q\right) = \beta$, it must hold that $\left(\boldsymbol{w}^i\right)^\intercal \boldsymbol{v} = \beta$. As a result, it follows that all $\boldsymbol{v} \in V^+(\boldsymbol{a})$ lie on the same hyperplane $\mathcal{H}(\boldsymbol{w}^i, \beta)$.

In order to show that this combination of vertices has not been considered before, let us assume per contradiction that this specific combination of vertices $\boldsymbol{v} \in V^+(\boldsymbol{a})$ has been considered before to expand the IPS. Then it must hold for a point $\hat{\boldsymbol{y}} = \hat{a}_1\boldsymbol{v}^1 + \cdots + \hat{a}_q\boldsymbol{v}^q \in$ IPS to be a convex combination of exactly these vertices in $V^+(\boldsymbol{a})$. Thus,

$$\begin{aligned}
\left(\boldsymbol{w}^i\right)^\intercal \hat{\boldsymbol{y}} &= \left(\boldsymbol{w}^i\right)^\intercal \sum_{\boldsymbol{v} \in V^+(\boldsymbol{a})} \hat{a}_{\boldsymbol{v}}\, \boldsymbol{v} \\
&= \sum_{\boldsymbol{v} \in V^+(\boldsymbol{a})} \hat{a}_{\boldsymbol{v}}\, \left(\boldsymbol{w}^i\right)^\intercal \boldsymbol{v} \\
&= \sum_{\boldsymbol{v} \in V^+(\boldsymbol{a})} \hat{a}_{\boldsymbol{v}}\, \beta \\
&= \beta \sum_{\boldsymbol{v} \in V^+(\boldsymbol{a})} \hat{a}_{\boldsymbol{v}} \\
&= \beta,
\end{aligned}$$

where $\hat{a}_{\boldsymbol{v}}$ is the positive coefficient of $\boldsymbol{v}$. This is a contradiction since $\hat{\boldsymbol{y}} \in$ IPS implies that $\left(\boldsymbol{w}^i\right)^\intercal \hat{\boldsymbol{y}} \geq d_i > \beta$.

Thus, it must hold that this specific combination of vertices $\boldsymbol{v} \in V^+(\boldsymbol{a})$ has not been considered before when expanding the IPS. □

Theorem 4.1. *The WSS sandwich algorithm can find all Pareto efficient points of an MOLP in a finite number of expansions of the IPS.*

Proof. From Lemma 4.1 we conclude that the WSS method will always expand the IPS if it does not yet contain $\mathcal{Y}_{MCS}$ and per extension $\mathcal{Y}_N$. Furthermore, from Lemma 4.2 we can conclude that as $\mathcal{Y}_{MCS}$ has a finite number of vertices q there are a finite number of combinations 2^q of positive coefficients for the vertices. Therefore, we conclude that there are only a finite number of expansions of the IPS possible before it finds $\mathcal{Y}_{MCS}$ and thus $\mathcal{Y}_N$. □

4.4.2 Upper bound y^{UB} does not affect maximum number of WSS LPs of the MCS for MOLP

In this section we show that the MCS formulation for the MOLP is not impacted by the choice of $\boldsymbol{y}^{\mathrm{UB}}$ if $\boldsymbol{y}^{\mathrm{UB}} > \boldsymbol{y}^{\mathrm{N}}$. That is, the number of found vertices will be constant to obtain $\mathcal{Y}_N$[3]. In order to prove this we first have to rewrite $\mathcal{Y}_{MCS}$ into another expression and prove some properties of the number of vertices of a polytope. Then, we show that the number of vertices of the MCS for the MOLP stays constant as long as $\boldsymbol{y}^{\mathrm{UB}} > \boldsymbol{y}^{\mathrm{N}}$.

Define $\mathcal{Y}_M(\tilde{\boldsymbol{y}}) \equiv \left(\mathcal{Y} + \mathbb{R}^p_{++}\right) \bigcap \{\boldsymbol{y} \mid y_i \leq \tilde{y}_i\}$. Note that $\mathcal{Y}_{MCS} = \mathcal{Y}_M\left(\boldsymbol{y}^{\mathrm{UB}}\right)$, in line with Section 4.1. Also note that $\mathcal{Y}_M(\tilde{\boldsymbol{y}})$ is a bounded convex polyhedron if $\mathcal{Y}$ is a convex polyhedron bounded from below.

Lemma 4.3. $\boldsymbol{y} \in \mathcal{Y}_M(\tilde{\boldsymbol{y}}) \Longleftrightarrow \boldsymbol{y} = \boldsymbol{z} + \boldsymbol{u}$ *where* $\boldsymbol{z} \in \mathcal{Y}_N$, $\boldsymbol{u} \geq \boldsymbol{0}$ *and* $\boldsymbol{z} + \boldsymbol{u} \leq \tilde{\boldsymbol{y}}$.

Proof. By the definition of the Minkowski sum and set intersection, we get $\boldsymbol{y} \in \mathcal{Y}_M(\tilde{\boldsymbol{y}}) \iff \boldsymbol{y} = \boldsymbol{z}' + \boldsymbol{u}'$ with $\boldsymbol{z}' \in \mathcal{Y}$, $\boldsymbol{u}' \geq \boldsymbol{0}$ and $\boldsymbol{z}' + \boldsymbol{u}' \leq \tilde{\boldsymbol{y}}$. However, note that $\boldsymbol{z}' = \boldsymbol{z} + \boldsymbol{u}''$ with $\boldsymbol{z} \in \mathcal{Y}_N$ and $\boldsymbol{u}'' \geq \boldsymbol{0}$. Thus, $\boldsymbol{y} = \boldsymbol{z} + (\boldsymbol{u}' + \boldsymbol{u}'') = \boldsymbol{z} + \boldsymbol{u}$ with $\boldsymbol{u} \geq \boldsymbol{0}$. □

Note that Lemma 4.3 immediately implies $\mathcal{Y}_M(\tilde{\boldsymbol{y}}) = \left(\mathcal{Y}_N + \mathbb{R}^p_{++}\right) \bigcap \{\boldsymbol{y} \mid y_i \leq \tilde{y}_i\}$.

Theorem 4.2. *If* $\boldsymbol{y}^{UB} \geq \boldsymbol{y}^{N}$, *then* $\mathcal{Y}_M\left(\boldsymbol{y}^{UB}\right) = \mathcal{Y}_M\left(\boldsymbol{y}^{N}\right) + \bigoplus_{i=1}^{p} conv\left\{\boldsymbol{0}, \boldsymbol{e}^i\right\}\left(y_i^{UB} - y_i^{N}\right)$.

Proof. Define $\mathcal{Y}_C \equiv \mathcal{Y}_M\left(\boldsymbol{y}^{\mathrm{N}}\right) + \bigoplus_{i=1}^{p} \operatorname{conv}\left\{\boldsymbol{0}, \boldsymbol{e}^i\right\}\left(y_i^{\mathrm{UB}} - y_i^{\mathrm{N}}\right)$. We first show that $\mathcal{Y}_M(\boldsymbol{y}^{\mathrm{UB}}) \subseteq \mathcal{Y}_C$. Take $\boldsymbol{y} \in \mathcal{Y}_M\left(\boldsymbol{y}^{\mathrm{UB}}\right)$, then according to Lemma 4.3 this can be formulated as $\boldsymbol{y} = \boldsymbol{z} + \boldsymbol{u}$ with $\boldsymbol{z} \in \mathcal{Y}_N$, $\boldsymbol{u} \geq \boldsymbol{0}$ and $\boldsymbol{z} + \boldsymbol{u} \leq \boldsymbol{y}^{\mathrm{UB}}$. Now, we rewrite this to

$$z_i + u_i = z_i + \min\left\{u_i, y_i^{\mathrm{N}} - z_i\right\} + \max\left\{z_i + u_i - y_i^{\mathrm{N}}, 0\right\} = z_i + \alpha_i + \beta_i,$$

where $\alpha_i = \min\left\{u_i, y_i^{\mathrm{N}} - z_i\right\}$ and $\beta_i = \max\left\{z_i + u_i - y_i^{\mathrm{N}}, 0\right\}$. Thus,

$$\boldsymbol{z} + \boldsymbol{u} = \boldsymbol{z} + \boldsymbol{\alpha} + \boldsymbol{\beta} = \boldsymbol{z} + \boldsymbol{\alpha} + \boldsymbol{e}^1\,\beta_1 + \cdots + \boldsymbol{e}^p\,\beta_p,$$

for which we argue that $\boldsymbol{z} + \boldsymbol{\alpha} \in \mathcal{Y}_M\left(\boldsymbol{y}^{\mathrm{N}}\right)$ and $\boldsymbol{e}^i\,\beta_i \in \operatorname{conv}\{\boldsymbol{0}, \boldsymbol{e}_i\}\left(y_i^{\mathrm{UB}} - y_i^{\mathrm{N}}\right)$ and thereby linking it to $\mathcal{Y}_C$. To show this, we list all relevant properties:

1. $\boldsymbol{z} \in \mathcal{Y}_N$,
2. $\boldsymbol{\alpha} \geq \boldsymbol{0}$ as $u_i \geq 0$ and $y_i^{\mathrm{N}} - z_i \geq 0$,
3. $\boldsymbol{z} + \boldsymbol{\alpha} \leq \boldsymbol{y}^{\mathrm{N}}$ as $z_i + \alpha_i \leq z_i + y_i^N - z_i = y_i^N$
4. $\boldsymbol{e}^i\,\beta_i \in \operatorname{conv}\{\boldsymbol{0}, \boldsymbol{e}_i\}\left(y_i^{\mathrm{UB}} - y_i^{\mathrm{N}}\right)$, since $\beta_i \in \left[0, y_i^{\mathrm{UB}} - y_i^N\right]$ and $y_i^{\mathrm{UB}} \geq y_i^{\mathrm{N}}$.

[3] Ignoring numerical precision.

It follows from 1. to 3. that $\boldsymbol{z}+\boldsymbol{\alpha} \in \mathcal{Y}_M\left(\boldsymbol{y}^{\mathrm{N}}\right)$ by Lemma 4.3. By the definition of Minkowski sum and 4., we have $\boldsymbol{y} = \boldsymbol{z} + \boldsymbol{\alpha} + \boldsymbol{\beta} \in \mathcal{Y}_C$.

Now, we show that $\mathcal{Y}_C \subseteq \mathcal{Y}_M(\boldsymbol{y}^{\mathrm{UB}})$. Take $\boldsymbol{y} \in \mathcal{Y}_C$, then this can be rewritten to $\boldsymbol{y} = \boldsymbol{z} + \boldsymbol{\alpha} + \boldsymbol{e}^1\,\beta_1 + \cdots + \boldsymbol{e}^p\,\beta_p$, with $\boldsymbol{z} \in \mathcal{Y}_N$, $\boldsymbol{\alpha} \geq \boldsymbol{0}$, $\boldsymbol{z} + \boldsymbol{\alpha} \leq \boldsymbol{y}^{\mathrm{N}}$, by Lemma 4.3, and $\beta_i \in \left[0, y_i^{\mathrm{UB}} - y_i^{\mathrm{N}}\right]$. Define

$$\boldsymbol{u} = \boldsymbol{\alpha} + \boldsymbol{e}^1\,\beta_1 + \cdots + \boldsymbol{e}^p\,\beta_p$$

and note that

$$z_i + u_i = z_i + \alpha_i + \beta_i \leq y_i^{\mathrm{N}} + y_i^{\mathrm{UB}} - y_i^{\mathrm{N}} = y_i^{\mathrm{UB}}$$

together with $\boldsymbol{u} \geq \boldsymbol{0}$, we conclude that $\boldsymbol{y} = \boldsymbol{z} + \boldsymbol{u} \in \mathcal{Y}_M\left(\boldsymbol{y}^{\mathrm{UB}}\right)$ by Lemma 4.3. □

The idea behind the above proof is visualized in Figure 5.

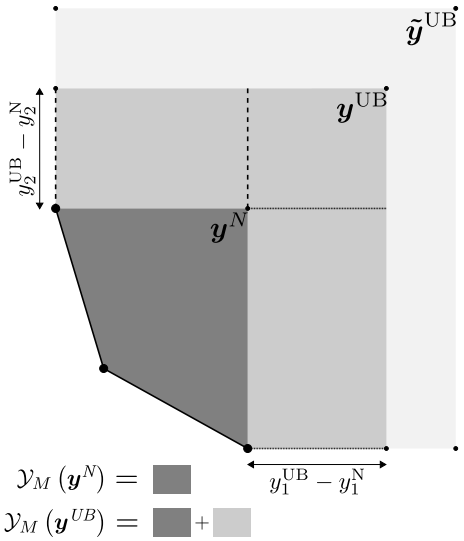


Figure 5: Example with $p = 2$ for Theorem 4.3, where $\mathcal{Y}_M\left(\boldsymbol{y}^{\mathrm{UB}}\right)$ is the Minkowski sum of $\mathcal{Y}_M\left(\boldsymbol{y}^{\mathrm{N}}\right)$ and $\bigoplus_{i=1}^{2} \operatorname{conv}\left\{\boldsymbol{0}, \boldsymbol{e}^i\right\}\left(y_i^{\mathrm{UB}} - y_i^{\mathrm{N}}\right)$.

Definition 4.1 ([16])**.** *Let P be a convex polyhedron, then $\boldsymbol{x} \in P$ is a vertex $\Longleftrightarrow$ $\exists \boldsymbol{\gamma}$ s.t. $\boldsymbol{\gamma}^\intercal \boldsymbol{x} < \boldsymbol{\gamma}^\intercal \tilde{\boldsymbol{x}}$ for all other points $\tilde{\boldsymbol{x}} \in P$.*

Let A be a bounded convex polyhedron and let $V(A)$ denote the vertices of A, that is $\{\boldsymbol{v}^1, \ldots, \boldsymbol{v}^k\}$. Define $B(a) = A + \operatorname{conv}\{\boldsymbol{0}, a\,\boldsymbol{u}\}$ for some fixed vector $\boldsymbol{u}$.

Theorem 4.3. *$|V(B(a))| = |V(B(b))|$ for all $a, b > 0$.*

Proof. Fix some $a, b > 0$. By the definition of Minkowski sum, the only possible vertices are $\boldsymbol{v}^i$ and $\boldsymbol{v}^i + a\,\boldsymbol{u}$ with $i = 1, \ldots, k$ for $B(a)$ and $\boldsymbol{v}^i$ and $\boldsymbol{v}^i + b\,\boldsymbol{u}$ with $i = 1, \ldots, k$ for $B(b)$. We now show that every vertex of $B(a)$ corresponds to a vertex of $B(b)$ and vice versa.

1. Assume $\boldsymbol{v}^i$ is a vertex of $B(a)$. Then, using Definition 4.1 we get

$$\exists \boldsymbol{\gamma} \text{ s.t. } \boldsymbol{\gamma}^\intercal \boldsymbol{v}^i < \boldsymbol{\gamma}^\intercal \boldsymbol{v}^j \qquad \forall j \neq i, \tag{2}$$

$$\boldsymbol{\gamma}^\intercal \boldsymbol{v}^i < \boldsymbol{\gamma}^\intercal \left(\boldsymbol{v}^j + a\,\boldsymbol{u}\right) \qquad \forall j. \tag{3}$$

Setting $j = i$ we derive from (3)

$$\boldsymbol{\gamma}^\intercal \boldsymbol{u} > 0, \tag{4}$$

as $a > 0$. Thus, we obtain the following properties, aligning with Definition 4.1,

$$\begin{aligned}
&\boldsymbol{\gamma}^\intercal \boldsymbol{v}^i < \boldsymbol{\gamma}^\intercal \boldsymbol{v}^j && \forall j \neq i \quad \text{use (2)},\\
&\boldsymbol{\gamma}^\intercal \boldsymbol{v}^i < \boldsymbol{\gamma}^\intercal \boldsymbol{v}^j < \boldsymbol{\gamma}^\intercal \boldsymbol{v}^j + b\,\boldsymbol{\gamma}^\intercal \boldsymbol{u} = \boldsymbol{\gamma}^\intercal \left(\boldsymbol{v}^j + b\,\boldsymbol{u}\right) && \forall j \neq i \quad \text{use (2) and (4)},\\
&\boldsymbol{\gamma}^\intercal \boldsymbol{v}^i < \boldsymbol{\gamma}^\intercal \boldsymbol{v}^i + b\,\boldsymbol{\gamma}^\intercal \boldsymbol{u} = \boldsymbol{\gamma}^\intercal \left(\boldsymbol{v}^i + b\,\boldsymbol{u}\right) && \quad \text{use (4)},
\end{aligned}$$

from which we conclude that $\boldsymbol{v}^i$ is also a vertex of $B(b)$.

2. Assume $\boldsymbol{v}^i + a\,\boldsymbol{u}$ is a vertex of $B(a)$. Then, using Definition 4.1 we get

$$\exists \boldsymbol{\gamma} \text{ s.t. } \boldsymbol{\gamma}^\intercal \left(\boldsymbol{v}^i + a\,\boldsymbol{u}\right) < \boldsymbol{\gamma}^\intercal \boldsymbol{v}^j \qquad \forall j, \tag{5}$$

$$\boldsymbol{\gamma}^\intercal \left(\boldsymbol{v}^i + a\,\boldsymbol{u}\right) < \boldsymbol{\gamma}^\intercal \left(\boldsymbol{v}^j + a\,\boldsymbol{u}\right) \qquad \forall j \neq i. \tag{6}$$

From (5) if we set $j = i$ we derive

$$\boldsymbol{\gamma}^\intercal \boldsymbol{u} < 0, \tag{7}$$

as $a > 0$ and from (6) we derive

$$\boldsymbol{\gamma}^\intercal \boldsymbol{v}^i < \boldsymbol{\gamma}^\intercal \boldsymbol{v}^j \quad \forall j \neq i. \tag{8}$$

Thus, we obtain the following properties, aligning with Definition 4.1,

$$\begin{aligned}
\boldsymbol{\gamma}^\intercal \left(\boldsymbol{v}^i + b\,\boldsymbol{u}\right) &= \boldsymbol{\gamma}^\intercal \boldsymbol{v}^i + b\,\boldsymbol{\gamma}^\intercal \boldsymbol{u} && &&\\
&< \boldsymbol{\gamma}^\intercal \boldsymbol{v}^i && &&\text{use (7)},\\
\boldsymbol{\gamma}^\intercal \left(\boldsymbol{v}^i + b\,\boldsymbol{u}\right) &= \boldsymbol{\gamma}^\intercal \boldsymbol{v}^i + b\,\boldsymbol{\gamma}^\intercal \boldsymbol{u} && &&\\
&< \boldsymbol{\gamma}^\intercal \boldsymbol{v}^i < \boldsymbol{\gamma}^\intercal \boldsymbol{v}^j && \forall j \neq i &&\text{use (7) and (8)},\\
\boldsymbol{\gamma}^\intercal \left(\boldsymbol{v}^i + b\,\boldsymbol{u}\right) &= \boldsymbol{\gamma}^\intercal \boldsymbol{v}^i + b\,\boldsymbol{\gamma}^\intercal \boldsymbol{u} && &&\\
&< \boldsymbol{\gamma}^\intercal \boldsymbol{v}^j + b\,\boldsymbol{\gamma}^\intercal \boldsymbol{u} = \boldsymbol{\gamma}^\intercal \left(\boldsymbol{v}^j + b\,\boldsymbol{u}\right) && \forall j \neq i &&\text{use (8)},
\end{aligned}$$

from which we conclude that $\boldsymbol{v}^i + b\,\boldsymbol{u}$ is a vertex of $B(b)$.

From 1. and 2. it follows that $|V(B(a))| \leq |V(B(b))|$. By swapping a and b and reapplying the above reasoning, we obtain $|V(B(b))| \leq |V(B(a))|$ and thus we find $|V(B(a))| = |V(B(b))|$. □

Combining the above Theorems we can now prove that the number of vertices of the MCS stays constant as long as $\boldsymbol{y}^{\mathrm{UB}} > \boldsymbol{y}^{\mathrm{N}}$.

Theorem 4.4. *If $\boldsymbol{y}^{UB}$, $\tilde{\boldsymbol{y}}^{UB} > \boldsymbol{y}^{N}$ and $\mathcal{Y}_N$ a bounded polyhedron, then $\left|V\left(\mathcal{Y}_M\left(\boldsymbol{y}^{UB}\right)\right)\right| = \left|V\left(\mathcal{Y}_M\left(\tilde{\boldsymbol{y}}^{UB}\right)\right)\right|$.*

Proof. Define

$$\mathcal{Y}^i \equiv \mathcal{Y}_M\left(\boldsymbol{y}^{\mathrm{N}}\right) + \bigoplus_{j=1}^{i} \operatorname{conv}\left\{\mathbf{0}, \boldsymbol{e}^j\right\}\left(y_j^{\mathrm{UB}} - y_j^{\mathrm{N}}\right) + \bigoplus_{j=i+1}^{p} \operatorname{conv}\left\{\mathbf{0}, \boldsymbol{e}^j\right\}\left(\tilde{y}_j^{\mathrm{UB}} - y_j^{\mathrm{N}}\right).$$

Using Theorem 4.2, it holds that $\mathcal{Y}^p = \mathcal{Y}_M\left(\boldsymbol{y}^{\mathrm{UB}}\right)$ and $\mathcal{Y}^0 = \mathcal{Y}_M\left(\tilde{\boldsymbol{y}}^{\mathrm{UB}}\right)$. For some $i = 1, \ldots, p-1$, let us consider $\mathcal{Y}^i$ and $\mathcal{Y}^{i+1}$, which can be rewritten as follows

$$
\begin{aligned}
\mathcal{Y}^i &= \mathcal{Y}_M\left(\boldsymbol{y}^{\mathrm{N}}\right) + \bigoplus_{j=i+1}^{p} \operatorname{conv}\left\{\mathbf{0}, \boldsymbol{e}^j\right\}\left(\tilde{y}_j^{\mathrm{UB}} - y_j^{\mathrm{N}}\right) + \\
&\quad + \bigoplus_{j=1}^{i} \operatorname{conv}\left\{\mathbf{0}, \boldsymbol{e}^j\right\}\left(y_j^{\mathrm{UB}} - y_j^{\mathrm{N}}\right) \\
&= \mathcal{Y}_M\left(\boldsymbol{y}^{\mathrm{N}}\right) + \bigoplus_{j=(i+1)+1}^{p} \operatorname{conv}\left\{\mathbf{0}, \boldsymbol{e}^j\right\}\left(\tilde{y}_j^{\mathrm{UB}} - y_j^{\mathrm{N}}\right) + \\
&\quad + \bigoplus_{j=1}^{i} \operatorname{conv}\left\{\mathbf{0}, \boldsymbol{e}^j\right\}\left(y_j^{\mathrm{UB}} - y_j^{\mathrm{N}}\right) + \operatorname{conv}\left\{\mathbf{0}, \boldsymbol{e}^{i+1}\right\}\left(\tilde{y}_{i+1}^{\mathrm{UB}} - y_{i+1}^{\mathrm{N}}\right) \\
\mathcal{Y}^{i+1} &= \mathcal{Y}_M\left(\boldsymbol{y}^{\mathrm{N}}\right) + \bigoplus_{j=(i+1)+1}^{p} \operatorname{conv}\left\{\mathbf{0}, \boldsymbol{e}^j\right\}\left(\tilde{y}_j^{\mathrm{UB}} - y_j^{\mathrm{N}}\right) + \\
&\quad + \bigoplus_{j=1}^{i+1} \operatorname{conv}\left\{\mathbf{0}, \boldsymbol{e}^j\right\}\left(y_j^{\mathrm{UB}} - y_j^{\mathrm{N}}\right) \\
&= \mathcal{Y}_M\left(\boldsymbol{y}^{\mathrm{N}}\right) + \bigoplus_{j=(i+1)+1}^{p} \operatorname{conv}\left\{\mathbf{0}, \boldsymbol{e}^j\right\}\left(\tilde{y}_j^{\mathrm{UB}} - y_j^{\mathrm{N}}\right) + \\
&\quad + \bigoplus_{j=1}^{i} \operatorname{conv}\left\{\mathbf{0}, \boldsymbol{e}^j\right\}\left(y_j^{\mathrm{UB}} - y_j^{\mathrm{N}}\right) + \operatorname{conv}\left\{\mathbf{0}, \boldsymbol{e}^{i+1}\right\}\left(y_{i+1}^{\mathrm{UB}} - y_{i+1}^{\mathrm{N}}\right).
\end{aligned}
$$

As $\left(\tilde{y}_{i+1}^{\mathrm{UB}} - y_{i+1}^{\mathrm{N}}\right)$, $\left(y_{i+1}^{\mathrm{UB}} - y_{i+1}^{\mathrm{N}}\right) > 0$ and $\mathcal{Y}_N$ is a bounded polyhedron, it follows from Theorem 4.3 that $\left|V\left(\mathcal{Y}^i\right)\right| = \left|V\left(\mathcal{Y}^{i+1}\right)\right|$. By assumption it holds that $\boldsymbol{y}^{\mathrm{UB}} - \boldsymbol{y}^{\mathrm{N}}$, $\tilde{\boldsymbol{y}}^{\mathrm{UB}} - \boldsymbol{y}^{\mathrm{N}} > 0$, and therefore the above reasoning can be repeated to conclude that $\left|V\left(\mathcal{Y}^0\right)\right| = \left|V\left(\mathcal{Y}^1\right)\right| = \cdots = \left|V\left(\mathcal{Y}^p\right)\right|$. In particular it holds that $\left|V\left(\mathcal{Y}^0\right)\right| = \left|V\left(\mathcal{Y}^p\right)\right|$. □

4.4.3 Maximum number of WSS LPs of the MCS for MOLP

In order to bound the maximum number of WSS LPs solved by the MCS, we use McMullen's upper bound theorem [17] which states that there is a general upper bound on the number of faces of a convex polyhedron P given a particular dimension p and number of vertices q. Let ϕ_i denote the number of i-dimensional faces of P, then it holds for $p \geq 2$ and $i = 0, \ldots, p-1$ that

$$
\phi_i \leq \sum_{r=0}^{\left\lfloor \frac{p}{2} \right\rfloor} \binom{r}{p-i-1} \binom{q-p+r-1}{r} + \sum_{r=\left\lfloor \frac{p}{2} \right\rfloor + 1}^{p} \binom{r}{p-i-1} \binom{q-r-1}{p-r}, \tag{9}
$$

which follows from rewriting McMullen's theorem (see e.g. [18]). Based on equation (9) ϕ_i is bounded by $\mathcal{O}\left(q^{\left\lfloor \frac{p}{2} \right\rfloor}\right)$ for all $i = 0, \ldots, p-1$.

During the WSS sandwich procedure, if the considered criterion space is bounded or modified to be bounded, two types of outcomes can occur:

1. **No expansion of the IPS.**
   In this case using the weight of the face of the IPS does not result in finding a new point that will expand the IPS, and thus the OPS will be updated to coincide with this particular face of the IPS. To verify whether the IPS is equal to the OPS, in the worst case every face of the IPS has to be used as a weight. As there are ϕ_{p-1} faces, it holds that the number of times this outcome can occur is bounded by $\mathcal{O}\left(q^{\lfloor \frac{p}{2} \rfloor}\right)$.

2. **Expansion of the IPS.**
   From the proof of Theorem 4.1 it follows that each combination of strictly positive coefficients of the vertices can be encountered only once. Therefore, at most $\sum_{i=0}^{p-1} \phi_i$ WSS LPs are performed which is again bounded by $\mathcal{O}\left(q^{\lfloor \frac{p}{2} \rfloor}\right)$, as p is fixed.

Based on these outcomes, the maximum number of WSS LPs solved by the sandwich procedure is bounded by $\mathcal{O}\left(q^{\lfloor \frac{p}{2} \rfloor}\right)$.

For the MCS it holds that $q \leq m\, 2^p$, where m represents the number of efficient points of $\mathcal{Y}_N$, as each efficient point can be projected 2^p times on each of the spanning vectors of $\{\boldsymbol{y} \mid y_i \leq y_i^{\mathrm{UB}}\}$. Combined with the results of the WSS sandwich algorithm, the number of WSS LPs solved by the MCS is bounded by $\mathcal{O}\left(m^{\lfloor \frac{p}{2} \rfloor}\, 2^{p\lfloor \frac{p}{2} \rfloor}\right)$ which is $\mathcal{O}\left(m^{\lfloor \frac{p}{2} \rfloor}\right)$.

5 Instances

In this section we describe two types of instances that are used to test the validity and numerical performance of the MCS.

5.1 Bensolvehedron

`Bensolvehedron` is a class of MOLP problems introduced on `bensolve.org` where the MOLP is denoted as:

$$\begin{aligned} \min \quad & \boldsymbol{C}^{\intercal}\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \end{aligned}$$

where cost matrix $\boldsymbol{C}$, constraint matrix $\boldsymbol{A}$ and constraint vector $\boldsymbol{b}$ have a fixed structure based on the input parameter p for the number of objectives and m for the complexity of the polyhedron. In general, the larger m is, the more facets in $\mathcal{Y}$ there are and the more variables n the MOLP has, where $n = (p + 2m)^p$. The problem is then defined as $\boldsymbol{A} = [\boldsymbol{I}_{n\times n}; -\boldsymbol{I}_{n\times n}]$, $\boldsymbol{b} = [\boldsymbol{1}_n\, \boldsymbol{0}_n]^{\intercal}$ and $\boldsymbol{C} \in \mathbb{R}^{p\times n}$ where each element in $\boldsymbol{C}$ ranges from $\left\{-\frac{p+2m-1}{2}, \ldots, \frac{p+2m-1}{2}\right\}$ with increments of 1 considering all possible combinations.

5.2 Rennen

In [7] a convex problem is analyzed with the following objectives and constraints:

$$
\begin{aligned}
f_1(\boldsymbol{x}) &= x_1 \\
f_2(\boldsymbol{x}) &= x_2 \\
f_3(\boldsymbol{x}) &= x_3 \\
x_1 &\geq (x_2 - 9)^2 + (x_3 - 3)^2 \\
x_2 &\geq (x_1 - 4)^2 + (x_3 - 3)^2 \\
x_3 &\geq (x_1 - 4)^2 + (x_2 - 9)^2.
\end{aligned}
$$

Note that for this problem there is a one-to-one relationship between the decision space and the criterion space. This problem instance is referred to as REN.

6 Results

The goal of this section is to show the performance of the MCS and the proposed distance approaches. In Section 6.1 we verify for certain instances whether all efficient points of the MOLP are found. In Section 6.2 we compare the performance of the MCS with the modification method of Solanki et al. [5] for an MOLP and convex MO, respectively. As their WSS LPs are fairly similar in the number of constraints and variables, the comparison is based on the number of WSS LPs needed to obtain the modified criterion space. Lastly, we investigate the use of a distance approach on the distance between the IPS and the OPS in Section 6.3. Here, we first compare the distance approaches SUB and CUR with SOL, the approach of [5]. Then, we investigate whether it is always beneficial to use a distance approach by comparing MCS without a distance calculation, referred to as NON, to the best performing approach that does use a distance calculation.

All results are obtained using an Intel Core i7-9700 3.00 GHz with 16GB RAM, and are implemented in Python 3.9.7 where Gurobi 9.5.2 is used to solve the WSS problems and the distance calculations.

6.1 Verification that MCS obtains $\mathcal{Y}_N$

We validate that MCS indeed finds $\mathcal{Y}_N$ for several instances of `Bensolvehedron` by comparing $\mathcal{Y}_N$ obtained with MCS to $\mathcal{Y}_N$ obtained with the solver of `bensolve.org`, which uses Benson's outer approximation [9]. As the latter method is able to find all efficient points for the MOLP, we know that the resulting $\mathcal{Y}_N$ is correct, which makes it well-suited for validation. Although there are alterations possible to the algorithm that allow for the optimization of convex instances as well [10], this is not implemented in the solver. Therefore, this solver is only used to validate the MOLP. As we aim to compare the exact fronts, we use $\varepsilon = 0$. The investigated instances are $(2, 0)$, $(2, 1)$, $(2, 2)$, $(2, 3)$, $(2, 4)$, $(3, 0)$, $(3, 1)$, $(3, 2)$ and $(4, 0)$, where the first digit represents p and the second digit represents m. For all these instances MCS was able to obtain $\mathcal{Y}_N$.

6.2 Comparison of modification methods

6.2.1 MOLP

We compare the performance of the MCS with the method of Solanki et al. [5] when generating $\mathcal{Y}_N$ for an MOLP, i.e. $\varepsilon = 0$, for the `Bensolvehedron` instance with $(p, m) = (3, 0)$[4].

[4]This is the same instance as in Section 4.1 with nadir point $\boldsymbol{y}^{\mathrm{N}} = (-1, -1, -1)$.

Here, we focus on the effect of the choice of $\boldsymbol{y}^{\mathrm{UB}}$ on the generated IPS of the different models. As the formulations of both WSS LPs are similar in size, its performance for a given weight will be comparable. Therefore, it is reasonable to compare the performance of MCS and [5] based on the number of WSS LPs solved. As the choice of distance calculation can affect the number of WSS LPs needed, we consider the number of WSS LPs using no distance calculation and the distance calculation in [5], which are NON and SOL, respectively. Here, NON serves as an upper bound on the number of WSS LPs needed to obtain the modified criterion space for the MCS and the modification method in [5].

Table 2 reports for the `Bensolvehedron` instance with $(p, m) = (3, 0)$ the number of WSS LPs solved given a particular $\boldsymbol{y}^{\mathrm{UB}}$ using NON or SOL with either the modification method in [5] or the MCS. Furthermore, the table reports the number of vertices and the number of dominated points of the IPS. Note that the generated structure of the IPS given $\varepsilon = 0$ is predetermined, and therefore the number of vertices of the IPS — and per extension the number of efficient points of the IPS — will always be the same irrespective of the chosen distance measure. Note that for $\boldsymbol{y}^{\mathrm{UB}} = (-2, -2, -2)$ a part of $\mathcal{Y}_N$ is cut off.

In general, the table shows that when $\boldsymbol{y}^{\mathrm{UB}}$ is poorly chosen, the modification method in [5] can result in the optimization of many uninformative LPs. As shown in Section 4.4.3, for the MCS it holds that the maximum number of WSS LPs is not dependent on $\boldsymbol{y}^{\mathrm{UB}}$ as long as $\boldsymbol{y}^{\mathrm{UB}} > \boldsymbol{y}^{\mathrm{N}}$, as the number of IPS vertices will stay constant.

In particular, the table shows for this instance that if no distance calculation is used and $\boldsymbol{y}^{\mathrm{UB}} > \boldsymbol{y}^{\mathrm{N}}$, the number of WSS LPs solved for the MCS is smaller than or equal to the number of WSS LPs solved with [5]. Furthermore, it shows that the IPS representation of the MCS contains fewer dominated points. In case SOL is used, the difference in performance is less defined but still notable. Here, exploiting the structure of the OPS helps to reduce the number of WSS LPs solved compared to not using a distance calculation.

Table 2: Number of WSS LPs solved using the modification method of Solanki et al. [5] and MCS for the `Bensolvehedron` with $(p, m) = (3, 0)$ and $\varepsilon = 0$ given a certain $\boldsymbol{y}^{\mathrm{UB}}$ using NON and SOL.

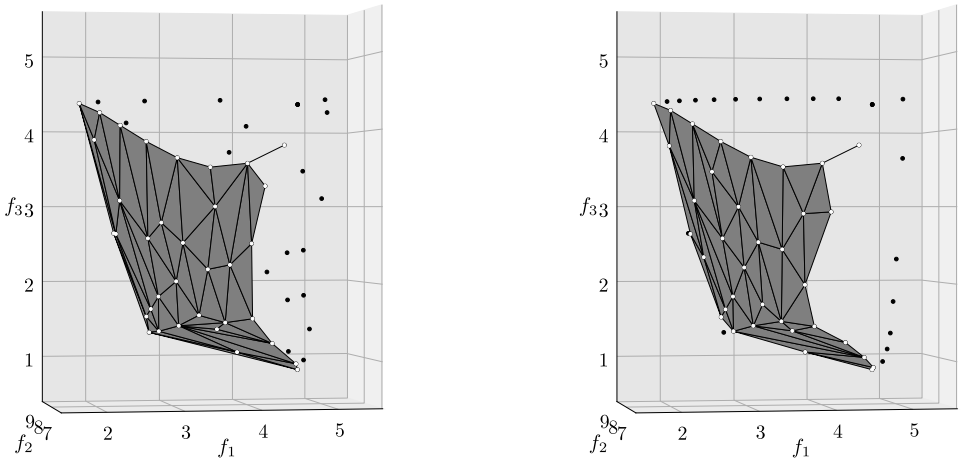
	Solanki et al.				MCS			
	WSS LPs		IPS		WSS LPs		IPS	
$\boldsymbol{y}^{\mathrm{UB}}$	NON	SOL	Vert.	Not eff.	NON	SOL	Vert.	Not eff.
$(-2, -2, -2)$	50	40	19	4	58	37	19	4
$(-1, -1, -1)$	47	32	16	4	52	33	16	4
$(0, 0, 0)$	52	34	22	10	52	36	22	10
$(10, 10, 10)$	205	168	85	73	52	41	22	10
$(100, 100, 100)$	178	167	73	61	52	39	22	10

Abbreviations: vert. is the number of vertices of the IPS and not eff. denotes the number of dominated points of the generated IPS.

6.2.2 Convex MO

We use the instance in [7], REN, to compare the performance of the MCS with the modification method of Solanki et al. [5]. Again, we focus on the effect of the selection of $\boldsymbol{y}^{\mathrm{UB}}$ on the generated IPS of the different models. For the first selection of $\boldsymbol{y}^{\mathrm{UB}}$, we use the

pseudonadir point, i.e. $\boldsymbol{y}^{\mathrm{UB}} = \boldsymbol{y}^{\mathrm{pN}} \approx (4.88, 9.00, 4.38)$[5]. Figure 6 shows the IPS obtained after 50 iterations, excluding the initialization, using the modification method of Solanki et al. [5] and the MCS. Although both methods result in 19 dominated points and 35 efficient points on the IPS, MCS obtained a better ε using SOL of 0.0170 vs. 0.0206 for the modification method of [5]. Here, the structure of the OPS as used in MCS helps to reduce the distance.



(a) Solanki (b) MCS

Figure 6: IPS of REN using $\boldsymbol{y}^{\mathrm{UB}} = \boldsymbol{y}^{\mathrm{pN}}$ with 50 iterations. The dots depict all found points of the IPS, where the white dots and the gray area represent the efficient part of the IPS.

For our second selection of $\boldsymbol{y}^{\mathrm{UB}}$ we use a point that is larger than the nadir point, namely $(7.5, 10, 5)$. This results in 21 dominated points and $\varepsilon = 0.0151$ for the MCS and 31 dominated points with $\varepsilon = 0.0254$ using the modification method of [5], see Figure 7. Again, we see that a poor choice of $\boldsymbol{y}^{\mathrm{UB}}$ heavily impacts the performance of the modification method of [5], whereas it does not affect MCS.

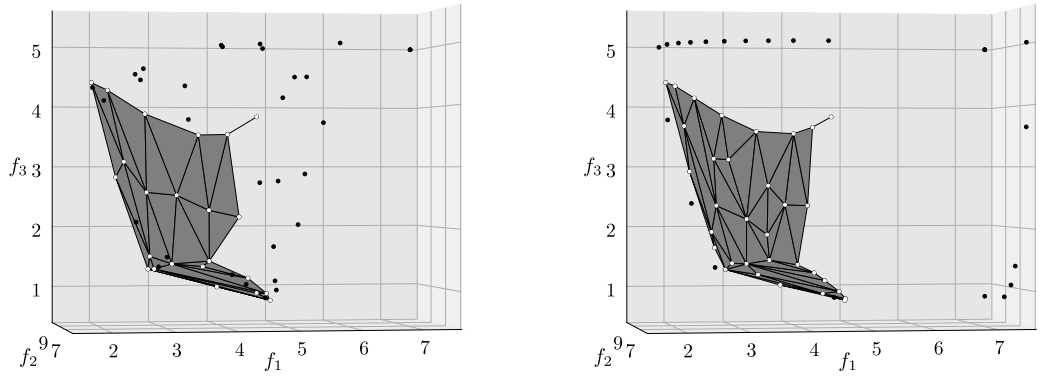
6.3 Impact of a distance approach

6.3.1 Comparison of distance approaches

Based on our findings in Section 6.2 we use the MCS to compare the three different distance approaches introduced in Section 4.2: SUB, CUR and SOL. To evaluate their performance we compare them based on the required solution time and the number of WSS and distance LPs solved given a particular ε for the `Bensolvehedron` instance with $(p, m) = (3, 2)$ and REN.

Table 3 reports for both instances for several choices of ε the total solution time, which is split into time required for solving the WSS LPs and time required for solving the distance calculation LPs, and denotes the number of LPs solved for WSS and the distance calculation. Furthermore, the table specifies additional information related to the distance calculation such as the average number of constraints and the average time to solve the corresponding distance LP. Bold marked rows indicate the method that has the shortest algorithm time, measured by summing the time for the solution of the WSS LPs and the distance LPs. Overhead time is not taken into account.

[5]See Figure 15 in [7].



(a) Solanki (b) MCS

Figure 7: IPS of REN using $\boldsymbol{y}^{\text{UB}} = (7.5, 10, 5)$ with 50 iterations. The dots depict all found points of the IPS, where the white dots and the gray area represent the efficient part of the IPS.

For both instances it holds that for all choices of ε the number of WSS LPs solved is smaller using SOL compared to SUB and CUR. This makes sense, as SUB and CUR take less information from the OPS into account and may therefore overestimate the distance as defined by [5]. This has two consequences. First of all, more iterations might be needed to ensure that ε is below the set threshold, even though the desired maximum distance may have already been reached. This effect, however, seems limited as the found ε using SUB and CUR is not substantially smaller than the one reported for SOL. Note that the found ε is determined with SOL, therefore they are comparable. Second, a face that did not maximally decrease the distance between the OPS and IPS might be selected. Therefore, for smaller choices of ε the time spent on solving the WSS using SOL is significantly smaller.

The general downside of SOL, however, is that many more LPs need to be solved for the distance calculation. Especially for $\varepsilon = 0.001$ the number of LPs solved grows fast. Combined with the fact that the time spent on solving each distance LP increases fast as well, almost all of the solution time for SOL is spent on solving distance LPs when ε becomes small.

For the `Bensolvehedron` instance, CUR seems to perform the best as it is able to balance the time spent on solving WSS LPs and distance LPs. By incorporating more information about the OPS than SUB, CUR needs to solve fewer WSS LPs than SUB while requiring fewer distance recalculations than SOL. However, for sufficiently small ε it seems that SUB is still an acceptable method as for CUR the distance time increases more rapidly as the time spent per distance LP increases with smaller ε.

For REN, SUB seems to outperform the other distance approaches. The main difference is that in comparison to the previous instance, the WSS LP is solved much faster. Therefore, the approach that has the fastest distance calculation will perform the best in terms of time. As the average size of the distance LP of SUB is the smallest, these LPs remain more tractable than those for CUR and SOL. This reduces the solution time of the distance LPs, even though SUB solves more WSS and distance LPs than CUR.

In order to compare the overall outcomes of the three distance approaches, we have

Table 3: Comparison of several distance approaches for different choices of ε. Here, the solution time in seconds is reported for two categories: time to solve the WSS LP and distance LP. **Bold** indicates the method which has the fastest algorithm, measured by summing the time for WSS and distance LP. The number of LPs solved is reported both for the WSS and the distance calculation. In addition, we report the final found ε using SOL and for the distance calculation we report the average number of constraints in the solved distance LP and the average time in milliseconds to solve this LP.

		ε		Sol. time		LPs solved		Dist. calc.	
Inst.	Mth.	Crit.	Found	WSS	Dist.	WSS	Dist.	Cstr.	t/LP
(3, 2)	SUB	0.100	0.0953	12.8	0.1	23	72	9.4	1.6
(3, 2)	SUB	0.050	0.0479	21.0	0.2	35	125	9.5	1.5
(3, 2)	SUB	0.025	0.0247	48.3	0.5	74	304	9.7	1.5
(3, 2)	SUB	0.010	0.0092	100.0	1.0	147	637	9.8	1.5
(3, 2)	SUB	0.005	0.0050	206.2	2.0	299	1,321	9.7	1.5
(3, 2)	SUB	0.001	0.0010	685.3	6.1	898	3,404	9.9	1.8
(3, 2)	**SUB**	0	0	1,442.8	7.7	2,058	4,643	10.5	1.7
(3, 2)	**CUR**	0.100	0.0953	10.4	0.2	20	56	13.0	3.0
(3, 2)	**CUR**	0.050	0.0490	18.2	0.4	31	100	18.5	3.7
(3, 2)	**CUR**	0.025	0.0247	35.2	1.0	55	196	30.5	5.3
(3, 2)	**CUR**	0.010	0.0099	79.1	4.5	116	482	61.0	9.2
(3, 2)	**CUR**	0.005	0.0049	144.6	12.9	207	875	106.0	14.8
(3, 2)	**CUR**	0.001	0.0010	535.5	115.1	750	2,675	358.5	43.0
(3, 2)	CUR	0	0	1,500.6	202.9	2,101	3,428	712.0	59.2
(3, 2)	SOL	0.100	0.0953	10.5	0.7	21	304	13.5	2.3
(3, 2)	SOL	0.050	0.0492	18.1	2.3	31	754	18.5	3.1
(3, 2)	SOL	0.025	0.0247	34.8	13.4	56	2,724	31.0	4.9
(3, 2)	SOL	0.010	0.0099	74.6	107.8	113	11,617	59.5	9.3
(3, 2)	SOL	0.005	0.0050	135.5	562.3	199	36,001	102.0	15.6
(3, 2)	SOL	0.001	0.0010	499.5	18,549.2	685	362,694	334.0	51.1
(3, 2)	SOL	0	0	OOT	OOT	OOT	OOT	OOT	OOT
REN	**SUB**	0.100	0.0902	0.0	0.2	22	72	9.4	2.2
REN	**SUB**	0.050	0.0497	0.1	0.4	38	145	9.4	2.5
REN	**SUB**	0.025	0.0243	0.1	0.6	60	247	9.7	2.3
REN	**SUB**	0.010	0.0090	0.2	1.5	135	585	11.5	2.6
REN	**SUB**	0.005	0.0048	0.4	3.5	275	1,257	14.2	2.8
REN	**SUB**	0.001	0.0010	2.0	24.1	1,215	5,990	27.4	4.0
REN	SUB	0	0.0000	NA	NA	NA	NA	NA	NA
REN	**CUR**	0.100	0.0887	0.0	0.2	21	65	13.5	2.9
REN	**CUR**	0.050	0.0497	0.0	0.4	31	105	18.5	3.8
REN	CUR	0.025	0.0245	0.1	0.7	44	161	25.0	4.6
REN	CUR	0.010	0.0094	0.1	3.0	91	376	48.5	8.0
REN	CUR	0.005	0.0049	0.2	7.2	146	624	76.0	11.6
REN	CUR	0.001	0.0010	0.9	97.8	551	2,547	278.5	38.4
REN	CUR	0	0	NA	NA	NA	NA	NA	NA
REN	SOL	0.100	0.0887	0.0	0.9	20	270	13.0	3.2
REN	SOL	0.050	0.0497	0.0	2.2	28	598	17.0	3.7
REN	SOL	0.025	0.0245	0.1	7.4	42	1,480	24.0	5.0
REN	SOL	0.010	0.0098	0.1	53.0	82	6,160	44.0	8.6
REN	SOL	0.005	0.0049	0.2	266.4	143	19,458	74.5	13.7
REN	SOL	0.001	0.0010	0.8	11,784.8	512	258,570	259.0	45.6
REN	SOL	0	0	NA	NA	NA	NA	NA	NA

Abbreviations: sol. for solution, dist. for distance, calc. for calculation, inst. for instance, mth. for method, crit. for criterion, cstr. for constraints, t/LP for time per LP, OOT for out-of-time and NA for not available.

summarized our findings based on the two instances in Table 4. Note that, even though we do not report overhead times, SUB will in general have more overhead than CUR and SOL.

Table 4: Comparison of distance approaches based on the perceived outcomes in Table 3, where **o** denotes a relatively stable amount and **+** denotes an increase in time spent/number LPS solved if ε is smaller (more **+**'s indicate a steeper increase).

	SUB	CUR	SOL
Overhead for recalculation	Yes	No	No
Time per OPS LP w.r.t. ε	**o**	**+**	**+**
Number of OPS LPs w.r.t. ε	**++**	**+**	**+++**
Number of WSS LPs w.r.t. ε	**++**/**+++**	**++**	**+**

6.3.2 Relevance of a distance approach

In the previous section we investigated which of the suggested distance approaches works best, i.e. has the best solution time to reach ε. However, the question remains whether the use of a distance calculation is always beneficial compared to not using a distance calculation. As seen in Section 6.3.1, a considerable amount of time can be spent on solving the distance LPs, which could have been spent on solving WSS LPs otherwise. This is especially the case for REN where the WSS LPs are solved relatively quickly.

To investigate whether and when the distance calculation is useful, we run the WSS sandwich algorithm without using a distance approach where we set a maximum time limit for the total solution time of the WSS LPs. This time limit is based on the best found approach in Section 6.3.1 — CUR for the `Bensolvehedron` instance and SUB for REN — where the time limit is based on the total solution time for both the WSS and the distance calculation. Afterwards we determine ε using the distance definition of [5] and compare it to the reference ε.

Table 5 reports for both instances the ε obtained with a given time limit, for which we also report the total number of WSS LPs performed. Note that, as SUB and CUR overestimate the maximum distance, the maximum distance of the reference is likely to be lower than the reference ε.

For the `Bensolvehedron` instance it holds that ε obtained without distance calculation is significantly higher than the reference value, although a fairly similar amount of WSS LPs is solved for NON compared to the reference. For this instance it is indeed beneficial to select the most promising face to be able to decrease the distance between the IPS and the OPS as much as possible.

The conclusion for REN is different, as the obtained ε is smaller than the reference value. Previously we concluded that the distance calculation takes up most of the total computation time. Allowing this time to be spent on WSS LPs can result in a factor 10 more WSS LPs solved compared to the reference. In this case the added benefit of a distance approach is limited.

Table 5: The found ε using no distance calculation given a certain time limit based on the best performing distance approach in Section 6.3.1. Additionally the number of WSS LPs are reported.

Inst.	Time Limit (s)	Reference ε	Reference WSS	NON ε	NON WSS
$(3,2)$	10.6	0.0953	20	0.129	21
$(3,2)$	18.6	0.0490	31	0.073	32
$(3,2)$	36.3	0.0247	55	0.061	58
$(3,2)$	83.5	0.0099	116	0.057	124
$(3,2)$	157.5	0.0049	207	0.047	233
$(3,2)$	650.6	0.0010	750	0.006	912
$(3,2)$	1,703.5	0	2,101	0	2,400
REN	0.2	0.0902	22	0.018	112
REN	0.4	0.0497	38	0.006	279
REN	0.7	0.0243	60	0.005	447
REN	1.8	0.0090	135	0.003	1,137
REN	3.9	0.0048	275	0.002	2,299
REN	26.1	0.0010	1,215	0.001	13,386
REN	NA	0	NA	NA	NA

Abbreviations: inst. for instance and NA for not available.

7 Discussion

This work provides an elegant modification of the criterion space that helps to reduce the number of WSS LPs to be solved in a WSS sandwich algorithm. This modification, referred to as the MCS, is an advancement on the formulation of Solanki et al. [5], as it reduces the number of unnecessary searches towards dominated areas. Besides its explainability, a great advantage of the MCS is that its principles can be easily incorporated within an existing framework to solve convex problems. Therefore, the MCS is an accessible method for DMs.

Based on the structure of the MCS we have shown several useful properties. First, the MCS is able to obtain all nondominated points of an MOLP in a finite number of expansions of the IPS, which is bounded in terms of the number of nondominated vertices. We validated that the MCS is indeed able to obtain all nondominated points for a couple of MOLP instances using Benson's outer approximation [9], which is a method that is proven to be able to find all nondominated points of an MOLP. Second, the maximum number of WSS LPs of the MCS is not dependent on the choice of $\boldsymbol{y}^{\mathrm{UB}}$ as long as $\boldsymbol{y}^{\mathrm{UB}} > \boldsymbol{y}^{\mathrm{N}}$, whereas the modification method of Solanki et al. [5] is. In comparison to [5], the MCS is especially useful in case it is difficult to get a good estimation of $\boldsymbol{y}^{N}$, and as shown in our results reduces the number of WSS LPs that is necessary to obtain the non-dominated set up to ε-precision.

Furthermore, we have proposed two heuristic approaches to calculate the distance between the inner and outer approximation, referred to as SUB and CUR, that can be used as an alternative for the distance calculation of Solanki et al. [5], SOL. SUB, CUR and SOL

can be used to select the next weight used in the next WSS iteration. SUB and CUR approximate SOL by providing overestimations to SOL, resulting in overall faster approaches. As apparent from our results, when using SOL almost all computation time is spent on determining which weight to select, especially when ε is small. SUB and, to a lesser extent, CUR only spend a fraction of their total solution time on the distance calculation. In fact, even though the number of WSS LPs solved increases using SUB and CUR, the total solution time of the LPs of SUB and CUR is much faster than of SOL. Based on our findings, SUB is a better option than CUR if either ε is very small or if the WSS LP is relatively quick to solve.

We do like to make some remarks regarding the distance calculation. First, the distance definition which forms the basis of SOL, SUB and CUR does not determine the weight for which the Euclidean distance between the IPS and the OPS is the largest, but it selects the weight of the corresponding halfspace that is furthest from the OPS. This difference in interpretation is important, as the maximum determined distance using the definition of [5] can sometimes increase over the iterations, which might be counterintuitive. An alternative distance definition would be the Hausdorff distance as used in [13], however computing this distance metric is computationally expensive.

Second, we do like to point out that for the above results we did not parallelize our code. If we would parallelize for example the distance calculation procedure, the performance of CUR will most likely benefit more from this than SUB. For smaller ε, CUR has a larger fraction of total time spent on the distance LPS, parallelizing this will drastically decrease the time required for distance calculation.

Third, although a distance calculation will in general help to select promising weights, the added benefit can be limited in case the WSS LP is solved very quickly in comparison to the distance calculation LP. In those cases it is better to not use a distance calculation and use the additional time to generate a lot of WSS solutions to improve the granularity of the generated non-dominated set. Then, for the resulting solution, ε could be determined (or approximated with SUB or CUR) in order to get an idea of the accuracy of the approximation.

In our paper we do not necessarily focus on providing a method that is the fastest to obtain an ε distance, such as [13]. Instead, our overall intention is to make MO methods more appealing to be incorporated in applied fields and we do so by proposing comprehensible algorithms. A great example for this is the use of MO for diet optimization, our main motivation for this work, as there are multiple conflicting goals which impact the composition of a food basket [19]. That is, an ideal food basket has to be culturally appropriate, affordable, nutritious and environmentally friendly. By showing trade-offs between these various goals, organizations such as the World Food Programme can enhance their analyses that use diet optimization [20, 21, 22]. We think that the current diet optimization literature can still improve upon its MO analyses, as most research either focuses on methods that can overlook relevant trade-offs, such as epsilon constraint methods [23, 24, 25] or WSS with a fixed weight only [26], suboptimal methods such as lexicographic optimization [27] or methods specifically designed for two objectives [22]. Highlighting algorithms that are explainable, while at the same time having useful properties, such as the MCS will help applied fields to obtain more useful insights from their analyses.

As a last remark, we would like to discuss the (dis)similarities between the MCS and the dummy points method as proposed in [7]. The dummy points method is an augmentation of the general WSS sandwich algorithm, where it artificially alters the IPS by adding dominated points in each step of the algorithm. This resulting structure is somewhat similar to the structure of the MCS, as the MCS also flattens the dominated side. The

main advantage of the MCS over the dummy point method is that one does not need to keep track which dummy points are redundant, i.e. that are linear combinations of other dummy points, and that it requires less effort to be well-implemented.

Conflict of interest statement

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Author contributions

MK was responsible for the overall research and writing. MB had an advisory role throughout the research. MB and HF helped to structure and revise the text. HF initiated the research. All authors contributed to the manuscript and approved the submitted version.

Funding

The Zero Hunger Lab is funded by the Ministry of Foreign Affairs of the Netherlands and Tilburg School of Economics and Management.

Data availability

All data generated or analysed during this study are included in this manuscript or publicly available.

References

[1] Matthias Ehrgott. *Multicriteria optimization*, volume 491. Springer Science & Business Media, Berlin, Heidelberg, 2005.

[2] Harold P Benson. Hybrid approach for solving multiple-objective linear programs in outcome space. *Journal of Optimization Theory and Applications*, 98(1):17–35, 1998.

[3] Kaisa Miettinen. *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media, New York, 1999.

[4] Margaret M Wiecek, Matthias Ehrgott, and Alexander Engau. Continuous multiobjective programming. In *Multiple criteria decision analysis*, pages 739–815. Springer, Boston, 2016.

[5] Rajendra S Solanki, Perry A Appino, and Jared L Cohon. Approximating the noninferior set in multiobjective linear programming problems. *European Journal of Operational Research*, 68(3):356–373, 1993.

[6] David L Craft, Tarek F Halabi, Helen A Shih, and Thomas R Bortfeld. Approximating convex pareto surfaces in multiobjective radiotherapy planning. *Medical Physics*, 33(9):3399–3407, 2006.

[7] Gijs Rennen, Edwin R van Dam, and Dick den Hertog. Enhancement of sandwich algorithms for approximating higher-dimensional convex pareto sets. *INFORMS Journal on Computing*, 23(4):493–517, 2011.

[8] Kathrin Klamroth, Jørgen Tind, and Margaret M Wiecek. Unbiased approximation in multicriteria optimization. *Mathematical Methods of Operations Research*, 56(3):413–437, 2003.

[9] Harold P Benson. An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem. *Journal of Global Optimization*, 13(1):1–24, 1998.

[10] Lizhen Shao and Matthias Ehrgott. Approximately solving multiobjective linear programmes in objective space and an application in radiotherapy treatment planning. *Mathematical Methods of Operations Research*, 68(2):257–276, 2008.

[11] Matthias Ehrgott, Lizhen Shao, and Anita Schöbel. An approximation algorithm for convex multi-objective programming problems. *Journal of Global Optimization*, 50(3):397–416, 2011.

[12] Matthias Ehrgott, Andreas Löhne, and Lizhen Shao. A dual variant of Benson's "outer approximation algorithm" for multiple objective linear programming. *Journal of Global Optimization*, 52(4):757–778, 2012.

[13] Rasmus Bokrantz and Anders Forsgren. An algorithm for approximating convex pareto surfaces based on dual techniques. *INFORMS Journal on Computing*, 25(2):377–393, 2013.

[14] Indraneel Das. *Nonlinear multicriteria optimization and robust optimality.* Rice University, Houston, Texas, 1997.

[15] Bernd Schandl, Kathrin Klamroth, and Margaret M Wiecek. Norm-based approximation in multicriteria programming. *Computers & Mathematics with Applications*, 44(7):925–942, 2002.

[16] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena scientific, Belmont, MA, 1997.

[17] Peter McMullen. The maximum numbers of faces of a convex polytope. *Mathematika*, 17(2):179–184, 1970.

[18] Komei Fukuda. Polyhedral computation. 2020.

[19] Corné van Dooren. A review of the use of linear programming to optimize diets, nutritiously, economically and environmentally. *Frontiers in Nutrition*, 5:48, 2018.

[20] Amy Deptford and Andrew Hall. Cost of the Diet. A practitioner's guide. Version 2., 2014.

[21] Indira Bose, Giulia Baldi, Lynnda Kiess, and Saskia de Pee. The "Fill the Nutrient Gap" analysis: An approach to strengthen nutrition situation analysis and decision making towards multisectoral policies and systems change. *Maternal & Child Nutrition*, 15(3):e12793, 2019.

[22] Melissa F Koenen, Marleen Balvert, and Hein Fleuren. Bi-objective goal programming for balancing costs vs. nutritional adequacy. *Frontiers in Nutrition*, 9, 2022.

[23] Corné van Dooren, Marcelo Tyszler, Gerard FH Kramer, and Harry Aiking. Combining low price, low climate impact and high nutritional value in one shopping basket through diet optimization by linear programming. *Sustainability*, 7(9):12837–12855, 2015.

[24] Jingjing Yin, Xinhuan Zhang, Wei Huang, Lingxuan Liu, Yufang Zhang, Degang Yang, Yun Hao, and Yaning Chen. The potential benefits of dietary shift in China: Synergies among acceptability, health, and environmental sustainability. *Science of The Total Environment*, 779:146497, 2021.

[25] Eliseu Verly-Jr, Aline Martins de Carvalho, Dirce Maria Lobo Marchioni, and Nicole Darmon. The cost of eating more sustainable diets: A nutritional and environmental diet optimisation study. *Global Public Health*, 17(6):1073–1086, 2022.

[26] Hassan Eini-Zinab, Seyyed Reza Sobhani, and Arezoo Rezazadeh. Designing a healthy, low-cost and environmentally sustainable food basket: an optimisation study. *Public health nutrition*, 24(7):1952–1961, 2020.

[27] Louise Seconda, Hélène Fouillet, Jean-François Huneau, Philippe Pointereau, Julia Baudry, Brigitte Langevin, Denis Lairon, Benjamin Allès, Mathilde Touvier, Serge Hercberg, et al. Conservative to disruptive diets for optimizing nutrition, environmental impacts and cost in French adults from the nutrinet-santé cohort. *Nature Food*, 2(3):174–182, 2021.

[28] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996.

Appendix

A Overestimation distance calculation of Craft et al. [6]

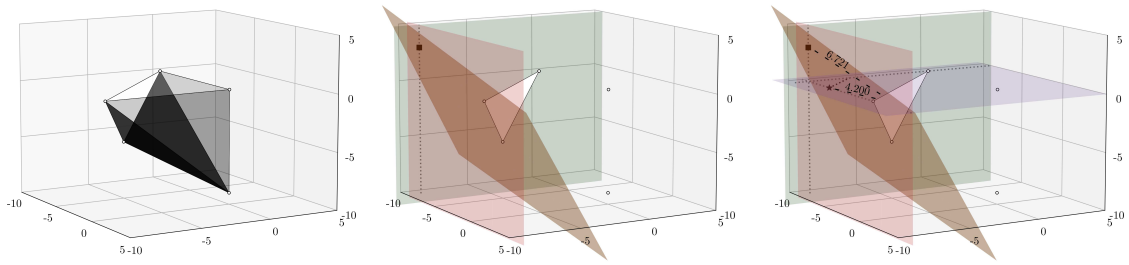
Here, we show that the distance calculation introduced in Craft et al. [6] can result in an overestimation of (1). We provide a small example based on `Bensolvehedron` with $p = 3$ and $m = 0$, where the intersection of the OPS halfspaces of the vertices of the current considered face of the IPS does not result in the correct distance. That is, the closed form expression results in an overestimation. This example shows that only using a subset of the OPS halfspaces can result in an overestimation of the distance, even though the intersection of this subset of OPS halfspaces is unique. For clarity we do not scale the objectives.

Let us assume that our current IPS consists of the anchor points $(-9, 0, 0), (0, -9, 0)$, $(0, 0, -9)$, upper bound $(0, 0, 0)$ and an efficient point $(-5, -5, -5)$ found during the first iteration of the algorithm. Figure 8a shows the current IPS. The corresponding OPS halfspaces of the anchor points are $[1\,0\,0]^\intercal y \geq -9$, $[0\,1\,0]^\intercal y \geq -9$ and $[0\,0\,1]^\intercal y \geq -9$, the OPS halfspace of $(-5, -5, -5)$ is $\left[\frac{1}{\sqrt{3}}\ \frac{1}{\sqrt{3}}\ \frac{1}{\sqrt{3}}\right]^\intercal y \geq -8.66025$ and the other OPS halfspaces are $[1\,0\,0]^\intercal y \leq 0$, $[0\,1\,0]^\intercal y \leq 0$, and $[0\,0\,1]^\intercal y \leq 0$.

Now, let us consider the inner normal of the IPS face that belongs to point $(-5, -5, -5)$ and anchor points $(-9, 0, 0)$ and $(0, -9, 0)$. This face defined as $[45\ 45\ -9]^\intercal y \leq -405$ is highlighted white in Figure 8a. To determine the distance from this IPS face to the OPS, Craft et al. [6] use the OPS halfspaces that correspond to these points to find an intersection point. As these halfspaces intersect uniquely, they can use their closed form expression. The found intersection point $(-9, -9, 3)$ is indicated as a square in Figure 8b. Using the distance approach of Craft et al. [6], the distance between the square and the face of the IPS is ≈ 6.721.

If we now consider the distance calculation according to Solanki et al. [5], we find a distance of ≈ 4.200. Here, the OPS inequality $[0\,0\,1]^\intercal y \leq 0$ cuts away the former intersection point, which is shown in Figure 8c. This OPS inequality is not one of the OPS inequalities that belong to the vertices of the white IPS face, and is therefore not included in the distance calculation of Craft et al. [6]. This results in the correct distance from the IPS to the OPS, where we find point $(-6, -9, 0)$ indicated with a star. Note that all points on the line indicated with the black dotted line in Figure 8c result in the correct distance. Thus, this example shows that the closed form expression of Craft et al. [6] may result in an overestimation of the distance.

Note that in this example the white face has a mixed normal. After determining the distance to the OPS, Craft et al. [6] modify this normal to obtain a non-negative normal for the WSS sandwich algorithm. The fact that it is a mixed normal does not alter the distance calculation.



(a) (b) (c)

Figure 8: In (a) the current IPS is shown. In (b) the intersection point, indicated with a square, is determined with the OPS halfspaces corresponding to the vertices of the white face. In (c) the distance to the intersection point is 6.721, whereas the correct distance from the white face to the OPS is 4.200. In the optimization we find the point indicated with a star, however all points on the black dotted line will result in the same distance.

B Implementation choices and details

In this section we provide comments related to the implementation of Algorithm 1 of Section 4.3:

- In general it is common practice to normalize the criterion space to avoid a bias towards objectives with a higher magnitude [6, 13], which increases numerical stability of the algorithm as well;
- In the initialization we use $\boldsymbol{y}^{\text{UB}}$ to construct the IPS and the OPS. An often used option for $\boldsymbol{y}^{\text{UB}}$ is the pseudonadir point, which is relatively easy to determine although it does not guarantee to find the complete $\mathcal{Y}_N$ front as it might cut off some efficient points;
- In order to determine the convex hull of the IPS and to obtain its halfspaces, we make use of `qhull`[6] based on the work of [28];
- In case the initialization from Line 1-8 does not contain $p+1$ linearly independent points, it is not possible to construct a convex hull and H in Line 9 will be empty resulting in a premature termination of the algorithm;
- In Line 16 the algorithm always adds $f(\bar{x})$ to the IPS, however similar to [5] one can also first check whether $f(\bar{x})$ will expand the current IPS. If that is not the case, find the next halfspace that has the lowest distance to the OPS that does expand the IPS.

[6] `www.qhull.org`