



HAL
open science

Local search versus linear programming to detect monotonicity in simplicial branch and bound

L G Casado, B. G.-Tóth, E M T Hendrix, F. Messine

► **To cite this version:**

L G Casado, B. G.-Tóth, E M T Hendrix, F. Messine. Local search versus linear programming to detect monotonicity in simplicial branch and bound. *Journal of Global Optimization*, 2023, 10.1007/s10898-023-01310-y . hal-04829511

HAL Id: hal-04829511

<https://ut3-toulouseinp.hal.science/hal-04829511v1>

Submitted on 10 Dec 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



Local search versus linear programming to detect monotonicity in simplicial branch and bound

L. G. Casado¹ · B. G.-Tóth² · E. M. T. Hendrix³ · F. Messine⁴

Received: 17 November 2022 / Accepted: 22 June 2023
© The Author(s) 2023

Abstract

This study focuses on exhaustive global optimization algorithms over a simplicial feasible set with simplicial partition sets. Bounds on the objective function value and its partial derivative are based on interval automatic differentiation over the interval hull of a simplex. A monotonicity test may be used to decide to either reject a simplicial partition set or to reduce its simplicial dimension to a relative border (at the boundary of the feasible set) facet (or face) by removing one (or more) vertices. A monotonicity test is more complicated for a simplicial sub-set than for a box, because its orientation does not coincide with the components of the gradient. However, one can focus on directional derivatives (DD). In a previous study, we focused on either basic directions, such as centroid to vertex or vertex to vertex directions, or finding the best directional derivative by solving an LP or MIP. The research question of this paper refers to using local search (LS) based sampling of directions from vertex to facet. Results show that most of the monotonic DD found by LP are also found by LS, but with much less computational cost. Notice that finding a monotone direction does not require to find the direction in which a derivative bound is the steepest.

This work has been funded by PID2021-123278OB-I00 funded by MCIN/AEI/ 10.13039/501100011033 and by “ERDF A way of making Europe”. We thank the anonymous referees for their valuable comments and suggestions.

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

L. G. Casado
leo@ual.es

B. G.-Tóth
boglarka@inf.szte.hu

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

¹ CeiA3, University of Almería, Almería, Spain

² University of Szeged, Szeged, Hungary

³ Universidad de Málaga and Wageningen University, Wageningen, The Netherlands

⁴ LAPLACE-ENSEEIH, Toulouse-INP, University of Toulouse, Toulouse, France

Keywords Simplex · Branch and bound · Interval arithmetic · Linear programming · Local search

1 Introduction

Global optimization branch and bound methods aim at finding the set of global minimum points with a given guarantee. Specific methods are based on interval arithmetic [7]. Such methods guarantee to reach a so-called rigorous solution for problems that are typically limited to less than $n = 10$ decision variables. All kinds of tools like monotonicity tests, constraint propagation, various lower bounds etc. may speed up the convergence. As the methods are based on box shaped partition sets, they do not directly fit to a simplicial feasible set.

Practical global optimization problems with a simplicial set as feasible region are among others blending and portfolio optimization problems. Optimization over a simplex is useful for many modern applications in machine learning, signal processing, control and game theory, see references in [12]. We specifically investigated the design of simplicial branch and bound for low dimensional mixture design problems to reach a guaranteed solution, [3]. For an overview of simplicial branch and bound we refer to the book [14]. One of the questions not touched in the described simplicial methods is the extension of the so-called monotonicity test of interval branch and bound methods to simplicial solution sets.

The use of a monotonicity test in global optimization branch and bound over a simplex has been addressed recently [2, 6, 8]. One of the main goals is to reject a simplicial partition set or to reduce it to a so-called border facet by finding a monotonic directional derivative (DD) from the vertex opposite to the facet towards the facet. In previous studies, naive directions, such as from vertex to the centroid of the simplex or other vertices, were used with good results [5]. Additionally, linear programming (LP) models and mixed-integer linear programming (MILP) models were formulated, which find the optimal (the steepest) DD bound in most of the cases [2]. However, it is time consuming to solve several LP-s or even just one MIP every time a monotonic DD is sought.

Our research question is whether a local search (LS) algorithm to find a monotonic DD over a simplex from the information provided by the gradient enclosure over the interval hull of the simplex may do the job more efficiently. The LS algorithm has to be computationally cheap with a stopping criterion that keeps track of convergence in order to stop if the monotonic DD will not be reached in a few steps. Moreover, we provide a LS algorithm that looks for a monotonic DD, regardless of its sign.

The rest of the paper is organized as follows. Notation is introduced in Sect. 2. Section 3 shows mathematical properties on monotonicity over simplices. Section 4 describes the evaluated DD with a direction based on the centroid, the vertices, LP and LS for a given facet. Section 5 shows numerical results for a practical simplicial branch and bound algorithm over a set of test instances, and Sect. 6 summarizes our findings.

2 Notation

We consider the minimization of a continuously differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, over a feasible set Δ , which is a p -simplex, i.e. $\Delta := \text{conv}(\mathcal{W})$ is defined by a set of $p + 1$ affine

independent vectors that serve as vertices

$$\mathcal{W} := \{v_0, \dots, v_p\} \subset \mathbb{R}^n, p \leq n. \tag{1}$$

The idea is to find or enclose all global minimum points of

$$\min f(x), x \in \Delta. \tag{2}$$

The consideration of n -dimensional functions over $p \leq n$ lower dimensional simplicial feasible area appears for instance in blending problems [3]. Our context is that of a branch and bound algorithm which encloses all minimum points of f on Δ . The used partition sets are m -simplices S , where $m \leq p$, i.e. their dimension can be lower than that of the feasible set, in contrast to the algorithms described in [14]. This means $S := \text{conv}(\mathcal{V})$ with \mathcal{V} a set of $m + 1 = |\mathcal{V}|$ vertices. The branch and bound algorithm works with a set Λ of partition sets, which as a whole include all global minimum points.

Although we usually limit our context to the use of longest edge bisection, where the longest edge (v, w) of a partition set S is bisected using mid-point $x := \frac{v+w}{2}$, we pose the monotonicity question in a larger context where any partition method may be used, as described in [9].

Specifically, we focus on dimension reduction due to monotonicity considerations, where a set \mathcal{V} of vertices of m -simplex S is reduced to $\mathcal{V} \setminus \{v\}$ 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 does not have 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

$$\text{rint}(S) = \{x = \sum_j \lambda_j v_j, \lambda_j > 0, j = 0, \dots, m, \sum_{j=0}^m \lambda_j = 1\}. \tag{3}$$

The relative boundary of a simplex S is defined by removing the relative interior from it. Given a simplicial partition set S , we are interested in whether its (simplicial) facets F are border with respect to the feasible set Δ . In general, we can define a simplex to be border with respect to a simplicial feasible set.

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

Binary labelling of vertices can be used to keep track of border facets, as discussed in [2].

Relevant information of sub-set S is an enclosure \mathbf{g} of the gradient $\nabla f(x) \subseteq \mathbf{g} := [\underline{\mathbf{g}}, \overline{\mathbf{g}}], \forall x \in S$. Interval vector \mathbf{g} can be calculated by Interval Automatic Differentiation [13, 15] over the interval hull of a simplex $S (\square S)$ which implies under and over estimations. Enclosure \mathbf{g} indicates the traditional interval enclosure, but our reasoning is still valid for any other rigorous and sharper enclosure method for $\nabla f(S)$. Now consider directional vector $d = x - y$ as the difference between two points $x, y \in S$. Then, the corresponding DD, i.e. $d^T \nabla f(x)$, is also included in the interval

$$\mathbf{h} = d^T \mathbf{g} = [\underline{\mathbf{h}}, \overline{\mathbf{h}}] = \left[\sum_{i=1}^n \min\{d_i \underline{\mathbf{g}}_i, d_i \overline{\mathbf{g}}_i\}, \sum_{i=1}^n \max\{d_i \underline{\mathbf{g}}_i, d_i \overline{\mathbf{g}}_i\} \right]. \tag{4}$$

We consider example 6 (EX6) from [10] in Fig. 1, $f(x_1, x_2) = 0.25x_1^2 + x_1 + x_2 + 0.25x_1x_2 + 0.5x_2^2$ over the simplex $S = \{v_0 = (-0.75, -1.5), v_1 = (-1, 1), v_2 =$

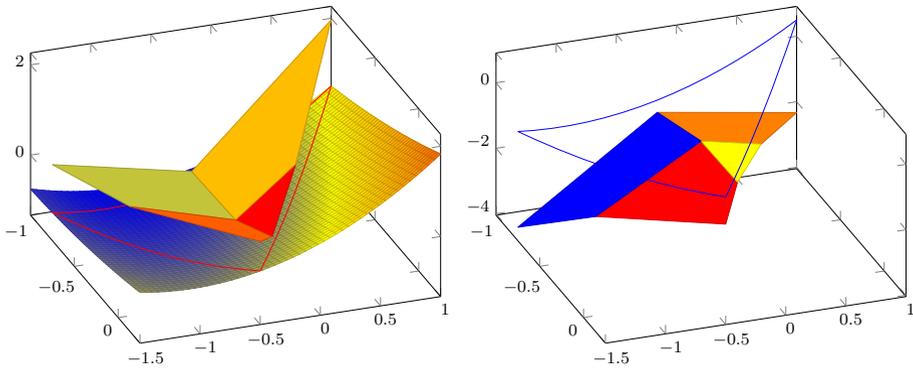


Fig. 1 Piecewise linear character of directional derivative bounds in x for $d = x - c$

$(0.25, -0.5)$. A gradient inclusion over box $([-1, 0.5], [-1.5, 1])$ is $\mathbf{g} = ([0.125, 1.375], [-0.75, 2.0625])$. As the figure shows, it means that following a first order Taylor idea, we have bounds starting from the center of the box, $c = (-0.375, 0.25)$. From the center, we also have bounds on the directional derivative in direction $d = x - c$. The illustration shows that the directional derivative is typically piecewise linear in the argument x , where the breaking points or planes are defined by the sign changing planes of d where $x_i = c_i$.

Our study focuses on the question of monotonicity, where we can show that there exist points x, y , such that the inclusion of the directional derivative does not contain zero. Consider

$$\mathbf{h}(x, y) = (x - y)^T \mathbf{g}, \tag{5}$$

as a function of $d = x - y, x, y \in S$.

Figure 2 shows an illustration of $\mathbf{h}(x, y)$ using example 6 (EX6): $f(x_1, x_2) = 0.25x_1^2 + x_1 + x_2 + 0.25x_1x_2 + 0.5x_2^2$ in [10], over the simplex $S = \{v_0 = (-0.75, -1.5), v_1 = (-1, 1), v_2 = (0.25, -0.5)\}$. An inclusion of the gradient is given by $\mathbf{g} = ([0.125, 1.375], [-0.75, 2.0625])$. This simplex has number 9 in Fig. 9. Thus, f is monotonically increasing to the right direction in any point of $\square S$ and therefore also in S . As directional vector we fix y in the centroid of $\square S$ and vary x over the unit circle around y . When varying x over the circle, the lower bound \underline{h} is piece-wise convex and \bar{h} is piece-wise concave.

In Fig. 3 we vary x over a polytope, specifically an octagon. It illustrates the piecewise linear character of the directional derivative bounds when we search over a hyperplane. This means it has breakpoints caused by $x_i = y_i$. Moreover, breakpoints are caused in this case by other vertices of the octagon. A minimum or maximum value will also occur in one of the breakpoints. This illustration shows that we might be looking for a monotonic direction d to approach a maximum value for \underline{h} (positive \mathbf{h}) or minimum for \bar{h} (negative \mathbf{h}).

Following the same example, Fig. 4 shows \mathbf{h} with direction $d = x - v, x \in \text{facet } F = \text{conv}(\mathcal{V} \setminus \{v\})$. The interesting values for directions d are those that correspond to $\underline{h} > 0$ or $\bar{h} < 0$. A monotonic direction may lead to reject S or to reduce it to a border facet(s). In this case, the only facet with a monotonic directional derivative bound \mathbf{h} is that leaves out v_2 with a minimum to the left direction from v_2 . In such a graphical example one can find a monotonic direction over a facet by running x over the facet, as illustrated in the figure for $n = 2$. In higher dimensions, finding a direction in which f is monotonic is not obvious.

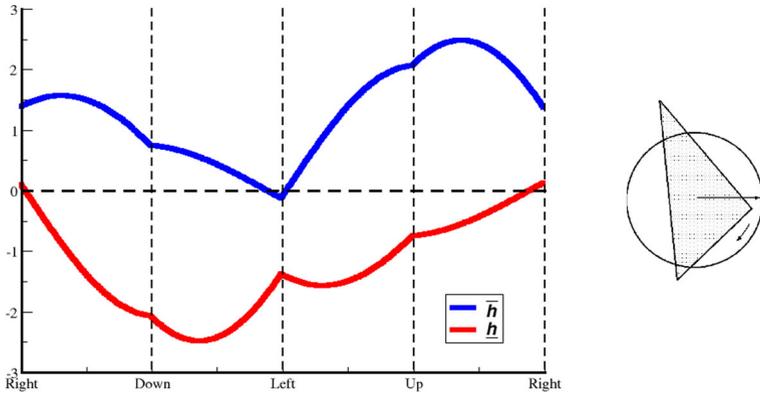


Fig. 2 Example of $h(x, y)$ with y the centre of the $\square S$ and x in $d = x - y$ varies over the unit circle with centre on y

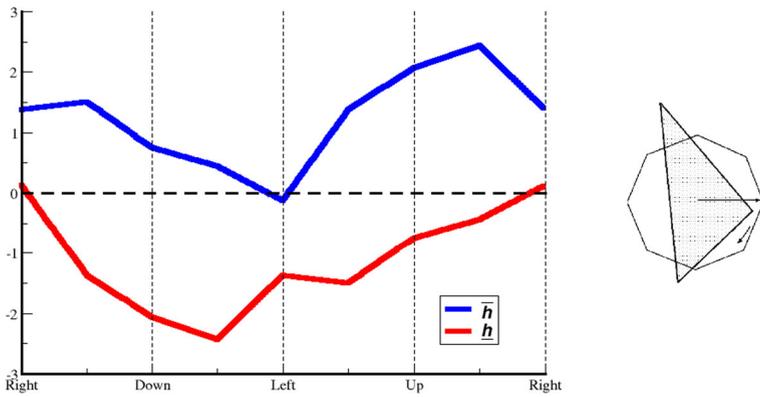


Fig. 3 Example of $h(x, y)$ with y the centre of the $\square S$ and x in $d = x - y$ varies over an octagon inscribed in the unit circle with centre on y

3 Mathematical properties on monotonicity

The monotonicity is based on the directional derivative bounds h (see (4)). Notice that condition $0 \notin g$ is necessary to have monotonicity, but not sufficient. The question is which direction d to consider. The most general result for an m -simplex is the following.

Proposition 1 [2] *Let $S \subseteq \Delta$ be an m -simplex with gradient enclosure g . If $\exists x, y \in S$, such that direction $d = x - y$ has corresponding directional derivative bounds (4) with $0 \notin h$ then $\text{rint}(S)$ does not contain a global minimum point of (2).*

For the proof, see [2].

Corollary 1 [2] *Let $S \subseteq \Delta$ be an m -simplex as partition set in a branch and bound algorithm with corresponding gradient enclosure g . If the conditions of Prop. 1 apply 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.

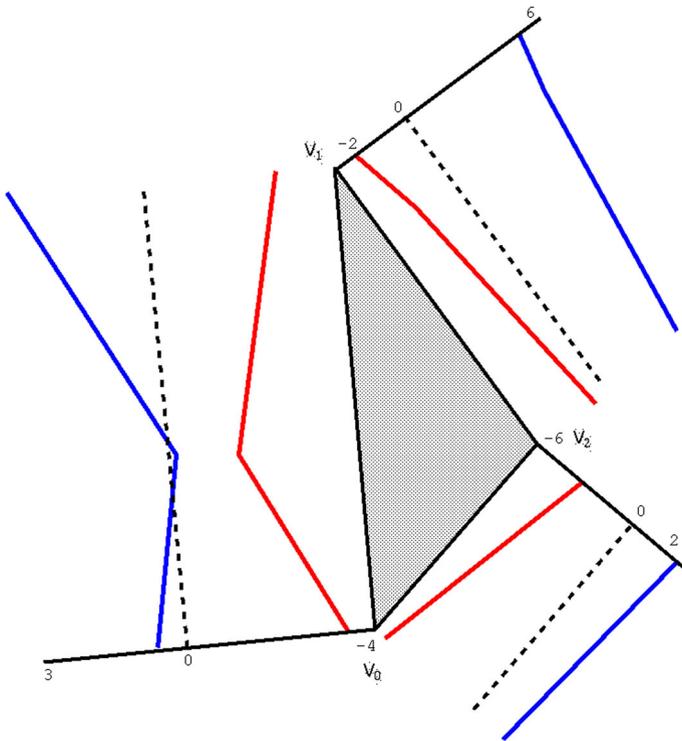


Fig. 4 Example of $h(x, v_j)$ for each simplex vertex $v_j, j = 0, 1, 2$. Endpoint x is varied over the opposite facet $F = \text{conv}(\mathcal{V} \setminus \{v_j\})$ using example 6 in [10] over the simplex $S = \{v_0 = (-0.75, -1.5), v_1 = (-1, 1), v_2 = (0.25, -0.5)\}$. The gradient bounds are $g = ((0.125, -0.75)^T, (1.375, 2.0625)^T)$

Given that the minimum is not in $\text{rint}(S)$, we have to decide which of the facets to focus on. In this paper, we focus on fixing one of the endpoints v and look for $d = (x - v), x \in S$, such that $0 \notin \mathbf{h}$.

Proposition 2 [2] *Given m -simplex $S = \text{conv}(\mathcal{V})$ and a facet F generated by removing vertex v from \mathcal{V} . Consider direction $d = x - v$ and $\mathbf{h} = d^T \mathbf{g}$. If $\mathbf{h} < 0$, then the facet F contains all minimum points in S , i.e. $\text{argmin}_{x \in S} f(x) \subseteq F$.*

Practically, Prop. 2 means that S can be replaced by F . However, a similar reasoning applies as in Corollary 1; if F is a non-border facet, then simplex S can be removed from further consideration in a branch and bound context. The idea is again that faces of F may contain the global minimum. However, because we are dealing with a partition, the same points are also included in other simplicial partition sets.

Proposition 3 *Given m -simplex $S = \text{conv}(\mathcal{V})$ and a facet F generated by removing vertex v from \mathcal{V} . Consider direction $d = x - v$ and $\mathbf{h} = d^T \mathbf{g}$. If $\mathbf{h} > 0$, then the facet F (even border) does not contain minimum points in S and S can be replaced by the union of its other border facets (see Prop. 1).*

The argument uses the opposite direction in Prop. 2.

It is more appealing to find a negative bound \mathbf{h} with d to a facet because we can reduce S to that facet if it is border or reject S otherwise. However, to find a positive direction for \mathbf{h} to a facet is also interesting because we can reduce the simplex to all other border facets, excluding the current one.

4 Directional derivatives to evaluate

We focus on finding a monotonic DD for a direction $d = x - v$ from vertex $v \in S$ towards a simplex facet $F = \text{conv}(\mathcal{V} \setminus \{v\})$. The first heuristic directions to consider is taking $x = c$ (centroid of the facet) and the other vertices in $\mathcal{V} \setminus \{v\}$. These are described in Sect. 4.1.

Based on these considerations, we define steps for the branch and bound algorithm with a certain order to check various directions on monotonicity. The algorithm will check the centroid (C) direction for each facet of S and checks vertex (V) directions in case a negative \mathbf{h} was not found. A further refinement considering LP or LS methods will be done after C+V only if a negative \mathbf{h} was not found for any facet. Notice that these tests are only relevant if $0 \notin \mathbf{g}$, which is a necessary condition for monotonicity. The order of the tests is described as follows.

1. S is full dimensional without border facets and $0 \notin \mathbf{g}$. Reject S .
2. For non-border facet $F, \exists d = x - v$ for which $\mathbf{h} < 0$. Reject S . See Prop. 2 and Corollary 1.
3. There are border facets F having $\mathbf{h} < 0$. Reduce S to the intersection of those facets. See Prop. 2.
4. There are facets F with $\mathbf{h} > 0$. Replace S by the union of its border facets. See Prop. 3. If there are no (other) border facets, reject S ; Corollary 1.
5. We have for S that $0 \notin \mathbf{g}$ but for all facets, and all tested direction, $0 \in \mathbf{h}$:
 - (a) S is full dimensional. Replace S by the union of its border facets. Minimum can neither be in $\text{rint}(S)$ and in non-border facets of S .
 - (b) S is not full dimensional. Monotonicity cannot be proved; divide S .

We summarize an LP problem which finds a positive bound on all directional derivatives in Sect. 4.2 and describe an LS method in Sect. 4.3. LP or LS will continue checking monotonicity for all facets when no $\mathbf{h} < 0$ was found towards a facet using naive C and V directions. Using C+V (CV) we can find more than one facet with a negative \mathbf{h} . LP or LS will stop after finding the first negative \mathbf{h} in order to reduce their impact in the computational burden.

We define a monotonicity measure for non-monotone DD-s, that is, when $0 \in \mathbf{h}$ as

$$\mu(\mathbf{h}) = \min\{\overline{\mathbf{h}}, -\underline{\mathbf{h}}\}, \tag{6}$$

which can be seen as the minimum distance in the interval \mathbf{h} to escape from zero.

In order to reduce the number of facets to be checked by LP and LS, facets with the lowest value of $\mu(\mathbf{h}(c, v))$, where c the centroid of facet $F = \text{conv}(\mathcal{V} \setminus \{v\})$, are considered first.

4.1 Naive directional derivatives using centroid and vertices

Having at least one coordinate i with $0 \notin \mathbf{g}_i$, the algorithm first checks monotonicity considering as directions

1. C: $d = (c - v)$, with c the centroid and v the opposite vertex of F , for each facet $F \in S$.
If no negative value for \mathbf{h} is found, we check
2. V: directions towards vertices, $d = (w - v)$, $v, w \in \mathcal{V}$.

The algorithm evaluates all $\binom{m}{2}$ combinations of vertices in \mathcal{V} without repetition for d in \mathbf{h} in lexicographical order. If $\mathbf{h}(v, w) > 0$ for any $v, w \in \mathcal{V}$, obviously $\mathbf{h}(w, v) < 0$. So every time a monotone direction is found, we can reduce the dimension (Prop. 1) or even reject the simplex (paragraph after Prop. 2).

4.2 The linear programming model

To prove that there exists a monotone direction in an m -simplex, at least we should have $0 \notin \mathbf{g}$. This is a necessary, but not sufficient condition for an m -simplex, $m < n$.

We can search for any monotone direction $d = (x - y)$, $x, y \in S$ to find $0 \notin \mathbf{h}(x, y)$. By fixing $y = c$ we reduce the degree of freedom. If any d is found with $\mathbf{h} > 0$, we know that the interior of the simplex can be removed. The corresponding LP is formulated in [2], as LP(6).

Regarding Prop. 2, we can focus on finding $\mathbf{h}(x, v) < 0$ in direction $d = (x - v)$, $x \in F$ and $F = \text{conv}(\mathcal{V} \setminus \{v\})$:

$$0 > \bar{\mathbf{h}} = \sum_{i=1}^n \max\{d_i \underline{\mathbf{g}}_i, d_i \bar{\mathbf{g}}_i\}. \quad (7)$$

Finding a direction d which solves (7) can be done by searching for the steepest monotone direction $\min_d \bar{d}^T \bar{\mathbf{g}}$. The corresponding LP is formulated in [2], as LP(7).¹ Notice that we do not necessarily need to know the minimum in (7), but whether there exists $x \in F$ for which $\bar{\mathbf{h}} < 0$.

As mentioned in Sect. 4, finding $\mathbf{h} > 0$ instead of $\mathbf{h} < 0$ in (7) is also useful. However, looking for $\mathbf{h} > 0$ requires to solve another LP and we do not consider it in the experiments due to the large computational cost of solving a large number of LP models. Nevertheless, it may happen that looking for $\mathbf{h} < 0$, we end up in $\mathbf{h} > 0$, i.e. $\min \bar{\mathbf{h}}$ results in $\underline{\mathbf{h}} > 0$. In order to save LP computations, we neither do LP(6) in the experiments, because a successful result of LP(7) makes LP(6) unnecessary.

LP can be used as the best solution to find a monotone direction, but it might not be computationally the best. This is investigated in the experiments of Sect. 5.

4.3 Local search algorithm

The LS focuses on a simplex $S = \text{conv}(\mathcal{V})$ and one of its facets $F = \text{conv}(\mathcal{V} \setminus \{v\})$. Consider a set \mathcal{P} of already evaluated points $x \in F$ with respect to $\mathbf{h}(x, v)$. This means, after checking the centroid and other vertices, we have $\mathcal{P} = \{\mathcal{V} \setminus \{v\}, c_0\}$. Each iteration generates a new endpoint x based on a centroid c_k of several points in \mathcal{P} , which replaces one of the points in \mathcal{P} trying to reduce the non-monotonicity measure in (6). In this way, we look for a bound $\mathbf{h} = \mathbf{h}(x, v)$ with a low value of $\mu(\mathbf{h})$ regardless of the sign of \mathbf{h} , see Props. 2 and 3.

Algorithm 1 shows the local search for a monotone direction towards a facet determined by removing v from \mathcal{V} .

For the stopping criterion, Algorithm 2 estimates the remaining number of iterations based on the progress of the bound $\mathbf{h}(x, v)$ for the last three centroids $x = c_{i-2}, c_{i-1}, c_i$ in Alg.

¹ But using $d = (v - x)$ and $\mathbf{h} > 0$ instead.

Algorithm 1 LS (S , g , v , n , m)

Require:

- n :Dimension of f .
- m Simplicial dimension of S .
- S :Simplex $S = \text{conv}(\mathcal{V})$.
- g :Enclosure of the gradient, $\nabla f(\square S) \subseteq g$.
- v :Vertex determining the facet.
- \mathcal{P} Set of endpoints $x \in \mathcal{P}$ with calculated $\mathbf{h} = \mathbf{h}(x, v)$.

- 1: $k = 0$ ▷ Number of iterations
- 2: Initialize $\mathcal{P} = \mathcal{V} \setminus v \cup \{c_0\}$ ($c_0 = c$ centroid), $\mathbf{h}_0 = \mathbf{h}(c_0, v)$
- 3: **repeat**
- 4: $k = k + 1$
- 5: **for** $j \in \{0, \dots, m\}$ **do**
- 6: Let p_j be point j in set $\mathcal{P} \setminus \{c_{k-1}\}$
- 7: determine new point $x_j := \frac{1}{m} \sum_{x \in \mathcal{P} \setminus \{p_j\}} x$,
- 8: **if** $0 \notin \mathbf{h}(x_j, v)$ **then**
- 9: **return true** ▷ Monotone direction is found.
- 10: determine $\ell = \text{argmin}_{j \in \{0, \dots, m\}} \mu(\mathbf{h}(x_j, v))$
- 11: remove p_ℓ from \mathcal{P} and add $c_k := x_\ell$, $\mathbf{h}_k := \mathbf{h}(c_k, v)$
- 12: **until** $k > n \times (m + 1)$ or $(k \geq 2$ and $\text{EstRemIters}(\mathbf{h}_{k-2}, \mathbf{h}_{k-1}, \mathbf{h}_k) > m + 1)$
- 13: **return false** ▷ Monotone direction is not found.

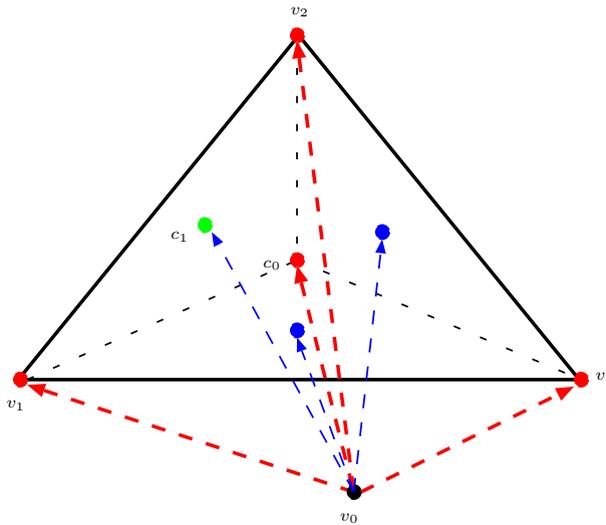


Fig. 5 Graphical example of one iteration of Algorithm 1. Facet $F = \text{conv}(\{v_1, v_2, v_3\})$ in a 3-simplex S is shown in black. Already evaluated directions $d = (v_0 - x)$, $x \in \mathcal{P} = \{v_1, v_2, v_3, c_0\}$ are drawn as dashed red arrows. The candidates x_j for new centroid c_1 and their direction $(v_0 - x_j)$ are drawn as dashed blue arrows. For instance, x_2 is the centroid of $\{v_1, v_2, c_0\} = \mathcal{P} \setminus \{v_3\}$. The green point shows the endpoint with the best μ among the candidates x_j which becomes c_1 . In the next iteration, $\mathcal{P} = \{v_1, v_2, c_0, c_1\}$ to get c_2 . The bounds $\mathbf{h}(c, v_0)$ with $c = c_0, c_1, c_2$ are used in $\text{EstRemIters}()$ to estimate the number of remaining iterations and to check whether we have slow convergence

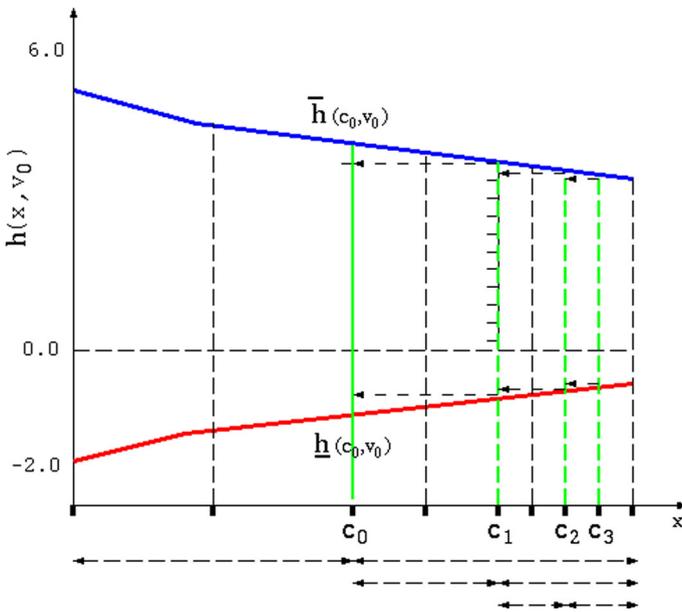


Fig. 6 Example of estimating number of remaining iterations by Algorithm 2 on facet 0 in Fig. 4. At iteration 2 (c_0, c_1, c_2), $UpRemIt_1=11$, $UpRemIt_2=21$, $LoRemIt_1=3$ and $LoRemIt_2=5$ for $EstRemIters(\mathbf{h}_0, \mathbf{h}_1, \mathbf{h}_2)$ resulting in $3 \leq |\mathcal{P}|$. At iteration 3 (c_1, c_2, c_3), $UpRemIt_2=21$, $UpRemIt_3=41$, $LoRemIt_2=5$ and $LoRemIt_3=9$ for $EstRemIters(\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3)$. The minimum is $5 > |\mathcal{P}| = 3$ and LS stops

1, in a similar way as the box-rejection idea in [1]. If the estimate is bigger than the number of endpoints in \mathcal{P} ($|\mathcal{P}| = m + 1$) in Algorithm 1, the search is stopped. Figure 6 shows an example using facet 0 of the simplex in Fig. 4. The algorithm stops at iteration 3 because the estimated number of remaining iterations from the current iteration is 9. At iteration 2 that number was 3. The total number of iterations of the local search is also limited in line 12.

Algorithm 2 $EstRemIters(\mathbf{h}_{k-2}, \mathbf{h}_{k-1}, \mathbf{h}_k)$

- 1: $LoRemIt_{k-1} = \infty, UpRemIt_{k-1} = \infty$ ▷ Remaining iterations by \mathbf{h}_{k-1} and \mathbf{h}_{k-2}
 - 2: $LoRemIt_k = \infty, UpRemIt_k = \infty$ ▷ Remaining iterations by \mathbf{h}_k and \mathbf{h}_{k-1}
 - 3: **for** $j = k - 1$ and $j = k$ **do**
 - 4: **if** $\underline{h}_j \leq \underline{h}_{j-1}$ **then**
 - 5: $LoRemIt_j = \lceil \frac{\underline{h}_j}{\underline{h}_{j-1} - \underline{h}_j} \rceil$
 - 6: **if** $\bar{h}_j \leq \bar{h}_{j-1}$ **then**
 - 7: $UpRemIt_j = \lceil \frac{\bar{h}_j}{\bar{h}_{j-1} - \bar{h}_j} \rceil$
 - 8: **return** $\min\{LoRemIt_{k-1}, LoRemIt_k, UpRemIt_{k-1}, UpRemIt_k\}$
-

In Fig. 6, one first checks centroid c_0 , and then the two vertices. Then LS checks another 7 directions. So, in total 9 DD were checked. LS on facet 0 is an example were the LS stops without finding a monotonic direction. Figure 7 shows an example were the algorithm finds a monotonic direction for facet 2. In fact, facet 0 needs not be checked, as facet 2 will be visited first, because the smallest non-monotonicity measure among the vertex to center

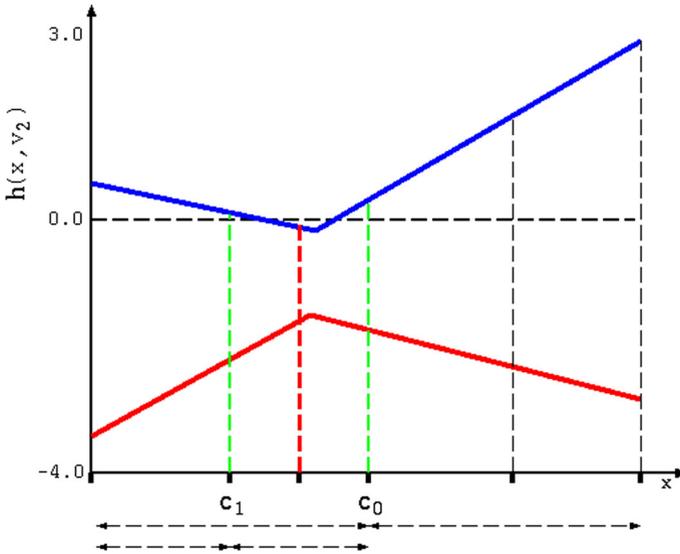


Fig. 7 Example of LS on facet 2 of simplex in Fig. 4

directions was attained for facet 2, i.e. $v_2 = \operatorname{argmin}_w \mu(\mathbf{h}(w, c))$ with c the centroid of facet $F = \operatorname{conv}(V \setminus \{v\})$. The LS found a negative directional derivative and therefore finishes after checking facet 2 on this simplex.

5 Numerical results

Algorithms were run on an Intel(R) Core(TM) i7-4770K CPU and 16GB of RAM running Fedora 36 Linux distribution. The algorithm was coded with g++ (gcc version 12.1.1) and uses [Kv-0.4.50](#) for Interval Arithmetic. Kv uses [boost](#) libraries. Algorithms were compiled with `-O3 -DNDEBUG -DKV_FASTROUND` options. For the Linear Programming, we use [PNL 1.10.4](#) compiled with `-DCMAKE_BUILD_TYPE = Release` and `-DWITH_MPI = OFF`, as a C++ wrapper to [5.5.2.0-28](#). LPSolve is free but it is slower than other commercial solvers². PNL uses by default `set_break_numeric_accuracy(lp, 5e-7)` in `lp_lib.c` and sets default parameter values and tolerances to `ProductionPARAM` in `lp_lib.h`. In order to reduce the computational time of LPSolve, we set its verbosity to "SEVERE", and time out after 1 s. We use the algorithm with lower bounding +CFVs from [5], which refers to a lower bound calculated by the central form using the vertex with maximal objective function value. The termination accuracy is $\alpha = 10^{-6}$.

The used test instances are similar as in [5] and are given in Table 1. Box constrained problems are transformed to simplicial constrained problems where the minimum is at the centroid of one of the facets of a simplex of size half of the bounding box. For instance, problem GP2 is box constrained on $([-2, 2], [-2, 2])$ and the minimum is at $(0, -1)$. The corresponding simplicial feasible set for GP2 has vertex set $\{(-0.5, -2), (1.5, -2), (0.5, 0.0)\}$, as illustrated in Fig. 8.

² <https://www.heureka.lu.se/wiki/Optimization>

Table 1 Test problems. An asterisk at n indicates that this is the selected dimension for a varying dimension test instance. Their formulation and minima can be found at <https://www.sfu.ca/~ssurjano/optimization.html>

Instance	Description	n
DP7	Dixon-price	7*
EX6-1	Karhbet example 6 over simplex 1	2
EX6-2	Karhbet example 6 over simplex 2	2
GP2	Goldstein-price	2
G7	Griewank	7*
H3	Hartmann 3	3
H4	Hartmann 4	4
H6	Hartmann 6	6
L8	Levy	8*
MC2	McCormick	2
MCH5	Michalewicz	5*
RB7	Rosenbrock	7*
SHCB2	Six Hump Camel Back	2
S4	Shekel 10	4
SCH2	Schubert	2
ST5	Styblinski-Tang	5*
THCB2	Three hump Camel Back	2
UPQP	Unconstrained Portfolio QP	4

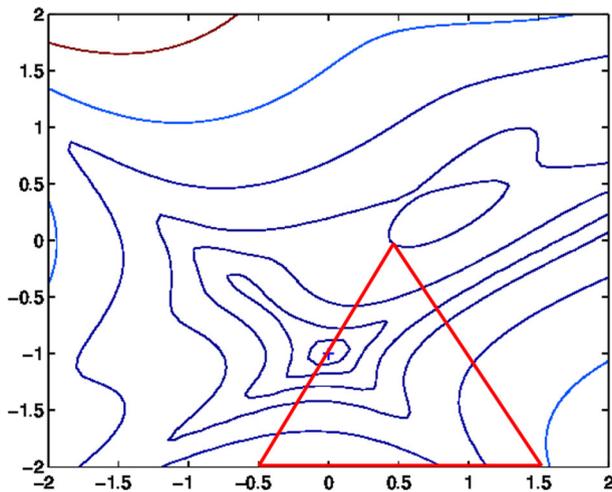


Fig. 8 Simplicial constrained version of test instance GP2

Instances EX6-1 and EX6-2 and UPQP have a simplicial feasible set. Specifically, EX6-1 and EX6-2 have vertex sets $\mathcal{V} = \{(-3, -1), (1, 1), (1.5, -2)\}$ and $\mathcal{V} = \{(-2, 0), (0, -3), (2, 3)\}$, respectively (see [10]). The portfolio problem is a standard application over the standard simplex and in fact not hard to solve. However, it illustrates well the effect of the various monotonicity tests on the instance. A description of the portfolio quadratic program (UPQP) can be found in “Appendix A”.

The studied variants of the monotonicity test are the following:

Table 2 Colors used in Figs. 9 to 11

	Color	Meaning
Simplex	Dark green	Rejected by monotonicity test
	Light green	Reduced to border facet(s)
	Brown	Rejected by RangeUp test: $\underline{f}(S) > \tilde{f}$ =incumbent $\min f(v)$
Directions	Light red	$h < 0$ for a direction C or V
	Dark red	$h < 0$ for a direction provided by LP or LS
	Blue	$h > 0$ for a direction C or V
	Dark blue	$h > 0$ for a direction provided by LP or LS
	Black	$0 \in h$
Vertex	Yellow	Evaluated vertex
	Red	Incumbent vertex

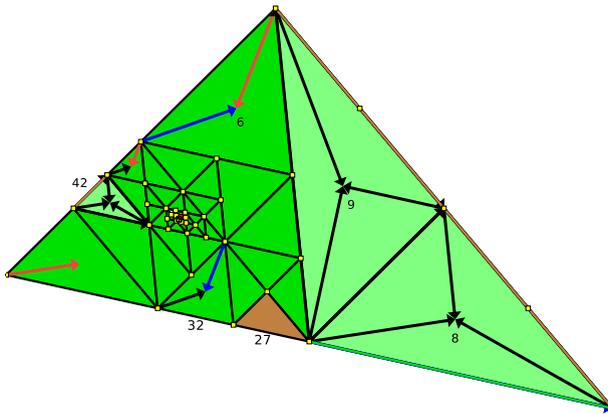


Fig. 9 Results for EX2-1 instance using the CV directions for monotonicity for facet oriented directions

CV	Using $(c - v)$ and $(w - v)$ directions for h (see Sect. 4.1).
CV+LP	Using CV plus linear programming (see Sect. 4.2).
CV+LS	Using CV plus local search (see Sect. 4.3).

Figures 9 to 11 show the graphical output of the CV, CV+LP and CV+LS methods. The colors are explained in Table 2.

Some simplices are numbered only in Fig. 9. The centroid of the simplex is used for the CV method to save computations. The centroid of the facet was used for CV+LP or CV+LS methods (see the last paragraph of Sect. 4). It depicts the state of the algorithm on the example EX2-1. Notice that this instance is full dimensional. As we check directions for the facets only when $0 \notin g$, monotonicity on the complete simplex has already been demonstrated before.

Figure 9 illustrates two monotonic directions from a vertex to centroid for simplex 6. The direction to a non-border facet with a positive bound (blue) implies that S can be replaced by the only border facet. The direction with a negative directional derivative bound (light red) to a non-border facet is more interesting; we can reject the simplex, see comments after Prop. 2.

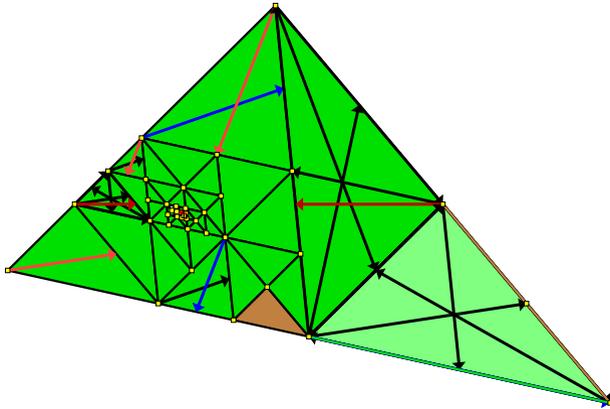


Fig. 10 Results for EX2-1 instance using LP method

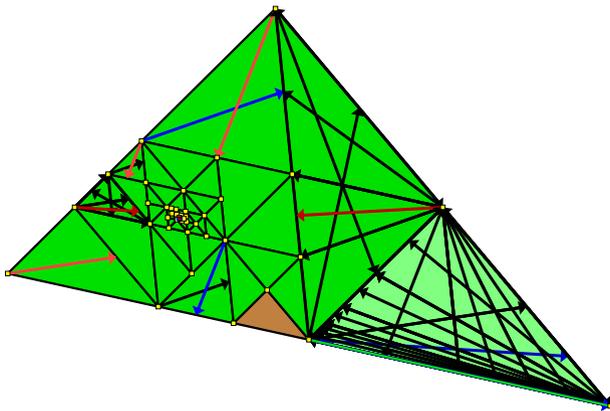


Fig. 11 Results for EX2-1 instance using LS to find monotonicity from a vertex

Simplices 8, 9 and 42 did not reveal a monotonic direction from a vertex to the centroid. As $0 \notin \mathbf{g}$, they can be reduced to their border facets. Simplex 27 is rejected by the RangeUp test, as well as the simplices being border facets of simplex 42 and 9. Simplex 32 has a positive directional derivative bound \mathbf{h} from a vertex to a border facet. This means the simplex can be rejected.

Figure 10 illustrates the direction with negative directional derivative bounds \mathbf{h} found by LP for simplices 9 and 42. Based on these directions, they can be rejected.

Figure 11 illustrates the directions found by the LS method with a negative directional derivative bound \mathbf{h} for simplices 9 and 42. The figure shows the directions from vertex to trial points x_j for which $\mathbf{h}(x_j, v)$ has been evaluated. The LS method additionally found a positive directional derivative bound \mathbf{h} for a direction to a border facet of simplex 8 (right-top facet). This finding let us discard that border facet from the search (see Prop. 3). The LP method aims to find a negative value for $\mathbf{h}(x, v)$. We did not run an opposite formulation due to computational cost of solving LPs (see Table 5).

Edge facets have two directions, the vertex to vertex directions, so only one has to be evaluated, the other one is opposite. For instance, the lower facet of simplex 8 has a positive

Table 3 Meaning of headers in the following tables

Header	Meaning
Ins	Problem instance
NS	Number of evaluated simplices
Time	Execution time. Differences smaller than 0.1s are not significant
N/N2	Number of simplices/segments checked for $h(x, v_i)$
NSCV	Number of simplices with monotonic directions C and/or V
NSLP	Number of simplices with monotonic LP direction with negative $h(x, v_i)$
NLPF	How many times solving an LP fails
NSLS	Number of simplices with monotonic directions by LS
Div	Reduced simplices with $0 \notin g$ but no monotone direction found, thus divided

Table 4 Results using Centroid-vertex and vertex-vertex directions (CV)

Ins	NS	Time	N/N2	NSCV	Div
EX6-2	67	0.01s	42/32	10	16
EX6-1	217	0.01s	10/3	4	2
MC2	399	0.01s	37/10	16	1
THCB2	528	0.01s	32/11	5	11
SHCB2	619	0.01s	39/7	30	4
SCH2	816	0.03s	4/0	3	0
UPQP	859	0.01s	793/0	364	429
G7	1,699	0.03s	44/0	0	0
H3	2,168	0.03s	306/0	179	77
GP-2	7,375	0.05s	15/3	6	3
S4	9,861	0.12s	1,112/0	287	751
H4	44,200	0.40s	8,559/0	4,400	2,754
MCH5	61,401	0.91s	10,971/0	4,100	6,025
RB7	88,276	0.54s	4,279/0	134	0
DP7	863,291	4.48s	9,055/0	381	0
ST5	1,035,921	5.47s	353,550/0	138,243	186,948
H6	2,288,026	30.77s	1,381,682/0	544,376	803,582
L8	9,203,596	3m35.23s	135,590/0	343	0

directional derivative bound h to the right vertex (border), which also gives a negative DD in the opposite way (to the left vertex, which is non-border). This means, we can reject the facet as a simplex (see comments after Prop. 2).

Tables 4, 5 and 6 show the numerical results using the different monotonicity test methods. The header meaning is explained in Table 3. Best results are shown in bold values.

Regarding monotonicity, those instances with a small value for N/N2, high NSCV, NSLP or NSLS and small value for Div are those where LP and Local Search are effective.

Using LP or LS reduces the number of evaluated simplices for most of the instances apart from EX6-2, G7, SCH2 and MC2. None of the methods found a monotonic directional derivative from a vertex in G7. However, 44 full dimensional simplices were reduced to

Table 5 Results using CV+LP

Ins	NS	Time	N/N2	NSCV	NSLP	NLPF	Div
EX6-2	67	0.01s	42/32	10	0	0	16
EX6-1	213	0.01s	9/2	4	2	0	1
MC2	399	0.01s	37/10	16	1	0	1
THCB2	500	0.02s	21/0	5	16	0	0
SHCB2	618	0.01s	39/7	30	2	0	4
SCH2	816	0.02s	4/0	3	0	0	0
UPQP	668	0.20s	641/0	259	51	0	331
G7	1,699	0.11s	44/0	0	0	0	0
H3	2,080	0.06s	238/0	143	32	0	44
GP-2	7,374	0.04s	15/3	6	1	0	3
S4	9,170	0.41s	691/0	171	58	0	413
H4	41,109	2,24s	6,353/0	3,016	1,137	1	1,357
MCH5	53,254	3.34s	4,793/0	1,351	926	0	1,948
RB7	88,003	7,78s	4,279/0	134	486	2	0
DP7	862,496	19,67s	9,055/0	381	1,049	1	0
ST5	825,006	1m56.46s	197,130/0	56,843	38,626	156	81,046
H6	1,304,871	7m4,32s	592,584/0	154,270	109,659	446	300,360
L8	9,202,755	9m0.73s	135,590/0	343	891	36	0

Table 6 Results using CV+LS

Ins	NS	Time	N/N2	NSCV	NSLS	Div
EX6-2	67	0.01s	42/32	10	0	16
EX6-1	210	0.01s	8/1	4	3	0
MC2	399	0.01s	37/10	16	11	1
THCB2	500	0.01s	21/0	5	16	0
SHCB2	618	0.01s	39/7	30	1	4
SCH2	816	0.03s	4/0	3	1	0
UPQP	626	0.01s	604/0	229	67	308
G7	1,699	0.03s	44/0	0	0	0
H3	2,080	0.03s	238/0	143	31	44
GP-2	7,374	0.03s	15/3	6	6	3
S4	9,399	0.11s	812/0	172	63	522
H4	40,679	0.38s	6,022/0	2,633	1,749	1,117
MCH5	54,502	0.83s	5,676/0	1,379	1,067	2,552
RB7	88,193	0,65s	4,279/0	134	155	0
DP7	863,130	4,79s	9,055/0	381	594	0
ST5	832,203	5,28s	203,371/0	48,231	51,297	83,105
H6	1,325,308	22.02s	607,026/0	127,106	143,296	309,042
L8	9,203,538	3m45,24s	135,590/0	343	213	0

Table 7 Results for shifted minimum instances using CV

Ins	NS	Time	N/N2	NSCV	Div
MC2	410	0.01s	43/13	23	6
SHCB2	527	0.01s	37/9	24	5
THCB2	538	0.01s	30/8	16	5
SCH2	767	0.02s	5/0	3	0
H3	1,689	0.02s	287/0	182	69
S4	4,568	0.06s	755/0	291	297
GP-2	7,204	0.05s	19/2	7	2
H4	28,732	0.27s	8,020/0	3,465	3,584
MCH5	63,869	0.93s	12,588/0	4,850	6,673
G7	278,326	3.25s	816/0	1	0
RB7	666,437	4.09s	21,978/0	1,522	0
ST5	1,081,443	5.69s	373,285/0	144,405	197,453
L8	1,168,435	27.79s	9,904/0	45	0
H6	2,918,326	39.61s	1,681,477/0	664,752	969,207
DP7	6,250,171	34.58s	58,569/0	3,303	0

border facets due to $0 \notin g$. LP or LS found additional monotonic directions for instances SCH2 and MC2, but this does not lead to a reduction in the number of evaluated simplices because we could not remove the simplex.

The largest reduction in number of evaluated simplices compared with the CV method are realized for instances H6 and ST5. The CV+LP method evaluates 983,155 and 210,915 simplices less respectively and the CV+LS requires 962,718 and 203,718 simplices less. As expected, the running time using CV+LP is the much larger. For H6 and ST5 it needs minutes whereas the CV and CV+LS finishes in seconds. The CV and CV+LS method spend a similar time on the ST5 instance, but the CV+LS method requires 22s and CV 30s on H6 instance.

The instance with the largest number of evaluated simplices is L8 where the number of found/existing monotonic direction is small. Therefore, in such case trying to find a monotonic direction mainly implies an increase of the execution time.

5.1 Result for shifted minimum instances

In previous experiments the minimum point is reached quickly. This is an advantage for the RangeUp test. However, the symmetry of the minimum point can lead to the so-called clustering problem where several simplices share the minimum point [4, 11].

In order to study how results are affected when the minimum point is not at the centroid of an initial facet, the initial simplex was shifted towards one of the vertices by 0.1. For the GP2 instance, the resulting vertex set is $\mathcal{V} = \{(-0.55, -2.1), (1.45, -2.1), (0.45, -0.1)\}$. Tables 7 to 9 show the numerical results with headers as described in Table 3.

The number of evaluated simplices is for some instances larger and for others lower, but the overall view does not change. CV+LP usually results in a low number of evaluated simplices against a large running time. CV+LS generates a similar number of evaluated simplices as CV+LP, but requires less execution time. For instance H6, CV+LS requires 29s, CV 40s and CV+LP 8m33s.

Table 8 Results for shifted minimum instances using CV+LP

Ins	NS	Time	N/N2	NSCV	NSLP	NLPF	Div
MC2	406	0.01s	43/13	23	3	0	6
SHCB2	517	0.01s	34/6	24	4	0	2
THCB2	525	0.01s	22/0	16	6	0	0
SCH2	767	0.02s	5/0	3	2	0	0
H3	1,582	0.05s	205/0	144	37	0	21
S4	4,284	0.28s	550/0	175	98	0	149
GP-2	7,202	0.04s	19/2	7	2	0	2
H4	24,954	2.00	5,149/0	1,782	1,045	2	1,707
MCH5	54,448	3.81s	5,113/0	1,404	1,021	0	1,957
G7	278,325	4.77s	816/0	1	2	0	0
RB7	665,421	40.05s	21,978/0	1,522	1,460	3	0
ST5	856,452	3m23,58s	206,070/0	55,852	43,215	307	83,784
L8	1,168,075	49.66s	9,904/0	45	375	3	0
H6	1,671,447	8m32,81s	668,447/0	167,072	138,256	1,070	325,114
DP7	6,246,242	2m9.34s	58,569/0	3,303	5,317	28	0

Table 9 Results for shifted minimum instances using CV+LS

Ins	NS	Time	N/N2	NSCV	NSLS	Div
MC2	406	0.01s	43/13	23	7	6
SHCB2	517	0.01s	34/6	24	4	2
THCB2	525	0.01	22/0	16	6	0
SCH2	767	0.02s	5/0	3	2	0
H3	1,579	0.02s	201/0	140	39	19
S4	4,279	0.06s	559/0	167	134	141
GP-2	7,202	0.03	19/2	7	10	2
H4	24,713	0.24s	4,950/0	1,517	1,301	1,564
MCH5	56,341	0.86s	6,701/0	1,671	1,308	2,843
G7	278,325	3.32s	816/0	1	5	0
RB7	666,170	4.65s	21,978/0	1,522	570	0
ST5	862,478	5.45s	211,311/0	46,768	56,732	85,186
L8	1,168,406	28.56s	9,904/0	45	106	0
H6	1,743,215	28.77s	726,296/0	150,325	178,685	359,355
DP7	6,249,189	36.87s	58,569/0	3,303	3,159	0

For those instances where CV shows the smallest running time, CV+LP and CV+LS have a small value of NSLP, NSLS respectively in comparison with N/N2, which means that small number of monotonic direction exists or few monotonic direction was found. This implies a large running time.

More rejection/reduction tests, as those used in box partitioning, may be developed in simplicial partitioning. Additionally, over-estimations in g can be reduced by covering the simplex by more boxes instead of just one.

6 Conclusions

From the experimental results we can conclude that the use of the studied local search for monotonic directional derivatives requires a similar number of simplex evaluations as finding monotonic directions by Linear Programming (LPsolve). However, it requires less execution time. An advantage of the local search (LS) is that it can stop when a monotonic direction is found, i.e. we do not need the steepest one, and also it may use the stopping criterion. In some cases, the number of evaluated simplices is smaller using LS than LP, because the later only looks for negative directional derivatives. Looking for positive ones requires running another LP model.

There exist instances for which the CV method results in the smallest execution time. The differences with CV+LS are small in these cases.

Additional search for monotonic directional derivatives by LP or LS, results in an additional small number of them found (they may not exist), incurring in an increase of the computational burden. However, for H6, the LS method saves 27% of the running time in comparison with using only the CV method.

Conflict of interest The authors declare that they have no conflict of interest.

Statements and Declarations The manuscript has no associated data. The used test instances are referenced in the text.

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/>.

A unconstrained portfolio as QP problem

Convex, $n = 4$.

$$\min f(x) = 10x^T Ax - \mu^T x, \tag{8}$$

$$\text{s.t. } \sum x_i \leq 1, x \geq 0 \tag{9}$$

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

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

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

References

1. Berenguel, J.L., Casado, L., García, I., Hendrix, E.: On estimating workload in interval branch-and-bound global optimization algorithms. *J. Global Optim.* **56**(3), 821–844 (2013). <https://doi.org/10.1007/s10898-011-9771-5>
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: *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
3. Casado, L.G., Hendrix, E.M.T., García, I.: Infeasibility spheres for finding robust solutions of blending problems with quadratic constraints. *J. Global Optim.* **39**(4), 577–593 (2007). <https://doi.org/10.1007/s10898-007-9157-x>
4. Du, K., Kearfott, R.B.: The cluster problem in multivariate global optimization. *J. Global Optim.* **5**(3), 253–265 (1994)
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)
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.* (2021)
7. Hansen, E., Walster, W.: *Global Optimization Using Interval Analysis*, 2nd edn. Marcel Dekker Inc., New York (2004)
8. Hendrix, E., Salmerón, J., Casado, L.: On function monotonicity in simplicial branch and bound. In: *LeGO 2018*, p. 4. Leiden (The Netherlands) (2018). <https://doi.org/10.1063/1.5089974>
9. 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>
10. Karhbet, S.D., Kearfott, R.B.: Range bounds of functions over simplices, for branch and bound algorithms. *Reliab. Comput.* **25**, 53–73 (2017). *Reliab.-Comput.-J*
11. Kearfott, B., Du, K.: *The Cluster Problem in Global Optimization: the Univariate Case*, pp. 117–127. Springer, Vienna (1993). https://doi.org/10.1007/978-3-7091-6918-6_10
12. Li, Q., McKenzie, D., Yin, W.: From the simplex to the sphere: faster constrained optimization using the hadamard parametrization (2021). [arxiv:2112.05273](https://arxiv.org/abs/2112.05273)
13. Moore, R.E., Kearfott, R.B., Cloud, M.J.: *Introduction to interval analysis*. Soc. Ind. Appl. Math. (2009). <https://doi.org/10.1137/1.9780898717716>
14. Paulavičius, R., Žilinskas, J.: *Simplicial Global Optimization*. Springer, New York (2014)
15. Rall, L.B. (ed.): *Examples of software for automatic differentiation and generation of Taylor coefficients*, pp. 54–90. Springer, Berlin (1981)

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