




# On Dealing with Minima at the Border of a Simplicial Feasible Area in Simplicial Branch and Bound

Boglárka G.-Tóth<sup>1</sup> · Eligius M. T. Hendrix<sup>2,3</sup>  · Leocadio G. Casado<sup>4</sup> · Frédéric Messine<sup>5</sup>

Received: 2 April 2023 / Accepted: 11 June 2024 / Published online: 22 July 2024  
© The Author(s) 2024

## Abstract

We consider a simplicial branch and bound Global Optimization algorithm, where the search region is a simplex. Apart from using longest edge bisection, a simplicial partition set can be reduced due to monotonicity of the objective function. If there is a direction in which the objective function is monotone over a simplex, depending on whether the facets that may contain the minimum are at the border of the search region, we can remove the simplex completely, or reduce it to some of its border facets. Our research question deals with finding monotone directions and labeling facets of a simplex as border after longest edge bisection and reduction due to monotonicity. Experimental results are shown over a set of global optimization problems where the feasible set is defined as a simplex, and a global minimum point is located at a face of the simplicial feasible area.

---

Communicated by Giancarlo Bigi.

This work has been funded by MCIN/AEI/10.13039/501100011033 and by “ERDF A way of making Europe”, Grant PID2021-123278OB-I00 of the Spanish ministry of Science and Innovation and grant TKP2021-NVA-09 of the Ministry for Innovation and Technology, Hungary.

---

✉ Eligius M. T. Hendrix  
eligius.hendrix@wur.nl

Boglárka G.-Tóth  
boglarka@inf.szte.hu

Leocadio G. Casado  
leo@ual.es

Frédéric Messine  
frederic.messine@laplace.univ-tlse.fr

<sup>1</sup> University of Szeged, Szeged, Hungary

<sup>2</sup> Universidad de Málaga, Málaga, Spain

<sup>3</sup> Wageningen University, Wageningen, The Netherlands

<sup>4</sup> University of Almería, CeiA3, Almería, Spain

<sup>5</sup> LAPLACE-ENSEEIH, Toulouse-INP, University of Toulouse, Toulouse, France

**Keywords** Simplex · Branch and bound · Monotonic · Directional derivative · Face

## 1 Introduction

Optimization over a simplex appears naturally in many applications: machine learning, signal processing, control, and game theory [1]. In such problems, the feasible set is determined by linear constraints that form a simplex.

Branch and Bound (B&B) algorithms aim at finding the set of global minimum points up to a certain accuracy. These types of algorithms have the advantage of providing an exhaustive search and the requested precision of the solution. Their computational burden increases with the dimension of the problem and the accuracy. Therefore, they are usually applied to low-dimensional problems. The overview book [11] discusses why partitioning a box-constrained area into simplices may have an advantage over using box-shaped partition sets. Their focus is mainly on Lipschitz-based bounds to guarantee the required accuracy. In this paper, we consider the feasible set as a simplicial set, that is a way to avoid dealing with linear constraints.

The use of longest edge bisection in simplicial branch and bound has a long tradition and has been studied thoroughly, see [3, 11, 13]. Recently, new ways of bounding the objective over a simplex have been studied [8–10, 15], where bounds on the first and second derivatives are considered. Bounding rules using gradient information were found to outperform other bounding rules, allowing monotonicity to be taken into account.

Monotonicity considerations to remove subsets and to reduce dimension have a long tradition in Interval Arithmetic (IA) based B&B, see [2, 12]. The basic property is to be able to remove interior boxes where the function is monotone over the box and to reduce the dimension with respect to monotone components, when a box facet is at the boundary of the search space.

The ideas of monotonicity have been extended for simplicial partition sets recently [5, 8, 9, 14]. One of our questions is how to handle the mathematical properties in an efficient way in a simplicial branch and bound. The main question is how to extend the ideas of a simplicial dimension reduction. This requires identifying whether a corresponding facet is at the boundary of the feasible set. The latter will be defined more precisely with the terminology of border faces.

The main contributions of this study are:

- A mathematical theory to support new monotonicity tests over a simplex that can be used independently on how reliable bounds of the objective and its derivatives were obtained. Here, we use Interval Arithmetic because it is a simple way to calculate the bounds.
- A method to determine when a facet of a simplex can be considered border of the search space.
- A computational algorithm.
- Computational illustration of the new monotonicity tests on artificially generated problems.

To investigate the research questions, Sect. 2 introduces the notation and mathematical properties that can be used in a simplicial branch and bound. Section 3 describes several ways to discover directions in which the objective function is monotonic. Section 4 presents a method for determining whether a facet of a simplex belongs to the border of the feasible set. Section 5 focuses on the algorithm under investigation and various ways to implement it. Section 6 reports on the behaviour of the algorithm for various instances. Our findings are reported in Sect. 7.

## 2 Notation and Mathematical Properties

### 2.1 Notation

We consider minimization of a continuously differentiable function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , over a feasible set  $\Delta$ , which is an  $n$ -simplex. An  $n$ -simplex can be defined by its vertices,  $v_0, \dots, v_n \in \mathbb{R}^n$ , a set of  $n + 1$  affine independent vectors. Namely,  $\Delta := \text{conv}(\mathcal{W})$  with set of vertex  $\mathcal{W} := \{v_0, \dots, v_n\}$ . The considered problem is

$$\min f(x), x \in \Delta. \quad (1)$$

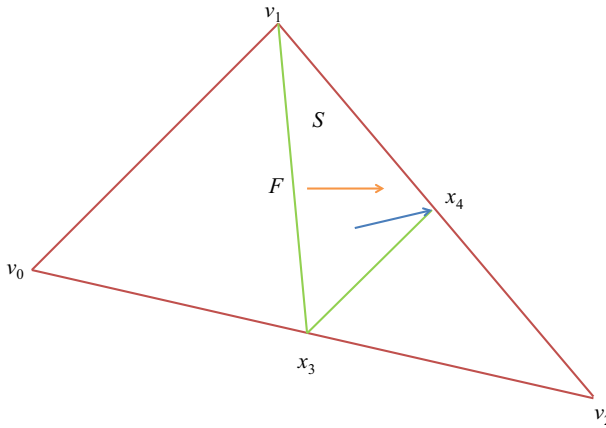
Actually, the problem can be posed more generically, where the number of vertices of  $\Delta$  is smaller, and thus the feasible set is not full-dimensional compared to the objective function [16]. The latter is relevant in mixture models, where quantities, probabilities or prices are normalized in such a way that the decision variables are on the standard simplex. We study a branch and bound algorithm to enclose all minimum points of  $f$  on  $\Delta$ . In contrast to the algorithms described in [11], the used partition sets are  $m$ -simplices  $S$ , where  $m \leq n$  can be not full-dimensional  $S := \text{conv}(\mathcal{V})$  with  $\mathcal{V}$  a set of  $m + 1 = |\mathcal{V}|$  vertices. The branch and bound algorithm works with a set  $\Lambda$  of partition sets, which as a whole includes all global minimum points.

The studied algorithm uses the longest edge bisection (LEB), where the longest edge  $(v, w)$  of a partition set  $S$  is bisected using mid-point  $x := \frac{v+w}{2}$  leading to two new simplices with vertex sets  $\mathcal{V} \setminus \{v\} \cup \{x\}$  and  $\mathcal{V} \setminus \{w\} \cup \{x\}$ . This is sketched in Fig. 1. Two Longest Edge bisections (LEB) are performed iteratively over the feasible area  $\Delta$  leading to new points  $x_3$  and  $x_4$  and a partition of 3 simplicial partition sets.

The algorithm also considers dimension reduction due to monotonicity considerations where the set  $\mathcal{V}$  of vertices of  $m$ -simplex  $S$  is reduced and  $S$  is replaced by one (or more) of its facets  $F := \text{conv}(\mathcal{V} \setminus \{v\})$  for some  $v \in \mathcal{V}$ . Notice that  $F$  is an  $(m - 1)$ -simplex. It may be clear that for  $m = 0$ , the 0-simplex  $S = \text{conv}(\{v\})$  is an individual point and has no faces. Its dimension cannot be reduced.

The centroid of  $m$ -simplex  $S = \text{conv}(\{v_0, v_1, \dots, v_m\})$  is given by  $c := \frac{1}{m+1} \sum_{j=0}^m v_j$  and the relative interior is defined by

$$\text{rint}(S) = \{x = \sum_j \lambda_j v_j, \lambda_j > 0, j = 0, \dots, m, \sum_{j=0}^m \lambda_j = 1\}. \quad (2)$$



**Fig. 1** Three partition sets generated by two longest edge bisections, using bisection points  $x_3$  and  $x_4$ . We focus on monotonic directions in  $S$

The relative boundary of a simplex  $S$  is defined by removing the relative interior from it. Considering a simplicial partition set  $S$ , we are interested in studying whether its (simplicial) facets  $F$  are border with respect to the feasible set  $\Delta$ .

**Definition 1** Given the feasible area as an  $n$ -simplex  $\Delta$ . An  $m$ -simplex  $S$  with  $m < n$  is called border with respect to  $\Delta$  if there exists an  $m$ -simplex face  $\varphi$  of  $\Delta$ , such that  $S \subseteq \varphi$ .

One of the main questions is how to determine whether a facet of a simplex is border in a numerically efficient way, see Sect. 4. Border facets and the concept of monotonicity are used to reject a simplex or to reduce its simplicial dimension.

To find monotonic directions, we will use the information of a so-called *enclosure*  $G$  of the gradient  $\nabla f(x) \subseteq G := [\underline{G}, \overline{G}]$ ,  $\forall x \in S$  where  $\underline{G}$  and  $\overline{G}$  denote the lower and upper bound of interval box  $G$ . The interval vector  $G$  can be calculated by automatic interval differentiation over the so-called *interval hull* of a simplex  $S$ . The interval hull, represented as  $\square S$ , is the smallest box which encloses  $S$  [4, 17]. Now consider a directional vector  $d$  as the difference of two points  $x, y \in S$ ,  $d = x - y$ . The corresponding directional derivative  $d^T \nabla f(x)$ ,  $x \in S$ , is included in the inner product

$$d^T G = \left[ \underline{d^T G}, \overline{d^T G} \right] = \left[ \sum_{i=1}^n \min\{d_i \underline{G}_i, d_i \overline{G}_i\}, \sum_{i=1}^n \max\{d_i \underline{G}_i, d_i \overline{G}_i\} \right]. \quad (3)$$

### 2.2 Mathematical Properties on Monotonicity

First, we discuss properties related to full-dimensional simplices, where  $m = n$  and then generalize the properties to the general case, where  $m \leq n$ . For the interior of a full-dimensional  $n$ -simplex, the monotonicity properties are the same as for any full-dimensional convex set:

**Proposition 1** *Let  $S \subset \text{rint}(\Delta)$  be an  $n$ -simplex with gradient enclosure  $G \supseteq \nabla f(x), \forall x \in S$ . If  $\exists i \in \{1, \dots, n\}$  with  $0 \notin G_i$ , then  $S$  does not contain a global minimum point of (1).*

For the proof, see Proposition 2 in [8].

**Proposition 2** *Let  $S \subseteq \Delta$  be an  $n$ -simplex with gradient enclosure  $G \supseteq \nabla f(x), \forall x \in S$ . If  $\exists i \in \{1, \dots, n\}$  with  $0 \notin G_i$  then  $\text{rint}(S)$  does not contain a global minimum point of (1).*

For the proof, see Proposition 2 in [8].

**Corollary 1** *Let  $S \subseteq \Delta$  be an  $n$ -simplex with corresponding gradient enclosure  $G (\supseteq \nabla f(x), \forall x \in S)$  as partition set in a branch and bound algorithm. If  $\exists i \in \{1, \dots, n\}$  with  $0 \notin G_i$  and  $S$  has no border facets, then  $S$  can be rejected.*

The argument is that the relative boundary of  $S$  may contain a global minimum point, but the same point is enclosed in the relative boundary of another partition set.

If an  $n$ -simplex  $S$  is not interior to  $\Delta$ , it can be replaced by its border facets. However, not all border facets are interesting partition sets. It is possible to show that the relative interior of a facet  $F$  cannot contain a minimum point. For this property, we could use the so-called normal vector  $p$  with respect to facet  $F$ . Let  $S = \text{conv}(\mathcal{V})$ ,  $v \in \mathcal{V}$ ,  $\mathcal{U} = \mathcal{V} \setminus \{v\}$  and facet  $F = \text{conv}(\mathcal{U})$ . Consider  $u \in \mathcal{U}$  and construct a set of vectors  $\mathcal{Y} := \{w - u, w \in \mathcal{U} \setminus \{u\}\}$  with corresponding matrix representation  $Y$ . The normal vector of  $F$  in the direction of  $v$  can be expressed by defining (see e.g. [3])

$$p := \left[ I - Y(Y^T Y)^{-1} Y^T \right] (v - u). \quad (4)$$

The projection procedure (4) is sketched in Fig. 2. Vector  $p$  points into the same half-space where  $S$  is located. From each point in  $\text{rint}(F)$  a direction  $d$  with  $p^T d > 0$  points into  $\text{rint}(S)$ . Considering the gradient enclosure  $G (\supseteq \nabla f(x), \forall x \in S)$  component-wise, we can come to several conclusions.

**Proposition 3** *Consider an  $n$ -simplex  $S = \text{conv}(\mathcal{V})$  with a facet  $F$  generated by removing vertex  $v$  from  $\mathcal{V}$  with normal vector  $p$  as in (4). If for one component  $i$  we have  $G_i p_i < 0$  (either  $\overline{G}_i < 0$  and  $p_i > 0$ , or  $\underline{G}_i > 0$  and  $p_i < 0$ ), then  $\text{rint}(F)$  cannot contain a minimum point of (1).*

**Proof** See Proposition 4.3 in [9]. □

In the B&B procedure, if the search space is a full-dimensional simplex, the partition sets are full-dimensional  $n$ -simplices, up to a dimension reduction occurs to its facets. Using the property of  $0 \notin G$ , if  $\text{rint}(S)$  can be removed, we can reduce it to its border facets. However, for lower dimension  $m$ -simplices,  $m < n$ , the condition  $0 \notin G$  is a necessary condition to have monotonicity, but not sufficient. Therefore, we have to focus on bounds of directional derivatives. The most general result for an  $m$ -simplex is the following.

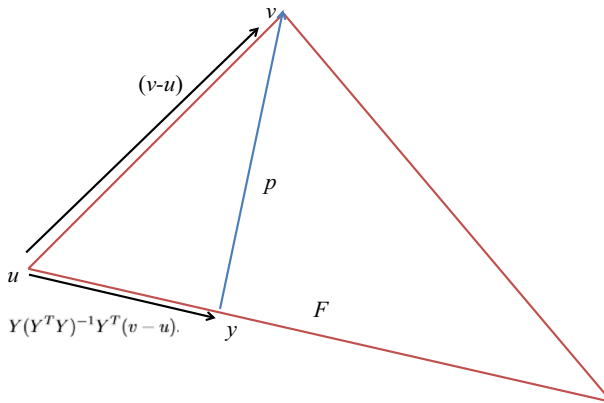


Fig. 2 Normal vector calculation via projection of an edge  $v - u$

**Proposition 4** Let  $S \subseteq \Delta$  be an  $m$ -simplex with  $m \leq n$  and gradient enclosure  $G \supseteq \nabla f(x), \forall x \in S$ . If  $\exists x, y \in S$ , such that direction  $d = x - y$  has corresponding directional derivative bounds (3) with  $0 \notin [d^T G, \overline{d^T G}]$  then  $\text{rint}(S)$  does not contain a global minimum point of (1).

**Proof** Consider  $z \in \text{rint}(S)$ . As  $z$  is in the relative interior, there exists a feasible direction  $d$  in which lower function values can be found, i.e.  $\exists \epsilon \in \mathbb{R}$  small enough, such that  $z + \epsilon d \in S$  and  $f(z + \epsilon d) < f(z)$ . So  $z$  cannot be a minimum point of  $f$ .  $\square$

The elaboration for the algorithm depends on the choice of the direction  $d$  and the way to compute it.

**Corollary 2** Let  $S \subseteq \Delta$  be an  $m$ -simplex as partition set in a branch and bound algorithm with corresponding gradient enclosure  $G \supseteq \nabla f(x), \forall x \in S$ . If the conditions of Proposition 4 apply and  $S$  has no border facets, then  $S$  can be rejected.

The argument is that the relative border of  $S$  may contain a global minimum point, but the same point is enclosed in the relative boundary of another partition set.

Given that the minimum is not in  $\text{rint}(S)$ , we have to decide which of the facets to focus on.

**Proposition 5** Given  $m$ -simplex  $S = \text{conv}(\mathcal{V})$ ,  $1 \leq m \leq n$  and a facet  $F$  generated by removing vertex  $v$  from  $\mathcal{V}$ , consider direction  $d = v - y$ , where  $y \in S$ . Consider the enclosure of the gradient  $G \supseteq \nabla f(x), \forall x \in S$ . If  $\underline{d^T G} > 0$ , i.e. the directional derivative in the direction  $d = v - y$  is positive, then the facet  $F$  contains all minimum points in  $S$ , i.e.  $\text{argmin}_{x \in S} f(x) \subseteq F$ .

**Proof** Consider an ordered vertex set  $\mathcal{V} = \{v_0, \dots, v_m\}$  and a facet  $F = \text{conv}(\{v_1, \dots, v_m\})$ , so the removed vertex is  $v_0$ . First, we show that  $\forall x \in S \setminus F, \exists \rho > 0$  such that  $z = x - \rho d \in F$ . We have  $x, y \in S$ , such that there exist vectors  $\lambda, \gamma$  with  $x = \sum_{j=0}^m \lambda_j v_j$  with  $\lambda_0 > 0$  and  $y = \sum_{j=0}^m \gamma_j v_j$ . Looking for the value  $\rho$ , such that  $z = x - \rho(v_0 - y) \in F$ , we have that  $\lambda_0 - \rho(1 - \gamma_0) = 0$ . Hence, taking  $\rho = \frac{\lambda_0}{1 - \gamma_0}$ , we

obtain  $z \in F$ . According to the mean value theorem, there exists a point  $\xi$  in between  $x$  and  $z$  such that

$$f(x) = f(z + \rho d) = f(z) + \rho d^T \nabla f(\xi) \geq f(z) + \rho \underline{d^T G} > f(z).$$

This means that each point  $x \in S \setminus F$  corresponds to a point  $z \in F$  with a lower function value.  $\square$

**Remark 1** Notice that if there are several facets fulfilling the condition of Proposition 5, then the consequence is also valid for the intersection of the facets, which is a face of  $S$ . So if Proposition 5 applies to  $F$  and  $H$ , then it also applies to  $F \cap H$ .

**Proposition 6** *Given  $m$ -simplex  $S$  fulfilling the conditions in Proposition 5. If  $F$  is a non-border facet, then global minimum points of (1) cannot be in  $\text{rint}(F)$ , i.e. global minimum points can only be found at the relative boundary of  $F$ .*

**Proof** According to Proposition 5, the minimum points of  $f$  over  $S$  can be found on facet  $F$ . We first show that  $x \in \text{rint}(F)$  does not contain global minimum points. Consider a point  $x \in \text{rint}(F)$ . Because  $F$  is non-border, there exists an  $\epsilon > 0$ , such that  $z(\epsilon) := x - \epsilon(v - y) \in \Delta$  and  $z(\epsilon) \notin S$ . As  $\underline{d^T G} > 0$  and  $f$  is continuously differentiable, there exists an  $\epsilon > 0$  (sufficiently small) such that for all  $\delta \in [0, \epsilon]$ , we have  $d^T \nabla f(z(\delta)) > 0$ . Therefore, following the mean value theorem, there exists a point  $\delta \in [0, \epsilon]$  such that  $f(z(\epsilon)) = f(x) - \epsilon(v - y)^T \nabla f(z(\delta)) < f(x)$ . This means that in a neighbourhood of  $x$  outside  $S$ , we can find a feasible point with a lower function value than  $x$ . The relative boundary of  $F$  could contain global minimum points of (1).  $\square$

**Remark 2** First, let us note that the boundary of  $F$  is shared by other partition sets, thus we can disregard simplex  $S$  in a branch and bound process.

Second, notice that there can be several facets fulfilling the condition of Proposition 5. However, we need only one non-border facet fulfilling the condition to remove simplex  $S$  from consideration.

**Corollary 3** *Let  $m$ -simplex  $S$  with centroid  $c$  fulfil the conditions of Proposition 5. Consider direction  $d = v - c$ . Consider the enclosure of the gradient  $G \supseteq \nabla f(x), \forall x \in S$ . If  $\underline{d^T G} > 0$ , then facet  $F$  contains all minimum points in  $S$ , i.e.  $\text{argmin}_{x \in S} f(x) \subseteq F$ . Moreover, if  $F$  is a non-border facet, then simplex  $S$  cannot contain the global minimum of (1).*

This follows from Propositions 5 and 6 by using the centroid  $y = c$ .

**Example 1** For the illustration of the concept, consider the simplices in Fig. 1. It shows three partition sets generated by bisection, using bisection points  $x_3$  and  $x_4$ . Consider  $S = \text{conv}(\mathcal{V})$  with  $\mathcal{V} = \{v_1, x_3, x_4\}$ , where we assume that the orange direction provides a monotonously increasing direction. According to Proposition 4, the interior of  $S$  does not contain a global minimum point. The blue direction provides an increasing monotonicity for facet  $F = \text{conv}(\mathcal{V} \setminus \{v\})$  with  $v = x_4$ . According to Proposition 5, facet  $F = \text{conv}(\{v_1, x_3\})$  contains all minimum points of  $S$ . Specifically, they are at

lower dimensional faces of  $F$  (the vertices), as  $\text{rint}(F)$  cannot contain minima either. Now, Proposition 6 states, that we even can remove  $S$ , as there is another partition set sharing  $F$ , that encloses all minimum points.

An alternative is to use the projection point  $y = (v - p) \in S$  of the normal vector  $p$  of (4) as illustrated in Fig. 2 for the choice of  $y$ . However, this is only correct if the simplex has angles of less than  $\frac{\pi}{2}$ . This is typically the case when we follow the longest edge bisection process.

**Corollary 4** *Given an  $m$ -simplex partition set  $S = \text{conv}(\mathcal{V})$  in the bisection and dimension reduction process and a facet  $F$  generated by removing vertex  $v$  from  $\mathcal{V}$  with normal vector  $p$  according to (4). Consider the enclosure of the gradient  $G \supseteq \nabla f(x), \forall x \in S$ . If  $\overline{p^T G} > 0$ , then  $F$  contains all minimum points over  $S$ . Moreover, if  $F$  is a non-border facet, then simplex  $S$  cannot contain the global minimum of (1).*

This follows from Propositions 5 and 6 by using the normal vector  $p$  in place of  $d = v - y$ .

**Proposition 7** *Given  $m$ -simplex  $S = \text{conv}(\mathcal{V}), 1 \leq m \leq n$ , consider a facet  $F$  generated by removing vertex  $v$  from  $\mathcal{V}$  and gradient enclosure  $G \supseteq \nabla f(x), \forall x \in S$ . If for direction  $d = v - y$  with  $y \in S$  we have  $\overline{d^T G} < 0$ , then  $\text{rint}(F) \subset S$  does not contain a minimum point of (1).*

**Proof** Following the proof of Proposition 5, considering that for any  $\xi \in S, \nabla f(\xi) \leq \overline{d^T G}$ , we have

$$f(x) = f(z + \rho d) = f(z) + \rho d^T \nabla f(\xi) \leq f(z) + \rho \overline{d^T G} < f(z)$$

and the result follows. □

### 3 Checking on Monotonicity

To prove that there exists a monotone direction in an  $n$ -simplex, it is sufficient to have  $0 \notin G$ . This is a necessary, but not sufficient condition for a lower dimensional  $m$ -simplex,  $m < n$ . To prove that such a direction exists, according to Proposition 4, we need to find a direction  $d = x - y$ , with  $x, y \in S$  corresponding to a positive lower bound of the directional derivative,  $\underline{d^T G} > 0$ , where

$$\underline{d^T G} = \sum_{i=1}^n \min\{d_i \underline{G}_i, d_i \overline{G}_i\}. \tag{5}$$

Note that it is enough to find a direction  $d$  with positive directional derivative bound, as  $-d$  will give a negative directional derivative. Finding such a direction can be done by searching for the steepest monotone direction  $\max_d \underline{d^T G}$  by an LP and check if it is positive. We reformulate the min part by introducing variable  $z_i = \min\{d_i \underline{G}_i, d_i \overline{G}_i\}$ .

In order to decrease the number of variables, we can fix  $x$  to the centroid in the directional vector, i.e.  $d = c - y$ , and solve

$$\begin{aligned}
 & \max \sum_{i=1}^n z_i \\
 \text{s. t. } & z_i \leq (c_i - y_i)\underline{G}_i, \quad i = 1, \dots, n \\
 & z_i \leq (c_i - y_i)\overline{G}_i, \quad i = 1, \dots, n \\
 & \sum_{j=1}^m \lambda_j = 1 \\
 & y = \sum_{j=0}^m \lambda_j v_j \\
 & \lambda_j \geq 0, \quad j = 0, \dots, m
 \end{aligned} \tag{6}$$

If there is no monotone direction, then  $y = c$  and  $\sum_{i=1}^m z_i = 0$ . If a monotone direction exists,  $y$  will lay on the relative boundary of  $S$  to maximize the directional derivative.

After proving a positive direction exists, one can validate Proposition 5 for a specific facet  $F$  of  $S$ . Consider the vertex set of  $F$  given as  $\{v_1, \dots, v_m\} := \mathcal{V} \setminus \{v_0\}$ . Focusing on the direction  $v_0 - y$  with  $y = \sum_{j=1}^m \lambda_j v_j$ , we can demonstrate there is a (maximum) positive directional derivative solving the LP

$$\begin{aligned}
 & \max \sum_{i=1}^n z_i \\
 \text{s. t. } & z_i \leq \underline{G}_i(v_{0i} - y_i), \quad i = 1, \dots, n \\
 & z_i \leq \overline{G}_i(v_{0i} - y_i), \quad i = 1, \dots, n \\
 & \sum_{j=1}^m \lambda_j = 1 \\
 & y = \sum_{j=1}^m \lambda_j v_j \\
 & \lambda_j \geq 0, \quad j = 1, \dots, m
 \end{aligned} \tag{7}$$

and checking the result is positive. It is the question whether it is worthwhile to solve (7) for each facet of a partition set. The focus is on having a direction  $d = v_0 - y$  where the function is monotonic and to consider the corresponding facet.

We can also search for a facet according to (7) over all facets simultaneously, solving an Integer Programming problem where a binary variable  $\delta_j$  selects the facet corresponding to the most positive directional derivative.

$$\max \sum_{i=1}^n z_i$$

$$\begin{aligned}
 \text{s. t. } & z_i \leq \underline{G}_i d_i, \quad i = 1, \dots, n \\
 & z_i \leq \overline{G}_i d_i, \quad i = 1, \dots, n \\
 & \sum_{j=0}^m \lambda_j = 1 \\
 & d = \sum_{j=0}^m \delta_j v_j - \sum_{j=0}^m \lambda_j v_j \\
 & \sum_{j=0}^m \delta_j = 1 \\
 & \lambda_j \geq 0, \quad j = 0, \dots, m \\
 & \delta_j \in \{0, 1\}, \quad j = 0, \dots, m
 \end{aligned} \tag{8}$$

Problem (8) can be used to replace LP problem (7) if such a monotonously increasing direction towards one of the vertices exists. If this is not the case, there may still exist an increasing direction according to (6).

The two steps, looking whether a monotone direction exists and identifying which facet contains all the minima (if any), can also be done in one step. Thus, instead of solving LP (6) and (7)  $m + 1$  times (or MILP (8)), we can solve directly one MILP as follows. Let  $d = x - y$ , where  $x, y \in S$ , so  $x = \sum_{j=0}^m \lambda_j v_j$  and  $y = \sum_{j=0}^m \mu_j v_j$ . Consider direction  $d = x - y = \sum_{j=0}^m (\lambda_j - \mu_j) v_j$

$$\begin{aligned}
 & \max \sum_{j=0}^m \delta_j \\
 \text{s. t. } & z_i \leq \underline{G}_i d_i, \quad i = 1, \dots, n \\
 & z_i \leq \overline{G}_i d_i, \quad i = 1, \dots, n \\
 & \sum_{j=0}^m \lambda_j = 1 \\
 & \sum_{j=0}^m \mu_j = 1 \\
 & d = \sum_{j=0}^m (\lambda_j - \mu_j) v_j \\
 & \sum_{i=1}^n z_i \geq \epsilon
 \end{aligned} \tag{9}$$

$$\begin{aligned} \mu_j &\geq \delta_j \quad \forall j = 0, \dots, m \\ \sum_{j=0}^m \delta_j &\leq 1. \\ \mu_j, \lambda_j &\in [0, 1], j = 0, \dots, m \\ \delta_j &\in \{0, 1\}, j = 0, \dots, m \end{aligned}$$

A solution with  $\mu_k = 1$  and  $\mu_j = 0, j \neq k$  represents a direction  $d$  pointing to vertex  $v_k$ . A solution with  $\lambda_k = 1$  and  $\lambda_j = 0, j \neq k$  represents a direction pointing from vertex  $v_k$ . We connect  $\mu_j$  with binary variables  $\delta_j$  such that  $\delta_j = 1$  implies  $\mu_j = 1$ . The inequality  $\sum_{i=1}^n z_i \geq \epsilon$  with  $\epsilon > 0$  assures  $d$  is a monotone direction. Moreover, it forces  $\mu \neq \lambda$ , because otherwise  $z = 0$  as well. If no monotone direction exists, (9) has no feasible solution.

The objective is to maximize the sum of  $\delta_j$ , meaning that we aim at finding a monotone direction with  $\mu_k = 1$  corresponding to  $\delta_k = 1$ . In this case, we know that facet  $F = \text{conv}(\mathcal{V} \setminus \{v_k\})$  contains all minima according to Proposition 5. If the objective is zero, there is no facet containing all the minima, i.e. there is no  $\delta_k = 1$ , but there is a monotone direction  $d$ .

### 4 Keeping Track of Border Facets

To know the border status of a given facet, we use a labelling system to find out which minimum dimensional face of  $\Delta$  the facet or simplex is included in. This is done by assigning to each face  $\varphi$  of feasible set  $\Delta = \text{conv}(\mathcal{W})$  a label  $\mathcal{B}(\varphi)$ , starting with the vertex faces labeled  $\mathcal{B}(v_j) = 0..010..0$  where the only 1 corresponds to bit  $j$ , for  $j = 0, \dots, n$ . For each face  $\varphi$ , which is a convex combination of vertices  $\mathcal{V} \subseteq \mathcal{W}$ , the corresponding bit-string  $\mathcal{B}(\varphi)$  has a value 1 for each vertex  $v \in \mathcal{V}$  in the same position as in  $\mathcal{B}(v)$ . For instance, in Fig. 3, the edge  $(v_1, v_2)$  has label 011, and the

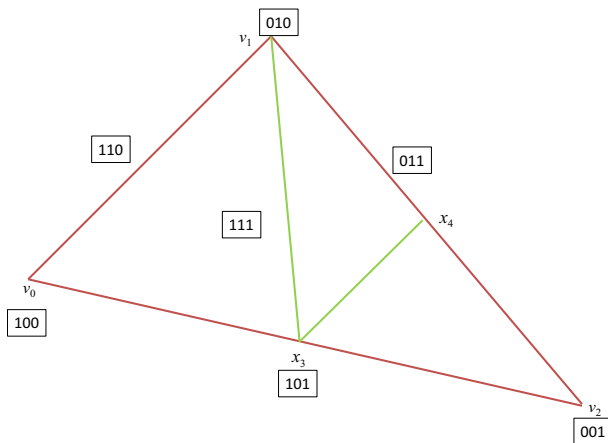


Fig. 3 Faces of the feasible set with bisection points  $x_3$  and  $x_4$

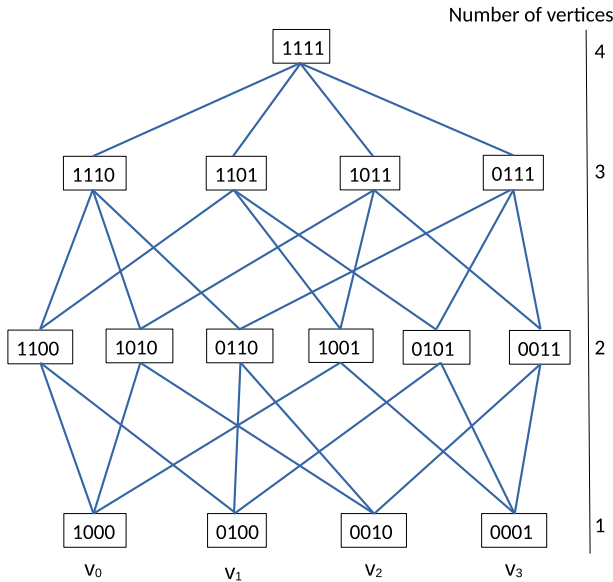


Fig. 4 Face graph for 4-vertex simplex. Binary labels in boxes for each face

simplex  $\Delta$  has label 111. In fact, for an  $m$ -simplex face  $\varphi$  of the feasible set, its label is given by the bitor of the label of all its vertices. The complete face graph is given in Fig. 4 for a 4-vertex simplex.

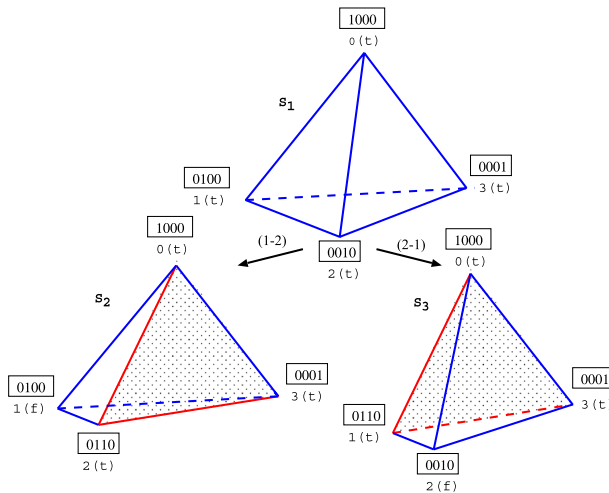
After bisecting the original set of vertices  $\mathcal{W}$ , we store the bisection points in set  $X$ . For instance in Fig. 3, we have  $X = \{v_0, v_1, v_2, x_3, x_4\}$ . We label all generated points  $x \in X$ , which serve as vertices for the partition sets. The label of point  $x$  is the same as the label of the minimum dimensional face  $\varphi$  of the feasible set  $x$  is in. For instance, the label of  $x_3$  is 101, the same as the label of face  $\text{conv}(v_0, v_2)$ . In the bisection, a new vertex  $x = \frac{v+w}{2}$  gets label  $\mathcal{B}(x) = \mathbf{BitOr}(\mathcal{B}(v), \mathcal{B}(w))$ .

Given an  $m$ -simplex  $S = \text{conv}(\mathcal{V})$ , the question is what is the label of the (smallest dimensional) face  $\varphi$  it is included in. This is determined by the label  $\mathcal{B}(\varphi) = \mathbf{BitOr}(\mathcal{B}(\mathcal{V}))$ , to be interpreted as a bitor on all its vertex labels. The number of ones of a bitstring  $\mathcal{B}(\varphi)$  is denoted by  $|\mathcal{B}(\varphi)|$ , giving the number of vertices of  $\varphi$ . According to Definition 1,  $m$ -simplex  $S$  is border if there exists an  $m$ -simplex face  $\varphi$  of the feasible set including  $S$  ( $m < n$ ).

**Proposition 8** Consider an  $m$ -simplex  $S = \text{conv}(\mathcal{V})$  with  $m < n$ . If  $|\mathbf{BitOr}(\mathcal{B}(\mathcal{V}))| = m + 1$ , then simplex  $S$  is border.

**Proof** Consider the face  $\varphi$ , which is the minimal dimensional face containing  $S$ , i.e. label  $\mathcal{B}(\varphi) = \mathbf{BitOr}(\mathcal{B}(\mathcal{V}))$ . As  $|\mathbf{BitOr}(\mathcal{B}(\mathcal{V}))| = m + 1$ , we have  $|\mathcal{B}(\varphi)| = m + 1$ , thus  $\varphi$  is an  $m$ -simplex. Therefore,  $S$  is enclosed by an  $m$ -simplex face of the feasible set and thus is border.  $\square$

For example in Fig. 3, the edge  $(x_3, v_2)$  is border, as  $|\mathbf{BitOr}(\mathcal{B}(\{x_3, v_2\}))| = |\mathbf{BitOr}(101, 001)| = |101| = 2$  corresponding to face  $\text{conv}(\{v_0, v_2\})$ . In contrast,



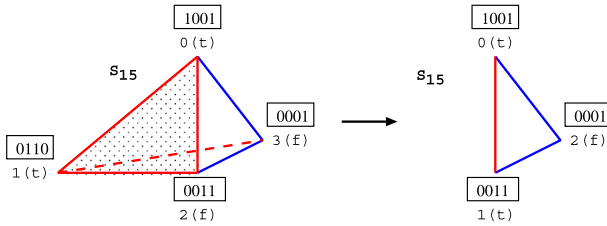
**Fig. 5** Part of the search tree where LEB splits simplex  $S$  into two new simplices. Blue edges are border and red ones are non-border. The border status of vertex  $v_j$  and corresponding facet  $F_j$  is given by  $j(t)$  if  $\beta(F_{v_j}) = true$  and  $j(f)$  if  $\beta(F_{v_j}) = false$ . The dotted facet is the new facet generated by the bisection

edge  $(x_3, x_4)$  is not border, because  $|\mathbf{BitOr}(\mathcal{B}(\{x_3, x_4\}))| = |111| = 3 \neq 2$ . In fact, the minimum dimensional face it is included in, is  $\Delta$  itself.

As we have seen in the theoretical results, for monotonicity testing the border certification of facets is of importance. One way to do so is to follow Proposition 8. One of our research questions is whether it might be more efficient in an algorithm to store the border status of the facets of a simplex in a boolean vector  $\beta$  indicating for each facet whether it is border or not. Then, we can transfer the status  $\beta$  to facets of newly generated simplicial subsets instead of evaluating the border status  $|\mathbf{BitOr}(\mathcal{B}(\mathcal{V}))| = m + 1$  for all facets. Therefore, we introduce a border status for each facet of  $m$ -simplicial subset  $S$ . Each facet  $F$  corresponds to a vertex  $v \in \mathcal{V}$  which is dropped from the vertex set  $S$  to obtain  $F$ . Let  $F_v$  denote  $\text{conv}(\mathcal{V} \setminus \{v\})$  with its marker  $\beta(F_v) \in \{true, false\}$ . Storing this information with subset  $S$  requires storing a binary vector  $\beta = (\beta_0, \dots, \beta_m) = (\beta(F_{v_0}), \dots, \beta(F_{v_m}))$  ordered in the same way as the vertex set and the corresponding facets.

### 4.1 Setting the Border Status for Facets in Longest Edge Bisection Refinement

Consider  $m$ -simplex  $S$  with ordered vertex set  $\{v_0, \dots, v_i, \dots, v_j, \dots, v_m\}$  and the corresponding facet border status vector  $(\beta_0, \dots, \beta_i, \dots, \beta_j, \dots, \beta_m)$ . Longest Edge Bisection (LEB) along edge  $(v_i, v_j)$  generates the new point  $x = \frac{v_i + v_j}{2}$  and two new partition sets  $S_1 = \text{conv}(\{v_0, \dots, x, \dots, v_j, \dots, v_m\})$  and  $S_2 = \text{conv}(\{v_0, \dots, v_i, \dots, x, \dots, v_m\})$ . Both new simplices inherit the border status  $\beta$  from parent  $S$ , but we set the border status of the new facet to false: for  $S_1$ , we set  $\beta_j := false$  and for  $S_2$  we put  $\beta_i := false$ . So, apart from the newly generated shared facet, all other facets inherit the border status from the bisected simplex. Fig-



**Fig. 6** Reduction of  $S_{15}$  by removing vertex  $v_1$  (denoted by  $1(t)$  on the left). Before reduction,  $S_{15}$  is included in face  $\varphi$  with label  $\mathbf{BitOr}(\mathcal{B}(\mathcal{V})) = 1111$ . So  $S_{15}$  contains interior points in  $\Delta$ . After reduction,  $\mathbf{BitOr}(\mathcal{B}(\mathcal{V} \setminus \{v_1\})) = 1011$ . The border status of facet  $F_{v_0}$  is true and stays the same. Vertex  $v_2$  ( $v_3$ ) becomes vertex  $v_1$  ( $v_2$ ). Because the border status of their facets is false, the new status requires recalculation

ure 5 shows the first bisection of the feasible set being 3-simplex  $\Delta$  with the border status of its facets.

### 4.2 Setting Border Status for Facets in Simplicial Dimension Reduction

Consider  $m$ -simplicial subset  $S$  with ordered vertex set  $\{v_0, v_1, \dots, v_m\}$  and corresponding border status  $(\beta_0, \beta_1, \dots, \beta_m)$ . Let us focus on dropping one of the vertices and reducing the simplex to the corresponding border facet. Without loss of generality consider dropping  $v_1$ . The remaining facet  $F_{v_1} = \text{conv}(\{v_0, v_2, \dots, v_m\})$  is border when  $\beta_1 = \text{true}$ . Otherwise  $F_{v_1}$  is not border. What will be the new border status  $(\beta_0, \beta_2, \dots, \beta_m)$ ?

**Proposition 9** *Given  $m$ -simplex  $S = \text{conv}(\mathcal{V})$  and border facets  $F = \text{conv}(\mathcal{V} \setminus \{v\})$  and  $E = \text{conv}(\mathcal{V} \setminus \{w\})$ ,  $v \neq w$ . The intersection  $F \cap E = \text{conv}(\mathcal{V} \setminus \{v, w\})$  is border.*

**Proof** According to Proposition 8, the including faces  $\varphi_F, \varphi_E$  corresponding to  $F$  and  $E$  have labels such that  $|\mathcal{B}(\varphi_F)| = |\mathcal{B}(\varphi_E)| = m$ , because  $F$  and  $E$  are border  $(m - 1)$ -simplices. The intersection  $F \cap E$  is an  $(m - 2)$ -simplex, as we removed two vertices from  $\mathcal{V}$ .

Assume that  $F \cap E$  is not border, then  $|\mathcal{B}(\varphi_{F \cap E})| > m - 1$ . As  $\varphi_{F \cap E} \subset \varphi_F$ , we have that  $|\mathcal{B}(\varphi_{F \cap E})| \leq m$  bits. Consequently,  $|\mathcal{B}(\varphi_{F \cap E})| = m$ .

As  $(F \cap E) \cup \{w\} = F$ , also  $\mathbf{BitOr}(\mathcal{B}(\varphi_{F \cap E}), \mathcal{B}(w)) = \mathcal{B}(\varphi_F)$ . From  $|\mathcal{B}(\varphi_{F \cap E})| = |\mathcal{B}(\varphi_F)| = m$  follows that  $\mathcal{B}(\varphi_{F \cap E}) = \mathcal{B}(\varphi_F)$ . Similarly, we obtain  $\mathcal{B}(\varphi_{F \cap E}) = \mathcal{B}(\varphi_E)$ . Therefore,  $\mathcal{B}(\varphi_F) = \mathcal{B}(\varphi_E)$  and  $\varphi_E = \varphi_F$ . Two different border facets  $F$  and  $E$  cannot be included in the same  $m$ -face of the feasible set. Thus,  $F \cap E$  must be border. □

The consequence of Proposition 9 is that a border status  $\beta(F_v) = \text{true}$  remains true after reduction and  $\beta(F_v) = \text{false}$  requires recalculation using Proposition 8. Figure 6 shows an illustrative example.

**Algorithm 1** SimplicialBranch-and-Bound( $f, \Delta, \alpha$ )**Require:**

$f$ : the  $n$  dimensional continuously differentiable objective function.  
 $\Delta$ : feasible set  $\Delta = \text{conv}(\{v_0, \dots, v_n\})$   
 $\alpha$ : accuracy

- 1: Store  $\Delta$  in  $\Lambda$  ▷ Set of partition sets
- 2:  $X = \{v_0, \dots, v_n\}$  ▷ List of points used as vertices
- 3: Evaluate and store  $f(x)$  and  $\mathcal{B}(x)$  for all  $x \in X$
- 4:  $\bar{f} = \min_{x \in X} f(x)$  ▷ Incumbent value
- 5:  $\underline{f} = \underline{f}(\Delta)$
- 6: **while**  $\bar{f} - \underline{f} > \alpha$  **do**
- 7:   Extract  $S$  from  $\Lambda$  ▷ Retrieve  $S$  from  $\Lambda$  with lowest value for  $\underline{f}(S)$
- 8:   Evaluate gradient enclosure  $G$  of  $S$
- 9:   **if**  $0 \in G$  **then** Divide = true ▷  $0 \notin G$  is a necessary but not sufficient condition
- 10:   **else** ▷ Trying to exploit monotonicity
- 11:     Divide = MonotonicityTest( $f, S, G, \Lambda$ ) ▷ Call Algorithm 2, 3, or 4
- 12:   **if** Divide **then**
- 13:      $\{S_1, S_2\} = \text{LEB}(S)$  at midpoint  $x$  ▷ Longest Edge Bisection
- 14:     **if**  $x \notin X$  **then**
- 15:       Evaluate and store  $f(x)$  and  $\mathcal{B}(x)$  and add  $x$  to  $X$
- 16:       **if**  $\underline{f}(x) < \bar{f}$  **then**
- 17:          $\bar{f} = \underline{f}(x)$
- 18:          $\Lambda = \text{CutOff}(\Lambda)$  ▷ Remove subsets  $S \in \Lambda$  with  $\underline{f}(S) > \bar{f}$
- 19:       **for** each new subset  $S_j$  **do**
- 20:         Evaluate  $\underline{f}(S_j)$
- 21:         **if**  $\underline{f}(S_j) \leq \bar{f}$  **then** ▷ RangeUp test
- 22:         Store  $S_j$  in  $\Lambda$  ▷ Store  $S_j$  and its bounds in  $\Lambda$
- 23:      $\underline{f} = \min_{S \in \Lambda} \underline{f}(S)$
- 24: **return**  $\cup_{S \in \Lambda} S$  and  $[\underline{f}, \bar{f}]$

**5 A Simplicial Branch and Bound Algorithm Exploiting Monotonicity**

The former section described several ways to deal with the monotonicity and border certification. Basically, one can evaluate the border status of facets, or store their status within the simplex object. This section describes the branch and bound algorithm that exploits the theoretical monotonicity results. It is based on a traditional scheme where a set  $\Lambda$  of subsets includes all global minimum points. The stopping criterion is taken as the classical concept of [6] on the gap between the incumbent upper bound  $\bar{f}$  of the best point found so far and a lower bound  $\underline{f}$  of the lowest bound among all subsets with an accuracy  $\alpha$ . This concept corresponds to guarantee to find a point which is not worse than an accuracy  $\alpha$  from the global minimum function value. Of course one can consider other criteria based on the distance from the minimum points or the relative function value. However, the focus is on investigating the impact of the different monotonicity rules.

There are several ways to define the lower bound  $\underline{f}$  over simplicial subsets [9]. We evaluate several ways to certify the monotonicity of the objective over a subset. Algorithm 2 describes the monotonicity test using center to vertex (C) and vertex to vertex (V) directions. In line 3, we generate the directions to check for all vertices: i) the center to vertex directions and ii) all vertex to vertex directions without repetition.

For each direction, we check its monotonicity, and we either discard the simplex (line 7, return false), reduce to some of its facets (line 13 or 17, return false) or return true to divide simplex  $S$ .

---

**Algorithm 2** MonotonicityTest( $f, S, G, \Lambda$ )

---

```

1:  $m$  is the dimension of  $S$ , i.e.  $S = \text{conv}(\mathcal{V})$ ,  $m = |\mathcal{V}|$ 
2: if  $m == n$  and  $S$  has no border facets then return false ▷ Corollary 1
3: Generate a set of directions  $d_v$  towards each vertex  $v$  ▷  $d_v$  can be  $v - c$ 
4:  $\text{MinInFacet} = \text{MinNotInF} = \text{Unknown} = \emptyset$  ▷ Sets to categorize facets
5: for all  $v \in \mathcal{V}$  and for all directions  $d_v$  do
6:   Compute  $d_v^T G$  for  $d_v = (v - y)$  ▷ Directional derivative for  $(v - y)$ 
7:   if  $d_v^T G > 0$  and  $F_v$  is not border then return false ▷ Prop. 6, Remark 2
8:   if  $d_v^T G > 0$  then  $\text{MinInFacet} = \text{MinInFacet} \cup \{F_v\}$  ▷ Prop. 5
   ▷  $\overline{\text{MinInFacet}}$  only contains border facets (see line 7).
9:   if  $d_v^T G < 0$  then  $\text{MinNotInF} = \text{MinNotInF} \cup \{F_v\}$  ▷ Prop. 7
10:  if  $0 \in d_v^T G$  then  $\text{Unknown} = \text{Unknown} \cup \{F_v\}$ 
11: if  $|\text{MinInFacet}| > 0$  then
12:   Create face  $F = \cap_{F_v \in \text{MinInFacet}} F_v$ 
13:   if  $f(F) \leq \bar{f}$  then Store  $F$  in  $\Lambda$  ▷  $S$  is reduced to  $F$ . Remark 1
14:   return false ▷ Remove  $S$ 
   ▷ At this point, there is no  $v \in \mathcal{V}$  with  $d_v^T G > 0$ .
   ▷  $m < n$  and  $\exists v$  with  $F_v \in \text{MinNotInF} \implies f$  is monotonic on  $S$ .
15: if ( $m < n$  and  $\exists v$  with  $F_v \in \text{MinNotInF}$ ) or  $m == n$ ) then ▷  $f$  is monotonic on  $S$ 
16:   for all  $v$  with  $(F_v \in \text{Unknown and } F_v \text{ is border})$  do
17:     if  $f(F_v) \leq \bar{f}$  then Store  $F_v$  in  $\Lambda$  ▷ Store unknown border facets
18:   return false ▷ Remove interior or the whole by Cor. 2, Prop. 2 or 7
19: return true ▷  $m < n$  and no monotone direction, divide  $S$ 

```

---

Algorithm 3 describes the monotonicity test using LP. This can be used instead of Algorithm 2, but it is better to use it when the CV based directions did not provide a monotone direction. Lines 6-14 seek for a monotone direction towards a vertex, such that we can reduce to as few facets as possible. These lines can be changed to solving MIP(8), and we will refer to exactly that by monotonicity test using MIP(8).

In Algorithm 4, the most compact version of the monotonicity test is described, where MIP(9) is solved, and from its solution (or lack of it), we know if the simplex can be removed, reduced, or should be divided.

## 6 Numerical Illustration

Algorithms were run on an Intel(R) Core(TM) i7-4770K CPU and 16GB of RAM running Fedora 37 Linux distribution. The algorithm was coded with g++ (gcc version 11.2.1-4) and it uses Kv–0.4.55 for Interval Arithmetic and Automatic Differentiation. Kv uses Boost libraries. Algorithms were compiled with `-O3 -DNDEBUG -DKV_FASTROUND` options. For the Linear Programming, we use PNL 1.11.0 compiled with `-DCMAKE_BUILD_TYPE = Release` and `-DWITH_MPI = OFF`, as a C++ wrapper to Lpsolve 5.5.2.0-29. The default Lp\_solve parameters are used in

**Algorithm 3** MonotonicityTest-LP( $f, S, G, \Lambda$ )

---

```

1:  $S = \text{conv}(\mathcal{V}), m = |\mathcal{V}|, \text{MinInFacet} = \text{MinNotInF} = \text{Unknown} = \emptyset$ 
2: if  $m == n$  and  $S$  has no border facets then return false ▷ Corollary 1
3: if  $m < n$  then
4:   Solve LP (6) finding  $d = c - y$ 
5:   if  $0 \in d^T G$  then return true ▷  $f$  is not monotonic on  $S$ 
6: for all  $v \in \mathcal{V}$  do
7:   Solve LP (7) for  $F_v$  finding  $d_v = v - y$ 
8:   if  $d_v^T G > 0$  then
9:     if  $\overline{F_v}$  is not border then return false ▷ Prop. 6
10:     $\text{MinInFacet} = \text{MinInFacet} \cup \{F_v\}$  ▷ Prop. 5
11:    if  $d_v^T \overline{G} < 0$  then  $\text{MinNotInF} = \text{MinNotInF} \cup \{F_v\}$  ▷ Prop. 7
12:    if  $0 \in d_v^T G$  then  $\text{Unknown} = \text{Unknown} \cup \{F_v\}$ 
13: if  $|\text{MinInFacet}| > 0$  then
14:   Create a face  $F = \cap_{F_v \in \text{MinInFacet}} F_v$ 
15:   if  $\underline{f}(F) \leq \overline{f}$  then Store  $F$  in  $\Lambda$ 
16:   return false ▷  $S$  is reduced to  $F$ . Remark 1
17: for all border  $F_v \in \text{Unknown}$  do
18:   if  $\underline{f}(F_v) \leq \overline{f}$  then Store  $F_v$  in  $\Lambda$  ▷ Store unknown border facets
19:   return false ▷ Remove interior, Prop. 7
20: return true

```

---

**Algorithm 4** MonotonicityTest-MIP( $f, S, G, \Lambda$ )

---

```

1:  $S = \text{conv}(\mathcal{V}), m = |\mathcal{V}|$ 
2: if  $m == n$  and  $S$  has no border facets then return false ▷ Corollary 1
3: Solve MIP (9) finding direction  $d = (x - y)$  or  $d_v = (v - y)$ 
4: if MIP (9) is infeasible then return true ▷ No monotonic direction exists
5: if the objective of MIP (9) is 1 with  $\delta_k = 1$  then ▷ Prop. 5
6:   if  $F_{v_k}$  is border and  $\underline{f}(F_{v_k}) \leq \overline{f}$  then ▷ Monotonic direction is to  $v_k$ 
7:   Store  $F_{v_k}$  in  $\Lambda$  ▷  $S$  is reduced to  $F_{v_k}$ 
8:   return false
   ▷  $f$  is monotonic in  $S$ , but no monotonic direction from any vertex exists
9: for all border  $F_v$  do
10:  if  $\underline{f}(F_v) \leq \overline{f}$  then Store  $F_v$  in  $\Lambda$  ▷ Store unknown border facets
11: return false ▷ Remove interior, Prop. 7

```

---

PNL. We also use the **IBM ILOG CPLEX Studio 22.1.0** for LP and MIP models. All results were obtained with a termination accuracy  $\alpha = 10^{-6}$ .

In Sect. 6.1, we first use a small instance to illustrate why the use of a simplicial B&B (sBB) is sometimes much more efficient than using standard Interval based B&B (iBB) with linear constraints. Then in Sect. 6.2, we compare variants of the simplicial algorithm on test instances introduced in [8].

In all experiments, the directions from the centre to the vertices (C) are checked first. In case a border facet with a minimum has not been found, vertex to vertex directions (V) are checked. As soon as a direction with a positive directional derivative is found, the search is stopped.

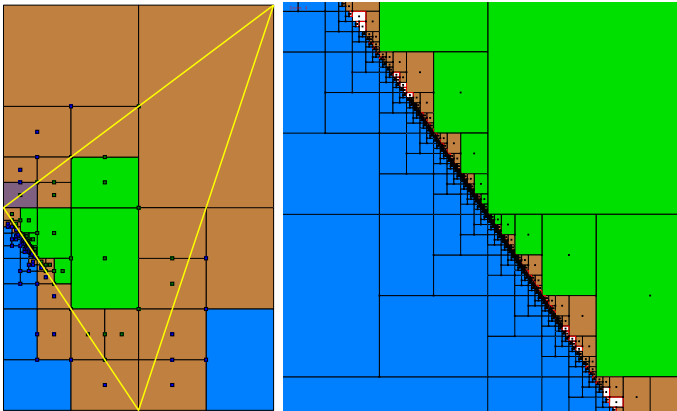


Fig. 7 Graphical output of iBB on instance Ex6-B. Right graph is a zoom over the minimum

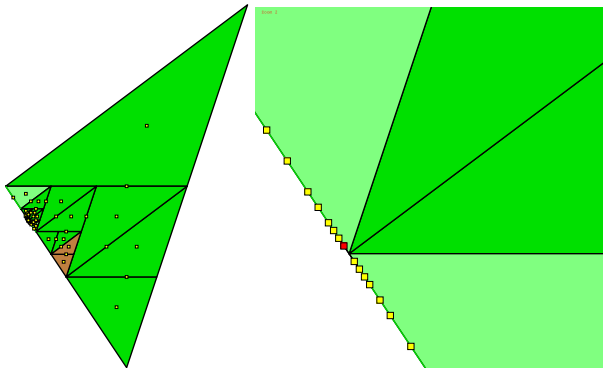
### 6.1 Small Illustrative Instance sBB versus iBB

Interval branch and bound has a long tradition and knows many implementations with acceleration devices [2, 12]. Several variants exist that can handle linear constraints. In this experiment, we take a small instance from [10] where a quadratic convex function is considered over a simplicial feasible set. The vertex structure of the feasible set is simply translated to a set of linear inequalities for iBB. In instance Ex6-B (see Appendix A.2.1), we define the constraints such that the minimum point lies on the boundary of the feasible set. In instance Ex6-In (see Appendix A.2.1), the minimum point is interior.

The idea of the design of experiments is to have an instance as small as possible to focus on the difference of having an interior minimum in a full dimensional case and of having the minimum on the boundary. For the interior optimum, we expect an established iBB to be far more efficient. However, we would like to observe what happens if the optimum is at the boundary, like in the instances we will use in Sect. 6.2. We compare the behavior of a standard iBB using linear constraints with the simplicial sBB which has been presented.

Both iBB and sBB, use the same inclusion information in the experiments. Interval Arithmetic uses boxes (interval vectors). Therefore, the interval hull of a simplex  $S$  ( $\square S$ ) implies an overestimation of the area of  $S$ . The iBB has to deal with linear constraints. Linear constraints derived from a set of vertices may have errors in a computer due to rounding. So, an interval is considered feasible if the enclosure of all constraints is negative or smaller than a given precision. For the precision, we use  $10^{-14}$ .

In this comparison, we use the centred form, which is the interval extension of the first-order Taylor expansion, using the centre of the set (+CFc), to try to improve objective function bounds that follow from direct use of Interval Arithmetic, see [8]. Algorithm sBB applies a monotonicity test based on C and V directional derivatives. Bounds of the derivatives obtained by Interval Arithmetic are component-wise. Therefore, they can be used directly on boxes in iBB.



**Fig. 8** Graphical output of sBB on instance Ex6-B. Right graph is a zoom over the minimum

**Table 1** Colors used in the Figures

	Color	Meaning
Set	Dark green	Rejected by monotonicity test
	Light green	Reduced to border facet(s)
	Brown	Rejected by RangeUp test: $\underline{f}(S) > \tilde{f}$ =incumbent $\min f(v)$
	Purple	Rejected by CutOff test: RangeUp test on stored sets
	Blue	Infeasible set
	White	Sets in $\Delta$
Evaluated point	Yellow	Used in sBB
	Blue	iBB infeasible
	DarkGreen	iBB feasible
	Red	Last incumbent
L. Const.	Yellow	Linear constraint

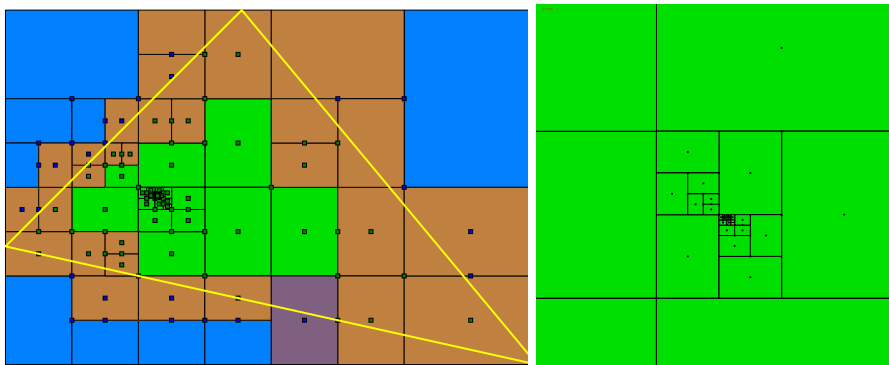
In Figs. 7 and 8, we can see the generated sets for Ex6-B using iBB and sBB. It is clear that the number of sets generated by sBB is much smaller. Table 1 describes the meaning of the used colors in the figures. For a more comprehensive comparison Table 2 provides the numerical results of iBB and sBB for this small illustrative instance.

Numerical results for Ex6-B, the test instance with the minimum at the boundary of the feasible set, show that simplex reduction in sBB provides a large advantage compared to the iBB results. At the final stages of the running algorithms, iBB must divide and check the feasibility of a lot of boxes near the global minimum while sBB is just dividing the reduced simplices (edges).

Numerical results for Ex6-In, the test instance with an interior minimum, show that somewhat more simplex than box evaluations are required. The main reason is that to reach the desired accuracy, a box in iBB and a simplex  $S$  in sBB (with bounds determined by enclosure  $\square S$ ), sBB requires more longest edge divisions than iBB and that number increases with the dimension. Another division method and devising

**Table 2** Comparing Interval versus Simplicial B&B for Ex6

Indicator	iBB-Ex6-B	sBB-Ex6-B	iBB-Ex6-In	sBB-Ex6-In
Time	0.007s	0.004s	0.005s	0.005s
Evaluated sets	2687	67	163	215
Infeasible boxes	765	–	9	–
Undetermined boxes	1802	–	68	–
Evaluated points	1916	67	150	215
Infeasible points	1221	–	28	–
Lower bound improvements	1906	64	145	212
Incumbent improvements	6	18	9	12
Monotonic sets	1898	48	105	97
Reduced sets	0	4	0	4
Rejected by rangeUp/CutOff	249	2	31	9
Rejected by monotonicity	30	28	41	92



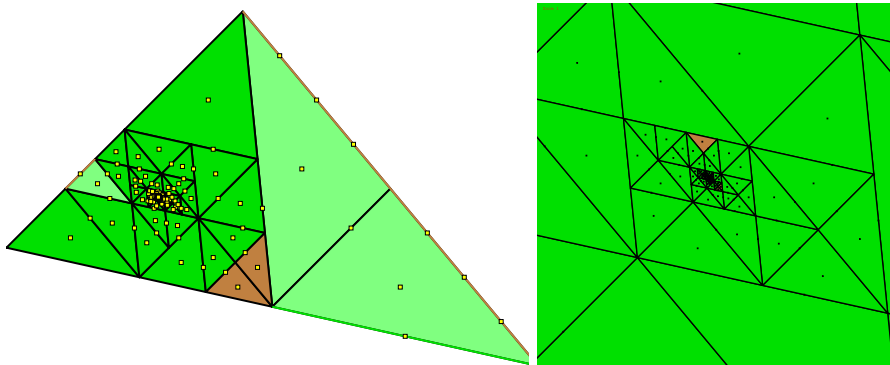
**Fig. 9** Graphical output of iBB on instance Ex6-In. Right graph is a zoom over the minimum

specific and sharper bounding for simplices could balance the scales. See Figs. 9 and 10 for the graphical outputs.

### 6.2 Comparing Variants of the Monotonicity Test in the Simplicial B&B

The implemented algorithm uses the centred form based on the highest vertex as the center for the computation of the lower bound, denoted by +CFvs, that was introduced in [8]. Several variants of Algorithm 2 are described and tested. Several directions  $(v - y)$  can be used. Additionally, the best direction can be returned by solving LP (7) in Algorithm 3 or a MIP in Algorithm 4. The symbols used for the different evaluated directional derivatives and models for sBB are given in Table 3.

For CVLP and CVMIP, LP and MIP models are evaluated only if CV did not find a positive directional derivative. This is done to reduce the number of LP and MIP models to be solved due to their high computational time. To reduce the computational time,



**Fig. 10** Graphical output of sBB on instance Ex6-In. Right graph is a zoom over the minimum

**Table 3** Notation for generated and evaluated directions

Notation	Directional vector and models
CV	$C:(v_j - c) + V:(v_j - v_\ell), \ell \neq j$
CVLP	CV + LP(6) + LP(7)
CVMIP8	CV + LP(6) + MIP(8)
CVMIP9	CV + MIP(9)

we sort the facets by having  $\min\{d_v^T G, \overline{d_v^T G}\}$  first, during the CV directions search. In case of a successful LP(6) search, LP(7) visits the facets in the predefined order and stops as soon as a positive directional derivative is found. So, some facets might not be checked. If LP(6) is infeasible, the simplex is divided and LP(7) or MIP(8) are not evaluated. If LP(7) or MIP(8) do not find a monotone directional derivative, then  $d_v^T G = 0$ .

LP(6) and LP(7) were solved using the PNL library and CPLEX in CVLP. Data in the LP model was reused as much as possible in PNL, but models are built each time and data is not reused in CPLEX for all models. The time limit for each model was set to 1 s and verbose was off, for both PNL (lp\_solve) and CPLEX.

We use the same set of test problems as in [8], given in Table 4. Traditional test functions are solved over a simplicial feasible set. The feasible simplex is determined such that the global optimizer is at the centre of its most left facet, and its size is half of the enclosing bounding box.

For instance, problem GP2, with enclosing box  $([-2, 2], [-2, 2])$  and minimum at  $(0, -1)$ , would have a corresponding feasible set  $\Delta = \text{conv}(\{(-0.5, -2), (1.5, -2), (0.5, 0.0)\})$ . However, in order to avoid an early hit of a global minimum point due to LEB and the clustering effect [7], the instances were adapted moving the simplicial feasible area a bit, such that the minimum point does not coincide with the centroid of the facet  $F$  it is included in. The simplex was shifted such that the minimum gets closer to a vertex  $v$  of  $F$  with 0.1 times the distance between the centroid of  $F$  and vertex  $v$ . For instance, problem GP2, with simplicial feasible area  $\Delta = \text{conv}(\{(-0.5, -2), (1.5, -2), (0.5, 0.0)\})$  and minimum at  $(0, -1)$ , is solved for the feasible set  $\Delta = \text{conv}(\{(-0.55, -2.1), (1.45, -2.1), (0.45, 0.1)\})$ . On one hand,

**Table 4** Test problems.

Instance	Description	Dimension $n$
DP7	Dixon-Price	7*
G7	Griewank	7*
GP2	Goldstein-Price	2
H3	Hartmann 3	3
H4	Hartmann 4	4
H6	Hartmann 6	6
L8	Levy	8*
MC2	McCormick	2
MCH5	Michalewicz	5*
RB7	Rosenbrock	7*
S4	Shekel 10	4
SCH2	Schubert	2
SHCB2	Six Hump Camel Back	2
ST5	Styblinski-Tang	5*
THCB2	Three Hump Camel Back	2
UPQP	Unconstrained Portfolio QP	4

**Table 5** Meaning of headers in the following tables

Header	Meaning
Ins	Problem instance
Time	Execution time. Differences smaller than 0.1s are not significant
ES	Number of evaluated simplices
N/N2	Number of simplices/edges checked for monotonicity
OkCV	Number of simplices with monotonic C and/or V directional derivatives
NSLP	Number of simplices checked by LP(6) or (7)
OkLP	Number of successful LP (6) or (7)
NMIP	Number of solved MIP (8) or (9) models
OkMIP	Number of successful MIP (8) or (9) models
Div	Reduced simplices with $0 \notin G$ but no monotone direction found, thus divided

the global minimum is not hit by one of the newly generated points and delays successful RangeUp and Cutoff tests. On the other hand, avoiding the clustering effect reduces the number of evaluated simplices. Results for instances with a shifted centroid are given in Tables 6, 7, 8, 9 and 10. The meaning of the symbols used in the tables to present the indicator values are given in Table 5.

The numerical results illustrate the usual explosion of the number of sub-simplices as the dimension increases in spatial branch and bound with few accelerating devices. Specifically, we are using just monotonicity (Alg. 1, line 11), RangeUp (Alg. 1, line 21) and Cutoff (Alg. 1, line 18) tests. Another characteristic is that the number of simplices where  $0 \notin G$ , the ratio (N/N2) is small compared to the number of evaluated

**Table 6** Results for the CV variant of the sBB algorithm

Ins	Time	ES	N/N2	OkCV	Div
MC2	0.01s	410	43/13	23	6
SHCB2	0.02s	527	37/9	24	5
THCB2	0.02s	538	30/8	16	5
SCH2	0.03	767	5/0	3	0
H3	0.03s	1689	287/0	182	69
S4	0.06s	4568	755/0	291	297
GP2	0.03s	7204	19/2	7	2
H4	0.27s	28,732	8020/0	3465	3584
MCH5	0.93s	63,869	12,588/0	4850	6673
G7	3.26s	278,326	816/0	1	0
RB7	4.19s	666,437	21,978/0	1522	0
ST5	5.83s	1,081,443	373,285/0	144,405	197,453
L8	27.79s	1,168,435	9904/0	45	0
H6	39.81s	2,918,326	1,681,477/0	664,752	969,207
DP7	34.25s	6,250,171	58,569/0	3303	0

simplices (ES) for most cases. This is intrinsic to the problem at hand, but it is also due to the overestimation in using interval hull  $\square S$  to get bounds of  $f(S)$  and  $G$  using Interval Arithmetic and Automatic Differentiation. The number of simplices where a monotonic directional derivative was found (OkCV) was neither specifically large in comparison to the number N/N2. The reason is that only few easy directions were checked using CV and  $0 \notin G$  for  $\square S$  does not imply the existence of a monotonic directional derivative in  $S$ . Extreme cases are G7 (OkCV=0) and L8 where OkCV=343 out of 135,590. Still, the use of the new described monotonicity test reduces the search space for most instances. In general, the shifted minimum instances are more challenging to be solved. Again, the number of N/N2 in comparison to the number of ES can be considered low in general as well as between N/N2 and OkCV for the most challenging instances. Anyway, the use of the described monotonicity test reduces the search space in all instances. However, just one simplex is rejected for instance G7.

Table 7 shows that adding directions generated by LP(6) + LP(7) results in a slightly smaller number of evaluated simplices, but a higher running time. Instances ST5 and H6 could not be solved within 5 min.

Using CPLEX instead of PNL (Table 8) results in a larger running time for most of the instances apart from instance ST5 that could now be solved in less than 5 min.

Using CV + LP(6) + MIP(8) (see Table 9) does not improve the running time. Instances ST5, H6 and DP7 could not be solved in less than 5 min.

Table 10 illustrates an increase in running time when using MIP models instead of LP models or not using models at all. For some instances, the running time is better than that reported in Table 9, but for other instances, it is worse.

**Table 7** Results using monotonicity tests evaluating CV and LP generated directions using solver PNL (CVLP)

Ins	Time	ES	N/N2	OkCV	NSLP6	OkLP6	NSLP7	OkLP7	Div
MC2	0.02s	406	43/13	23	0	0	13	3	6
SHCB2	0.02s	517	34/6	24	0	0	8	4	2
THCB2	0.02s	525	22/0	16	0	0	9	6	0
SCH2	0.03s	767	5/0	3	0	0	2	2	0
H3	0.05s	1582	205/0	144	25	6	43	37	21
S4	0.29s	4284	550/0	175	208	134	308	98	149
GP2	0.04s	7202	19/2	7	0	0	15	2	2
H4	1.91s	24,965	5139/0	1768	2400	1467	2470	1036	1703
MCH5	3.91s	54,448	5113/0	1404	2644	2154	3238	1021	1957
G7	4.85s	278,325	816/0	1	0	0	815	2	0
RB7	41.33s	665,421	21.978/0	1522	0	0	20,457	1460	0
ST5	> 5 m								
L8	51.58s	1,168,075	9904/0	45	0	0	9862	376	0
H6	> 5 m								
DP7	2 m 17.28s	6,246,242	58,569/0	3303	0	0	55,351	5317	0

**Table 8** Results using monotonicity tests evaluating CV and LP generated directions using solver CPLEX (CVLP)

Ins	Time	ES	N/N2	OkCV	NSLP6	OkLP6	NSLP7	OkLP7	Div
MC2	0.02s	406	43/13	23	0	0	13	4	6
SHCB2	0.02s	517	34/6	24	0	0	8	4	2
THCB2	0.02s	525	22/0	16	0	0	9	6	0
SCH2	0.03s	767	5/0	3	0	0	2	2	0
H3	0.08s	1587	210/0	148	26	9	46	34	23
S4	0.52s	4284	550/0	175	208	142	316	98	149
GP2	0.04s	7202	19/2	7	0	0	15	2	2
H4	4.00s	25,161	5302/0	1876	2455	1394	2397	931	1825
MCH5	7.29s	54,435	5110/0	1404	2641	2175	3259	1023	1954
G7	5.96s	278,325	816/0	1	0	0	815	2	0
RB7	1 m 13.37s	665,421	21,978/0	1522	0	0	20,457	1460	0
ST5	4 m 04.07s	867,835	215,961/0	61,628	122,906	88,646	120,626	38,300	91,443
L8	1 m 08.77s	1,168,075	9,904/0	45	0	0	9862	376	0
H6	>5 m								
DP7	3 m 56.44s	6,246,277	58,569/0	3303	0	0	55,351	5273	0

**Table 9** Results using monotonicity tests with CVMIP8 directions

Ins	Time	ES	N/N2	OkCV	NSLP6	OkLP6	NMIP8	OkMIP8	Div
MC2	0.06s	406	43/13	23	0	0	13	3	6
SHCB2	0.05s	526	37/9	24	0	0	8	1	5
THCB2	0.05s	531	27/5	16	0	0	9	1	2
SCH2	0.04s	767	5/0	3	0	0	2	0	0
H3	0.23s	1678	280/0	177	67	20	57	8	66
S4	1.46s	4382	634/0	230	237	164	338	44	203
GP2	0.09s	7202	19/2	7	0	0	15	2	2
H4	11.10s	26,457	6332/0	2414	2947	1730	2733	583	2483
MCH5	21.35s	57,206	7427/0	2308	4054	3409	4493	843	3338
G7	7.06s	278,322	816/0	1	0	0	815	76	0
RB7	1 m41.78s	665,572	21,978/0	1522	0	0	20,457	744	0
ST5	>5 m								
L8	1 m28.6s	1,168,031	9904/0	45	0	0	9862	592	0
H6	>5 m								
DP7	>5 m								

**Table 10** Results using monotonicity tests with CVMIP9 directions

Ins	Time	ES	N/N2	OkCV	NMIP8	OkMIP8	Div
MC2	0.06s	394	35/0	28	15	9	6
SHCB2	0.04s	520	35/4	27	8	4	6
THCB2	0.04s	524	22/0	16	9	7	2
SCH2	0.04s	766	5/0	3	2	2	0
H3	0.30s	1697	283/0	186	100	11	89
S4	1.71s	4482	734/0	307	463	130	333
GP2	0.09s	7205	17/0	7	15	5	10
H4	15.40s	27,460	6871/0	2936	4159	735	3424
MCH5	13.42s	53,850	4844/0	1762	3170	1091	2079
G7	7.46s	277,511	816/0	1	815	815	0
RB7	1 m 30.62s	645,287	21,978/0	1522	20,457	14,632	5825
ST5	>5 m						
L8	1 m 23.68s	1,158,698	9904/0	45	9862	9862	0
H6	>5 m						
DP7	>5 m						

## 7 Conclusions

We found that the concept of a directional derivative given bounds on the gradient is of importance to decide if a simplicial feasible set can be reduced to some of its facets or eliminated from consideration. Moreover, the concept of a border simplex serving as a facet of a subset is relevant. The paper summarizes the theoretical results and their proofs. Furthermore, it has been described how solving a Linear Program or Mixed Integer Linear Program can be used to detect with certainty the monotonicity in a subset.

The methods using LPs and MIPs have a theoretical character and due to numerical aspects can be hard to handle when subsets get small. Solving LPs and MIPs may require more computing time than easier tests, which are more practical to handle. Several numerical experiments are reported on (adapted) benchmark instances from literature.

A potential improvement of the described method can be found in the overestimation of the range of the gradient over a simplex, because the interval hull of the simplex is used to apply Interval Arithmetic and Automatic differentiation. Better and probably more expensive methods to get sharper bounds can be studied.

Regarding the use of LP and MIP models to search for a positive directional derivative bound in a simplex, the numerical results show that the number of positive results can be improved at an increasing running time. Their effectiveness could be increased by having sharper bounds on the gradient. Due to the large amount of simplices generated in a spatial branch and bound with a small number of accelerating devices, the time spent on each simplex should be short. We plan to develop a simple local search for a positive directional derivative. This could be less effective, but may result in a faster algorithm.

## A Small Instances

### A.1 Unconstrained Portfolio as QP Problem

Convex,  $n = 4$ .

$$\min f(x) = 10x^T Ax - \mu^T x, \tag{10}$$

$$\text{s.t. } \sum x_i \leq 1, x \geq 0 \tag{11}$$

$$\mu = (11.5, 9.5, 5, 4)^T,$$

$$A := \frac{1}{100} \begin{pmatrix} 166 & 34 & 58 & -1 \\ 34 & 64 & 4 & -1 \\ 58 & 4 & 100 & 1 \\ -1 & -1 & 1 & 10 \end{pmatrix}$$

Minimum  $f^* = -5.3732$  at  $x = (0.1678, 0.405, 0.0, 0.4272)^T$

### A.2 EX6

Taken from Example 6 in [10].

Convex,  $n = 2$ .

$$\min f(x) = 0.25x_0^2 + x_0 + x_1 + 0.25x_0x_1 + 0.5x_1^2$$

#### A.2.1 EX6-B: Minimum at the Boundary

For iBB:

$$\text{s.t. } x \in ([-2.0, 2.0], [-3.0, 3.0])^T, Ax \leq b.$$

$$A = \begin{pmatrix} 3 & -1 \\ -3 & 4 \\ -3 & -2 \end{pmatrix}, b = \begin{pmatrix} 3 \\ 6 \\ 6 \end{pmatrix}$$

Minimum  $f^* = -1.140625$ , at  $(-1.625, -0.5625)^T$ .

For sBB, the corresponding vertex matrix is  $V = \begin{pmatrix} -2.0 & 0.0 & 2.0 \\ 0.0 & -3.0 & 3.0 \end{pmatrix}$

#### A.2.1 EX6-In: Minimum in the Interior

For iBB:

$$\text{s.t. } x \in ([-3.0, 1.5], [-2.0, 1.0])^T, Ax \leq b.$$

$$A = \begin{pmatrix} -1 & 1 \\ 6 & 5 \\ -2 & -9 \end{pmatrix}, b = \begin{pmatrix} 2 \\ -1 \\ 15 \end{pmatrix}$$

$$\text{Minimum } f^* = -1.14286 \text{ at } \left(-\frac{12}{7}, -\frac{4}{7}\right)^T \approx (-1.71429, -0.57143)^T$$

For sBB, the corresponding vertex matrix is  $V = \begin{pmatrix} -3.0 & -1.0 & 1.5 \\ -1.0 & 1.0 & -2.0 \end{pmatrix}$

**Data Availability** The data of the constructed test instances can be found in [www.sfu.ca/~ssurjano/optimization.html](http://www.sfu.ca/~ssurjano/optimization.html) and is publicly available.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Aparicio, G., Casado, L.G., Hendrix, E.M.T., G.-Tóth, B., García, I.: On the minimum number of simplex shapes in longest edge bisection refinement of a regular  $n$ -simplex. *Informatica* **26**(1), 17–32 (2015). <https://doi.org/10.15388/informatica.2015.36>
2. Casado, L.G., G.-Tóth, B., Hendrix, E.M.T., Messine, F.: On monotonicity detection in simplicial branch and bound over a simplex. In: O. Gervasi, B. Murgante, S. Misra, A.M.A.C. Rocha, C. Garau (Eds.) *Computational Science and its Applications—ICCSA 2022 Workshops*, pp. 113–126. Springer International Publishing, Cham (2022). [https://doi.org/10.1007/978-3-031-10562-3\\_9](https://doi.org/10.1007/978-3-031-10562-3_9)
3. De Klerk, E.: The complexity of optimizing over a simplex, hypercube or sphere: a short survey. *CEJOR* **16**, 111–125 (2008). <https://doi.org/10.1007/s10100-007-0052-9>
4. Du, K., Kearfott, R.B.: The cluster problem in multivariate global optimization. *J. Global Optim.* **5**(3), 253–256 (1994). <https://doi.org/10.1007/BF01096455>
5. G.-Tóth, B., Casado, L.G., Hendrix, E.M.T., Messine, F.: On new methods to construct lower bounds in simplicial branch and bound based on interval arithmetic. *J. Global Optim.* **80**(4), 779–804 (2021). <https://doi.org/10.1007/s10898-021-01053-8>
6. G.-Tóth, B., Hendrix, E.M.T., Casado, L.G.: On monotonicity and search strategies in face based copositivity detection algorithms. *Central Eur. J. Oper. Res.* **30**, 1071–1092 (2021). <https://doi.org/10.1007/s10100-021-00737-6>
7. Horst, R.: On generalized bisection of  $n$ -simplices. *Math. Comput.* **66**(218), 691–699 (1997). <https://doi.org/10.1090/s0025-5718-97-00809-0>
8. Hendrix, E.M.T., G.-Tóth, B., Messine, F., Casado, L.G.: On derivative based bounding for simplicial branch and bound. *RAIRO* **55**(3), 2023–2034 (2021). <https://doi.org/10.1051/ro/2021081>
9. Hansen, E., Walster, W.: *Global Optimization Using Interval Analysis*, 2<sup>nd</sup> edn. Marcel Dekker Inc., New York (2004)
10. Hendrix, E.M.T., Salmerón, J.M., Casado, L.G.: On function monotonicity in simplicial branch and bound. In: *LeGO 2018*, pp. 4. Leiden (The Netherlands) (2018). <https://doi.org/10.1063/1.5089974>
11. Horst, R., Tuy, H.: *Global Optimization (Deterministic Approaches)*. Springer, Berlin Heidelberg (1990). <https://doi.org/10.1007/978-3-662-03199-5>
12. Karhbet, S.D., Kearfott, R.B.: Range bounds of functions over simplices, for branch and bound algorithms. *Reliable Computing* **25**, 53–73. <https://interval.louisiana.edu/reliable-computing-journal> (2017)
13. Kearfott, R.B.: An interval branch and bound algorithm for bound constrained optimization problems. *J. Global Optim.* **2**(3), 259–280 (1992). <https://doi.org/10.1007/BF00171829>
14. Mohand, O.: Tighter bound functions for nonconvex functions over simplexes. *RAIRO Oper. Res.* **55**, S2373–2381 (2021). <https://doi.org/10.1051/ro/2020088>

15. Moore, R.E., Kearfott, R.B., Cloud, M.J.: Introduction to Interval Analysis. Society for Industrial and Applied Mathematics, USA (2009). <https://doi.org/10.1137/1.9780898717716>
16. Paulavičius, R., Žilinskas, J.: Simplicial Global Optimization. Springer, New York (2014)
17. Rall, L.B. (ed.): Examples of software for automatic differentiation and generation of Taylor coefficients, pp. 54–90. Springer, Berlin Heidelberg, Berlin, Heidelberg (1981)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.