Computational Optimization and Applications (2025) 90:691–720
https://doi.org/10.1007/s10589-025-00652-5

On the use of restriction of the right-hand side in spatial branch-and-bound algorithms to ensure termination

Peter Kirst[1] · **Christian Füllner**[2]

Received: 12 October 2023 / Accepted: 16 January 2025 / Published online: 8 February 2025

Abstract

Spatial branch-and-bound algorithms for global minimization of non-convex problems require both lower and upper bounding procedures that finally converge to a globally optimal value in order to ensure termination of these methods. Whereas convergence of lower bounds is commonly guaranteed for standard approaches in the literature, this does not always hold for upper bounds. For this reason, different so-called convergent upper bounding procedures are proposed. These methods are not always used in practice, possibly due to their additional complexity or possibly due to increasing runtimes on average problems. For that reason, in this article we propose a refinement of classical branch-and-bound methods that is simple to implement and comes with marginal overhead. We prove that this small improvement already leads to convergent upper bounds, and thus show that termination of spatial branch-and-bound methods is ensured under mild assumptions.

Keywords global optimization · branch-and-bound · upper bounding procedure · feasible points · feasibility verification · restriction of the right-hand side

Mathematics Subject Classification 90C26

1 Introduction

In this article, we address the computation of upper bounds in spatial branch-and-bound algorithms in global optimization, as well as the termination of these algorithms. In

✉ Peter Kirst
peter.kirst@wur.nl

Christian Füllner
christian.fuellner@kit.edu

1 Operations Research and Logistics (ORL), Wageningen University & Research (WUR), Wageningen, The Netherlands

2 Institute for Operations Research, Stochastic Optimization, Karlsruhe Institute of Technology, Karlsruhe, Germany

this context, we consider problems of the form

$$\begin{aligned} P(B): \quad & \min_{x\in\mathbb{R}^n} f(x) \\ \text{s.t.} \quad & g_i(x) \le 0, \quad i \in I, \\ & x \in B \end{aligned}$$

with a finite set I. The box B is defined by $B = \left\{x \in \mathbb{R}^n | \underline{b} \le x \le \overline{b}\right\}$ with $\underline{b}, \overline{b} \in \mathbb{R}^n$, $\underline{b} < \overline{b}$, where inequalities are understood component-wise. We assume the functions f and g_i to be continuous, but we do not require f and g_i to be convex. For that reason, the feasible set

$$M(B) := \{x \in B \mid g_i(x) \le 0, i \in I\}$$

does not need to be convex either. In total, problem $P(B)$ is a non-convex problem. Throughout this article the definition $M(X) := X \cap M(B)$ will be convenient for some box $X \subset B$.

For a problem of type $P(B)$ and some predefined optimality tolerance $\varepsilon_v > 0$ a typical aim in global optimization is to determine a so-called ε_v-optimal feasible point $x^* \in M(B)$, i.e. a feasible point x^* with

$$f(x^*) \le f(x) + \varepsilon_v$$

for all $x \in M(B)$. Clearly, this immediately implies $v^* \le f(x^*) \le v^* + \varepsilon_v$ where v^* denotes the globally minimal value of $P(B)$. Note that we impose hard constraints here, which means that approximately feasible points satisfying the ε_v-optimality criterion are not accepted.

The most common approach to globally solve problems of type $P(B)$ in this sense is to apply spatial branch-and-bound algorithms. In such methods, the problem is iteratively branched into subproblems $P(X)$ of the form

$$P(X): \min_{x\in\mathbb{R}^n} f(x) \quad \text{s.t.} \quad x \in M(X)$$

with sub-boxes $X \subset B$. Then, for those subproblems lower bounds are constructed and, furthermore, overall lower bounds at the globally minimal value v^* of the original problem $P(B)$ are computed as a minimum of all these lower bounds. Wherever possible, boxes that cannot contain globally optimal points are excluded from the search space. In addition, upper bounds at globally minimal values are computed and the algorithm terminates if lower and upper bounds are sufficiently close to each other.

Whereas convergence is typically ensured for lower bounds, this is not guaranteed for commonly used upper bounding procedures. Upper bounds for v^* can be constructed by explicitly evaluating the objective function at feasible points of $P(B)$ or by applying local solvers, which implicitly make use of such evaluation. However, as the problem $P(B)$ is non-convex, finding a feasible point is already NP-hard, which makes it challenging to generate a sequence of feasible points that lead to improved upper bounds. Therefore, most research on upper bounding procedures focuses on heuristics which perform sufficiently well for many practical applications, but are not guaranteed to ensure convergence of upper bounds in spatial branch-and-bound methods in general [10, 11].

A popular strategy is the aforementioned approach to solve the non-convex problem $P(B)$, or some subproblem $P(X)$, locally [7]. Although this often works well in practice, in general this does not *guarantee* sufficiently good upper bounds for termination of branch-and-bound algorithms. Since exact feasibility is hard to ensure, another common concept is to accept so-called ε_f-feasible points, i.e. points $x \in B$ with $g_i(x) \leq \varepsilon_f, i \in I$, for some tolerance $\varepsilon_f > 0$. However, this concept is not sufficient to compute valid upper bounds for v^* either. This even holds for ε_f close to zero, as discussed in detail by Tuy [56] and Kirst et al. [30].

Hence, whereas for most lower bounding procedures in the literature certain convergence results are available, unfortunately, this does not hold for classical upper bounding procedures. Since spatial branch-and-bound algorithms rely on convergent valid upper bounds in the termination criterion, however, such convergence guarantees for the upper bounds are crucial to ensure termination after a finite number of iterations.

Research in this direction has been limited so far. As discussed above, computing upper bounds for v^* and identifying feasible points are closely related. Therefore, the work that does exist is mostly focused on feasibility verification. Once verification is successful for some box X, valid upper bounds are obtained by computing an upper bound for the objective function over X. In [13, 28, 29] several different feasibility verification methods are presented. They are based on computing approximately feasible points, e.g., by using conventional nonlinear solvers, and then verifying the existence of feasible points in specifically constructed boxes around such points using interval Newton methods. While these methods are rigorous in the sense that they rule out false positive feasibility verification, and thus do yield valid upper bounds for v^*, there exist no proven convergence guarantees. In our emphasis on the convergence of the upper bounding procedure, our work clearly differs from these methods.

For the case of purely inequality-constrained and box-constrained problems, a convergent upper bounding procedure is presented in [30] based on perturbing infeasible iterates along Mangasarian-Fromovitz directions. It is not straightforward to extend this approach to equality-constrained problems, though. In reverse, for the case of purely equality-constrained and box-constrained problems, a convergent upper bounding procedure is presented in [19] based on a generalization of Miranda's Theorem [37]. This method, however, does not allow for inequality constraints in problems $P(B)$ and requires the box constraints to be *strictly* satisfied. An extension to problems that also include inequality constraints is presented in [18] based on utilizing approximations of active index sets of inequalities.

Another common drawback of the existing upper bounding procedures with proven convergence in the literature is that they are rather technical and often tedious to implement. On the contrary, some very simple upper bounding procedures, such as starting non-linear solvers at different points during the solution process, are not guaranteed to ensure convergence of spatial branch-and-bound algorithms in general, but provide sufficiently good upper bounds to achieve termination for many problems in practice.

In this article, we propose a new upper bounding procedure that exploits the strengths and overcomes the weaknesses of both types of approaches. That is, our proposed procedure

- is simple to implement,
- is computationally efficient in the sense that there is at most little overhead,
- can be combined easily with other common methods such as the local solution of the problem at hand,
- *but* is still proven to provide sufficiently good upper bounds in order to terminate the algorithm after a finite number of iterations under mild assumptions.

Note that, although some illustrative computational examples are provided using a simple implementation as a proof-of-concept, it is not our aim to develop an entire new solver. Instead, we focus on a new upper bounding procedure that can be incorporated in a wide variety of solvers.

The main idea in this article is based on the concept of *restriction of the right-hand side*, which has recently been proposed in [40, 41] in the context of semi-infinite programming to compute feasible points. Whereas this technique is used for (standard as well as generalized) semi-infinite programs in [40, 41], to our best knowledge it has never been examined for standard non-convex problems in global optimization.

This article is structured as follows. In Sect. 2 we briefly review some basic concepts from global optimization and further discuss difficulties of non-convergent upper bounding procedures. In Sect. 3 we explain how the concept of restriction of the right-hand side from the literature can be applied within a spatial branch-and-bound method for continuous non-linear global optimization. Based on this, in Sect. 4 we prove that this leads to a convergent algorithm given some assumptions, which are discussed in Sect. 5. In Sect. 6 we provide computational results for some illustrative test problems, which highlight that, while providing proven convergence guarantees, the proposed method has little computational overhead. Finally, Sect. 7 concludes the paper with some final remarks.

The notation in this article is standard. In particular, Df denotes the row vector of partial derivatives of a function f and by $\mathrm{diag}(X)$ we denote the diagonal length of a box X.

2 Preliminaries and assumptions

In this section we briefly review some important concepts from the literature that are needed for our approach. We start with a small overview on spatial branch-and-bound methods in global optimization where we focus in particular on lower bounding procedures. A general definition of a convergent lower bound taken from [30] is described in Sect. 2.2. The concept of restriction of the right-hand side, which is the basis for our new upper bounding procedure, is briefly explained in Sect. 2.3.

2.1 Lower bounding procedures for spatial branch-and-bound methods

Spatial branch-and-bound algorithms for global optimization were first proposed by Falk and Soland in [16]. Since then, various enhancements have been developed, for instance, branch-and-reduce [47, 48], symbolic branch-and-bound [51, 52], branch-and-contract [59] or branch-and-cut [55]. In addition to these purely scientific works, many state-of-the-art global optimization solvers are based on implementations of spatial branch-and-bound algorithms, for example BARON [49], COUENNE [9], ANTIGONE [39], LINDOGLOBAL [33] or SCIP [1, 57]. For monographs covering the theory on global optimization and, in particular, spatial branch-and-bound algorithms we refer to [17, 23, 34].

Typically, in spatial branch-and-bound algorithms, lower bounds for v^* are obtained by special bounding procedures. A well-known approach is to compute lower bounds by solving convex relaxations of problems $P(X)$ to optimality. Determining these relaxations, in turn, is based on computing as tight as possible convex underestimators of the functions f and g_i, $i \in I$, contained in $P(B)$. Many different underestimators have been proposed for specific classes of functions, among them underestimators and envelopes for bilinear terms [35], polynomials [32], non-convex piecewise linear functions [24] and general lower-semicontinuous functions [53, 54]. Additionally, generic convex underestimators can be constructed for arbitrary non-convex functions, as described in [2, 21, 36] and applied in the αBB algorithm [3, 4, 7]. Since tight underestimators can only be determined explicitly for functions in low dimensions, these techniques are usually combined with factorization and symbolic reformulation approaches [35, 52]. However, recently there has also been some progress on obtaining tight relaxations for composite functions directly [22]. Moreover, convex underestimators can often be considerably strengthened by bounds tightening techniques [9, 44]. Different lower bounding procedures are based on exploiting duality [14, 15], using piecewise linear approximations [20, 38, 45, 46], using Lipschitz constants [43] or applying interval arithmetic [42] and related concepts, such as centered forms [8, 31]. For several of those lower bounding procedures it is proven that the determined lower bounds converge to v^* for decreasing box sizes, as they naturally occur in spatial branch-and-bound algorithms.

2.2 Convergence of lower bounding procedures

In this article we assume that convergent lower bounding procedures are available, which is commonly fulfilled for the aforementioned approaches. However, in order to keep the exposition as general as possible we briefly review some definitions from [30], which will be convenient throughout this article. Furthermore, this enables us to prove convergence in a rather general manner without restricting our consideration to a particular lower bounding procedure. We start with some special classes of bounding procedures.

Definition 1 (Bounding procedures, from [30])

- A function ℓ from the set of all sub-boxes X of B to $\mathbb{R}$ is called *M-dependent lower bounding procedure* for the objective function of $P(B)$, if $\ell(X) \le \inf_{x \in M(X)} f(x)$ holds for all sub-boxes $X \subseteq B$ and any choice of the functions $f, g_i, i \in I$.
- A function ℓ from the set of all sub-boxes X of B to $\mathbb{R}$ is called *M-independent lower bounding procedure* for a function, if it satisfies $\ell(X) \le \min_{x \in X} \phi(x)$ for all sub-boxes $X \subseteq B$ and any choice of the function $\phi : B \to \mathbb{R}$.
- A lower bounding procedure ℓ is called monotone, if $\ell(X_1) \ge \ell(X_2)$ holds for all boxes $X_1 \subseteq X_2 \subseteq B$.

In the following, by ℓ_ϕ we denote an M-independent lower bounding procedure applied to a specific function ϕ, e.g. f or $g_i, i \in I$.

To make sure that our spatial branch-and-bound algorithm converges, it is crucial that all applied lower bounding procedures are convergent. To define the concept of convergent lower bounding procedures, we consider so-called *exhaustive sequences of boxes* and apply the bounding procedures to these sequences. A sequence of boxes $(X_k)_{k \in \mathbb{N}}$ is called exhaustive, if it is nested ($X_k \subset X_{k-1}$ for all $k \in \mathbb{N}$), contains no empty boxes ($X_k \ne \emptyset$ for all $k \in \mathbb{N}$) and satisfies $\lim_{k \to \infty} \mathrm{diag}(X_k) = 0$. We refer to [23] for more information.

For commonly applied box division strategies in global optimization, e.g., dividing a box along the midpoint of a longest edge, any exhaustive sequence of boxes $(X_k)_{k \in \mathbb{N}}$ converges to a single point $\tilde{x}$, i.e. we have $\bigcap_{k \in \mathbb{N}} X_k = \{\tilde{x}\}$. Note that under continuity of f, this implies

$$\lim_{k \to \infty} \min_{x \in X_k} f(x) = f(\tilde{x}),$$

and in particular the limit exists [30]. We can now introduce the notion of convergent bounding procedures.

Definition 2 (Convergent bounding procedures, from [30])

- An M-independent lower bounding procedure ℓ_ϕ is called *convergent* if it satisfies

  $$\lim_{k \to \infty} \ell_\phi(X_k) = \lim_{k \to \infty} \min_{x \in X_k} \phi(x)$$

  for any exhaustive sequence of boxes $(X_k)_{k \in \mathbb{N}}$ and any function $\phi : B \to \mathbb{R}$.
- An M-dependent lower bounding procedure ℓ is called *convergent* if it satisfies

  $$\lim_{k \to \infty} \ell(X_k) = \lim_{k \to \infty} \min_{x \in M(X_k)} f(x),$$

  where $\lim_{k \to \infty} \min_{x \in M(X_k)} f(x) = +\infty$ if $\tilde{x} \notin M(B)$, for any exhaustive sequence of boxes $(X_k)_{k \in \mathbb{N}}$.

2.3 Restriction of the right-hand side

Convergence of lower bounds of many spatial branch-and-bound algorithms in global optimization is ensured. Similarly, it is straightforward to compute a sequence of points

$(x_k)_{k\in\mathbb{N}}$ that possesses a convergent subsequence $(x_{k_\nu})_{\nu\in\mathbb{N}}$ such that $\lim_{\nu\to\infty} x_{k_\nu} = x^*$, where x^* denotes a globally minimal point of the problem $P(B)$. This follows immediately from the theory on global optimization (see, e.g., [23]) and is ensured under mild assumptions. Depending on the exact type of algorithm, for instance, the midpoints of the boxes that are currently examined in the branch-and-bound framework possess a subsequence that fulfills this requirement, as we shall see in Sect. 4. In contrast, similar convergence results for upper bounds have not been established in general, as discussed in Sect. 1.

In this article, we are concerned with the computation of convergent upper bounds. As already discussed before, computing valid upper bounds is closely related to finding feasible points because evaluation of the objective function f at a feasible point immediately yields an upper bound for v^*. Therefore, our new upper bounding procedure is based on an efficient method to generate feasible points of problem $P(B)$.

As a key idea, our approach draws on a technique called *restriction of the right-hand side*, which is introduced in [40, 41] for (generalized) semi-infinite problems. Using this technique, we introduce some tolerance in the constraints. Importantly, in contrast to the concept of ε_f-feasibility, here it is used to further restrict them. Hence, we require $g_i(x) \le -\varepsilon_f$ for all constraints g_i. Clearly, a point satisfying these conditions is also feasible for $P(B)$.

However, if ε_f is chosen too large, the feasible set can become empty. Moreover, using a fixed value of ε_f in general does not lead to a convergent upper bounding procedure. For that reason, in order to combine restriction of the right-hand side with a standard branch-and-bound algorithm in global optimization, a crucial component of our procedure is to drive ε_f to zero in an appropriate way. As we shall see throughout this article, this can be achieved such that convergence of upper bounds is ensured, and thus branch-and-bound algorithms are proven to terminate.

We remark that in [40, 41] in presence of semi-infinite constraints several nonlinear problems have to be solved from scratch using different parameter settings to obtain the desired upper bounds. However, in standard global optimization, i.e. in the absence of semi-infinite constraints, it is not possible to solve several optimization problems from scratch in order to compute the solution of a single optimization problem, since this is by far too expensive from a computational point of view. Therefore, in this article we propose to carefully incorporate the concept of restriction of the right-hand side into a spatial branch-and-bound algorithm, such that we get along without this requirement. Essentially, as we shall see, this incorporation results in a different box selection rule in the branch-and-bound algorithm.

3 Incorporation of restriction of the right-hand side into spatial branch-and-bound methods

In this section, we propose a new upper bounding procedure which is convergent. It is based on finding feasible points of $P(B)$ with guarantee and then evaluating the objective function f in such points. To achieve this, the main idea is to exploit convergence of subsequences $(x_{k_i})_{i\in\mathbb{N}}$ for a slightly altered problem.

More precisely, given a parameter $\delta > 0$, we consider a problem of the form

$$\begin{aligned} P(B,\delta): \quad & \min_{x\in\mathbb{R}^n} f(x) \\ \text{s.t.} \quad & g_i(x) \le -\delta, \quad i \in I, \\ & x \in B. \end{aligned}$$

We refer to this as the *restricted problem*. The feasible set of $P(B,\delta)$ is denoted by $M(B,\delta)$ and, moreover, let $\widetilde{x}$ denote a globally optimal point of $P(B,\delta)$.

According to the explanation at the beginning of this section, given that δ is chosen carefully such that $M(B,\delta)$ still contains a feasible point, by applying a branch-and-bound algorithm a subsequence $(x_{k_\nu})_{\nu\in\mathbb{N}}$ is generated that satisfies $\lim_{\nu\to\infty} x_{k_\nu} = \widetilde{x}$. This implies $g_i(\widetilde{x}) \le -\delta$ and due to continuity of the functions g_i there exists some $\widetilde{\nu}$ such that for all $\nu \ge \widetilde{\nu}$ we have $g_i(x_{k_\nu}) < 0$. Thus, as the box constraints are fulfilled as well, x_{k_ν} is feasible for the original problem $P(B)$, and we obtain an upper bound for v^* by evaluating the objective function f in x_{k_ν}. Note that the main difference to applying the same reasoning to the original problem $P(B)$ is that by restriction of the right-hand side we ensure that a feasible point satisfying $g_i(x) \le 0$ is found after a finite number of steps and not only in limit.

Simply applying a spatial branch-and-bound algorithm to the problem $P(B,\delta)$ instead of the original problem $P(B)$ is still not sufficient for our purpose. In particular, in that case the obtained lower bounds converge to the optimal value of the restricted problem which may differ from the value of interest v^*. This is clearly not desired. Moreover, simply using some fixed $\delta > 0$ is not sufficient to obtain a sequence of upper bounds that converges to v^* as upper bounds may remain too large in such a setting.

For that reason, our main idea is to combine the solution of the original problem $P(B)$ and the use of restriction of the right-hand side in a reasonable way. To this end, we consider a standard spatial branch-and-bound algorithm for solving $P(B)$ and, in addition, incorporate the solution of more restricted problems for different values of δ. This is explained in the following and stated formally in Algorithm 1.

As is common for spatial branch-and-bound algorithms in global optimization, each iteration k is started by choosing a tuple (X_k, v_k) from a list $\mathcal{L}$ of boxes still to explore, with v_k a lower bound for the optimal value $v^*(X_k)$ on X_k (Step 1). In every second iteration, we follow the selection rule of classical spatial branch-and-bound algorithms in global optimization, which typically means that a box with the smallest lower bound is chosen, as it appears most promising to contain a globally minimal point x^*. We refer to this as a *normal selection* step.

In the remaining iterations, we only choose boxes X_k that may contain feasible points for the restricted problems $P(X_k,\delta)$, thus aiming at the solution of problems $P(B,\delta)$. This is checked by computing lower bounds $\ell_{g_i}(X_k)$ for all g_i, $i \in I$, on X_k using lower bounding procedures, such as interval arithmetic [42]. In case that no such box can be selected, we reduce the value $\delta > 0$ in order to refine the approximation obtained by restriction of the right-hand side. The same refinement is applied if a feasible point for $P(B)$ is found (see Step 4) in order to ensure convergence of the upper bounds to v^*. We refer to this as a *restricted selection* step.

Algorithm 1 Branch-and-bound framework

Input: Tolerance $\varepsilon_v > 0$, initial $\delta_0 > 0$, initial lower bound $\widehat{v}_0 = -\infty$, initial upper bound $u_0 = +\infty$, multiplier $\gamma \in (0, 1)$ and iteration counter $k = 1$.

Initialization: Set list $\mathcal{L} = \{(B, -\infty)\}$.
while $u_{k-1} - \widehat{v}_{k-1} > \varepsilon_v$ and $\mathcal{L} \neq \emptyset$ **do**
  Step 1: Choose tuple $(X_k, v_k) \in \mathcal{L}$ as follows:
  if k is even **then**
    Restricted selection. Compute

$$\overline{\mathcal{L}} := \left\{ (X_k, v_k) \in \mathcal{L} \mid \max_{i \in I} \ell_{g_i}(X_k) \leq -\delta_k \right\}.$$

    if $\overline{\mathcal{L}} = \emptyset$ **then**
      Set $\delta_{k+1} := \gamma \delta_k$.
      Increment k.
      Go to Step 1.
    end if
    Choose $(X_k, v_k) \in \overline{\mathcal{L}}$ with $v_k = \min\{v_k \mid (X_k, v_k) \in \overline{\mathcal{L}}\}$.
  else
    Normal selection. Choose $(X_k, v_k) \in \mathcal{L}$ with $v_k = \widehat{v}_{k-1}$.
  end if
  Step 2: Divide X_k along midpoint of a longest edge into X_k^1 and X_k^2 and remove tuple (X_k, v_k) from the list $\mathcal{L}$.
  Step 3: For $j = 1, 2$, compute lower bounds $\ell(X_k^j)$ for $v^*(X_k^j)$.
  if $\ell(X_k^j) < \infty$ **then**
    Add the pair $(X_k^j, \ell(X_k^j))$ to the list $\mathcal{L}$.
  end if
  Step 4: For $j \in \{1, 2\}$ choose $x_k^j \in X_k^j$ and define

$$f_k^j := \begin{cases} f(x_k^j) & \text{if } x_k^j \in M(X_k^j) \text{ (or, equivalently, } x_k^j \in X_k^j \text{ and } g_i(x_k^j) \leq 0, i \in I) \\ +\infty & \text{else.} \end{cases}$$

  if $x_k^j \in M(X_k^j)$ for some $j \in \{1, 2\}$ **then**
    Set $\delta_{k+1} := \gamma \delta_k$.
  end if
  Step 5: Set $u_k = \min\{u_{k-1}, f_k^1, f_k^2\}$ and choose $x_k^* \in \{x_{k-1}^*, x_k^1, x_k^2\}$ with $f(x_k^*) = u_k$.
  Step 6: Fathoming:
  for $(X, v) \in \mathcal{L}$ with $v > u_k$ **do**
    Remove (X, v) from $\mathcal{L}$.
  end for
  Step 7: Update of lower bound:
  if $\mathcal{L} \neq \emptyset$ **then**
    $\widehat{v}_k = \min\{v \in \mathbb{R} \mid (X, v) \in \mathcal{L}\}$.
  end if
  Step 8: Increment k.
end while

Output: Globally ε_v-optimal point x^* (if it exists).

We should make three important remarks at this point. First, for simplicity we assume that a restricted selection step based on restriction of the right-hand side is executed in every second iteration. However, for our following convergence results it is only important that such step is always executed after a finite number of iterations with normal selection. Therefore, much more sophisticated schemes are possible as well. For instance, in our computational tests in Sect. 6 we use restricted selection steps every κ-th iteration for different values of κ with $\kappa \in \mathbb{N}, \kappa > 1$.

Second, we should emphasize that deviations from the standard procedure in spatial branch-and-bound algorithms have to be correctly designed in order to not compromise the convergence behavior. For instance, as shown by Dickinson [12], alternating between the standard division rule (division along the longest edge) and free box division rules may prevent the emergence of exhaustive sequences of boxes. Importantly, in our proposed method, we alternate between different box *selection* rules, while always dividing along the longest edge. In this case, exhaustiveness of subsequences of boxes is preserved.

Third, restriction of the right-hand side can be implemented efficiently by computing the value $\delta(X_k) := \max_{i \in I} \ell_{g_i}(X_k)$ just *once* when box X_k is constructed, as this value is independent of δ_k. It can then be stored together with the tuple (X_k, v_k). In the restricted selection step, $\delta(X_k)$ then only has to be compared to the current δ_k for all $(X_k, v_k) \in \mathcal{L}$ instead of recomputing it for each element in $\mathcal{L}$.

Steps 2 to 7 basically remain unchanged from standard spatial branch-and-bound algorithms in global optimization. In Step 2, the selected box X_k is branched into two sub-boxes by division along a longest edge. More sophisticated branching strategies are possible as well. In Step 3, a lower bounding procedure is used to compute lower bounds $\ell(X_k^j)$ for the globally minimal value $v^*(X_k^j)$ of the newly obtained boxes X_k^j, $j \in \{1, 2\}$. As already discussed in Sect. 1, for instance, a convex relaxation of $P(X_k^j)$ can be solved to obtain such a bound. If such a relaxation is infeasible, we set $\ell(X_k^j) = +\infty$. If this is not the case, the corresponding box is added to $\mathcal{L}$. In Step 4, some point $x_k^j \in X_k^j$ is determined and checked for feasibility by evaluating $g_i(x_k^j)$ for all $i \in I$. If $x_k^j \in M(X_k^j)$, then x_k^j and its objective value $f(x_k^j)$ can be used to update the best known point x_k^* and the best known upper bound for u_k for v^* (Step 5). If $x_k^j \notin M(X_k^j)$, then x_k^* and u_k remain unchanged. In Step 6, boxes which can be ruled out as they do not contain a globally minimal point are removed from $\mathcal{L}$ in order to reduce the search space. Then, taking into account that boxes have been added and removed from $\mathcal{L}$ in the current iteration, the global lower bound $\widehat{v}_k$ for v^* is updated (Step 7). As long as the termination criterion is not satisfied, a new iteration is started.

Importantly, Algorithm 1 is rather a conceptual description of a spatial branch-and-bound algorithm in global optimization. Further improvements that are known to work well in practice, for instance, bounds tightening or using additional upper bounding procedures, such as starting a local non-linear solver, can be incorporated as well. By doing so, it is possible to construct a solver that is fast yet proven to terminate after a finite number of iterations as we shall see in the remainder of this article. Furthermore, with respect to the choice of the point x_k^j we are rather generic. In fact, every common selection rule will work, e.g., choosing the midpoint of a box.

We should note that evaluating midpoints of sequences of shrinking boxes shares similarities with the DIRECT algorithm proposed by Jones et al. [25, 26]. However, our proposed method differs in several aspects. In our case, choosing midpoints in Step 4 is sufficient, but not mandatory; a locally optimal point can be used as well. Our proposed method is a spatial branch-and-bound method with proven convergence (see Sect. 4), whereas DIRECT is a pure branching method and a heuristic. Finally, in DIRECT, the next box to partition is selected based on the diagonal lengths of boxes and the objective values at the box midpoints. In our restricted selection steps, the selection is based on restriction of the right-hand side and computing deterministic bounds on constraint functions $g_i, i \in I$.

Worth mentioning, the feasibility check for points x_k^j in Step 4, whereas straightforward on first sight and an integral part of most implementations of spatial branch-and-bound methods, can be rather time-consuming in practice. Moreover, in case of a rigorous implementation, rounding errors need to be taken into account in order to prevent some of the issues related to ε_f-feasible points, see the discussion in Sect. 1, which may complicate this step considerably. While feasibility verification and rigorous implementation are not the main focus of this paper, we should mention that restriction of the right-hand side might also help to facilitate this step. If we check for $x_k^j \in M(X_k^j, \bar{\delta})$ for some $\bar{\delta} > 0$, we may allow for some ε_f-tolerance, but still guarantee a rigorous feasibility check for $P(B)$ as long as $\varepsilon_f < \bar{\delta}$. This might be particularly helpful for points x_k^j obtained by applying local solvers to $P(X_k^j, \bar{\delta})$, as these are often guaranteed to be ε_f-feasible only for the considered problem. As we shall see in Sect. 4.1, given that $\bar{\delta} < \delta$, this does not compromise the convergence of Algorithm 1.

Finally, we illustrate the mechanism of restriction of the right-hand side using an example.

Example 1 The illustrative problem is given by

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \; & f(x) = 6x_1^2 + 2x_2^2 - 60x_1 - 8x_2 + 166 \\ \text{s.t.} \quad & g_1(x) = \frac{1}{4}(x_1 x_2 - x_1 - x_2) \le 0, \\ & g_2(x) = \frac{1}{4}(-x_1 - x_2 + 3) \le 0, \\ & x \in [0, 10] \times [0, 10]. \end{aligned}$$

It is based on problem `zecevic4` from the *COCONUT benchmark* library [50].

After several branching steps, in iteration 26 of Algorithm 1, we obtain a list $\mathcal{L}$ of 25 boxes. These boxes, as well as the level curves of g_1 and g_2 (blue and red lines, respectively), are depicted in Fig. 1. Without restriction of the right-hand side, all these boxes can be chosen in the next step of Algorithm 1. Therefore, all of them are highlighted in yellow.

Using our approach of restriction of the right-hand side, the set of boxes from which the next box can be selected in iteration 26 is reduced to $\overline{\mathcal{L}} \subseteq \mathcal{L}$. This subset depends on parameter $\delta_{26} > 0$. For different values of δ_{26} the resulting lists $\overline{\mathcal{L}}$ are

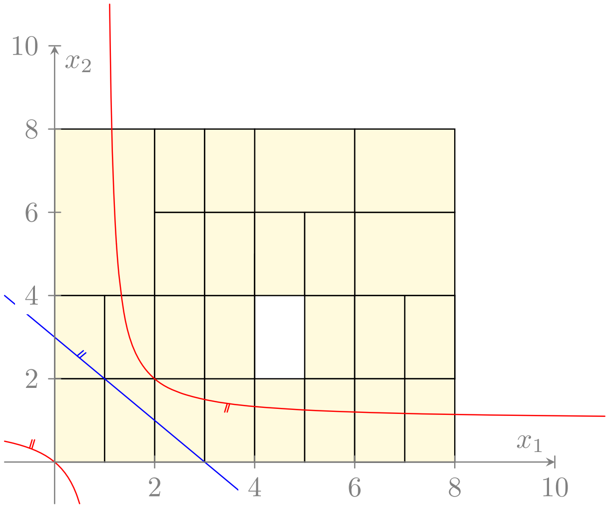

Fig. 1 Level curves and selectable boxes for Example 1 without restriction of the right-hand side

visualized in Fig. 2. It is also illustrated how the choice of δ_{26} affects the considered level curves associated with the functions g_1 and g_2 (depicted in red and blue, whereas the original level curves are depicted in grey). Note that more boxes are contained in $\overline{\mathcal{L}}$ than the ones containing feasible points of the restricted problems, as $\overline{\mathcal{L}}$ is defined using lower bounds for g_1 and g_2, which are computed using centered forms. As we can see, choosing $\delta_{26} = 2$ is too conservative and leads to an empty list $\overline{\mathcal{L}}$. Hence, δ_{26} is decreased in Algorithm 1 for the next iteration.

4 Proof of convergence

In this section, we give a proof of convergence for Algorithm 1. We require that a convergent lower bounding procedure is used, which is a mild assumption, as this is fulfilled for many concepts that are available. This immediately leads to convergence of the overall lower bound that is generated by standard branch-and-bound algorithms, as we describe in Sect. 4.2. Similarly, this holds for the optimal points. However, in addition, we are also able to prove convergence of upper bounds to the globally optimal value v^*. This is presented in Sect. 4.1. Moreover, it will become clear that this ensures finite termination of the algorithm. Feasibility of our assumptions will be discussed in Sect. 5.

4.1 Convergence of upper bounds

In this subsection we state our main result together with the corresponding proof, i.e., we show convergence for the sequence of upper bounds generated in Algorithm 1 for $\varepsilon_v = 0$. In fact, we show that this sequence converges to the globally optimal value v^*

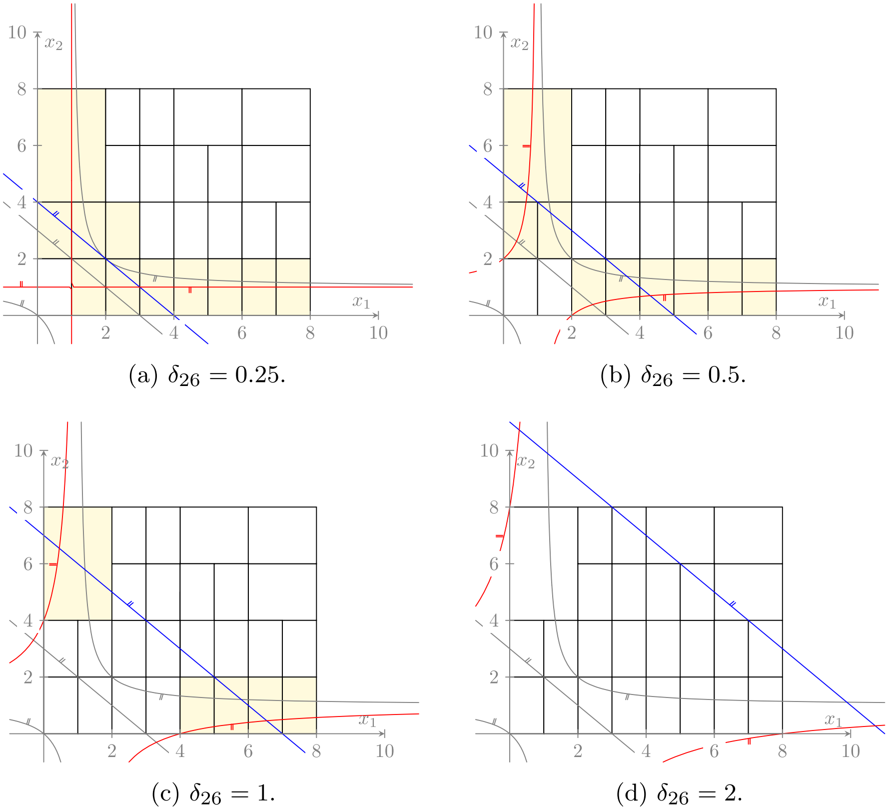

Fig. 2 Level curves and selectable boxes in $\overline{\mathcal{L}}$ for Example 1 and different choices of δ_{26}

under mild assumptions. This is needed in order to ensure that the algorithm terminates after a finite number of iterations for any termination tolerance $\varepsilon_v > 0$, as we shall see in the next subsection.

To achieve convergence of upper bounds, we require that all lower bounds in Algorithm 1 are generated by convergent and monotone M-independent and M-dependent lower bounding procedures, respectively. We formalize this requirement in the following assumption.

Assumption 1 We assume that

- the M-independent lower bounding procedure $\ell_{g_i}, i \in I$, used in Step 1 of Algorithm 1 is convergent and monotone,
- the M-dependent lower bounding procedure ℓ used in Step 3 of Algorithm 1 is convergent and monotone.

Note that both requirements can be satisfied by applying standard lower bounding procedures from the literature, such as interval arithmetic [42], centered forms [8] or using αBB underestimators [2] among others.

With respect to problem $P(B)$, our main assumption is the following type of Slater condition.

Assumption 2 We assume that in every neighborhood $N(x^*)$ of every globally optimal point x^* of $P(B)$ there exists a Slater point, i.e. a point x satisfying $g_i(x) < 0$ for all $i \in I$.

While the former assumption is standard in global optimization anyway, the latter is needed in order to ensure that for small values of δ the problems $P(B, \delta)$ and $P(B)$ do not differ too much, as shown in the following lemma. We discuss feasibility of these requirements in more detail in Sect. 5.

Lemma 1 *Let $(\delta_k)_{k\in\mathbb{N}}$ be a sequence with $\lim_{k\to\infty} \delta_k = 0$ and $\delta_k > 0$ for all $k \in \mathbb{N}$. Moreover, let $M(B, \delta_k) \neq \emptyset$ for all $k \in \mathbb{N}$ and let us assume that the Slater condition in Assumption 2 holds for the problem $P(B)$. Then, there exists a sequence of optimal values $(v_k^*)_{k\in\mathbb{N}}$ of $P(B, \delta_k)$ such that*

$$\lim_{k\to\infty} (v_k^* - v^*) = 0.$$

Proof First of all, note that in view of $M(B, \delta_k) \neq 0$ and the fact that $M(B, \delta_k)$ is compact, there exists a sequence of optimal values $(v_k^*)_{k\in\mathbb{N}}$ of $P(B, \delta_k)$. Furthermore, the sequence is bounded below by v^* and monotonically decreasing and, hence, convergent.

Next, we shall prove that for every $d > 0$ there exists some $\bar{k}$ such that $|v_k^* - v^*| \leq d$ for all $k \geq \bar{k}$. To this end, we assume that the assertion does not hold and derive a contradiction. Hence, we require that there is some $\bar{v} > v^*$ such that for all $\bar{k} \in \mathbb{N}$ there is some $k \geq \bar{k}$ such that $v_k^* > \bar{v}$. Now consider a Slater point $\widehat{x}$ in a sufficiently small neighborhood $N(x^*)$ around an optimal point x^* such that $f(\widehat{x}) \leq \bar{v}$. Note that due to Assumption 2 and continuity of the defining functions such a point $\widehat{x}$ exists.

Next, we set $g := \max_{i\in I} g_i(\widehat{x}) < 0$. For some $\delta > 0$ with $g < -\delta$ we conclude that $\widehat{x}$ is feasible for problem $P(B, \delta)$. Due to $\lim_{k\to\infty} \delta_k = 0$ there is some $\bar{k}$ such that $g < -\delta_k$ for all $k \geq \bar{k}$. Therefore, the point $\widehat{x}$ is feasible for all problems $P(B, \delta_k)$ with $k > \bar{k}$, and hence $v_k^* \leq f(\widehat{x}) \leq \bar{v}$ for all $k \geq \bar{k}$. This contradicts our aforementioned assumption which completes the proof. □

In the remainder of this article it will often be convenient to draw conclusions based on convergence of certain sequences. To this end, we may artificially set the termination tolerance $\varepsilon_v = 0$. In this case, it may indeed happen that Algorithm 1 does not terminate. In the following, this is referred to as the *possibly infinite branch-and-bound procedure corresponding to* $\varepsilon_v = 0$. Using this we can prove the following lemma, which can be used to show that after a finite number of iterations in Algorithm 1 a feasible point of the original problem $P(B)$ is guaranteed to be found.

Lemma 2 *Let $\delta > 0$ such that there exists some point $x \in M(B)$ with $g_i(x) \leq -\delta$ for all $i \in I$. Moreover, let Assumption 1 hold and assume that the possibly infinite branch-and-bound procedure corresponding to $\varepsilon_v = 0$ in Algorithm 1 does not terminate. Then, after a finite number of iterations a feasible point of the original problem $P(B)$ is found by Algorithm 1.*

Proof We consider an exhaustive subsequence of boxes $(X_{k_\nu})_{\nu\in\mathbb{N}}$ chosen in Step 1 with

$$\max_{i\in I} \ell_{g_i}(X_{k_\nu}) \le -\delta \qquad \text{for all} \qquad \nu \in \mathbb{N}.$$

First, we point out that given the box selection rule in Step 1 such an exhaustive sequence of boxes exists, provided the possibly infinite branch-and-bound procedure corresponding to $\varepsilon_v = 0$ does not terminate. This follows immediately using standard arguments from global optimization. In order to see this, it is important to note that there are at most a finite number of such boxes in the list $\mathcal{L}$ in every iteration and at least in every second iteration one must choose such an element. For a more formal explanation we refer to the proof of Theorem IV.1. as well as to Corollary IV.1. in [23].

We now consider a point $\widetilde{x}$ that is contained in all boxes X_{k_ν} for all $\nu \in \mathbb{N}$. From the definition of M-independent lower bounding procedures as well as continuity of the defining functions it follows $\lim_{\nu\to\infty} \ell_{g_i}(X_{k_\nu}) = g_i(\widetilde{x})$ for all $i \in I$, and hence we have $\max_{i\in I} g_i(\widetilde{x}) \le -\delta$. Again, using continuity of the functions g_i, for ν sufficiently large we conclude that all points in X_{k_ν} are feasible for $P(B)$. Therefore, every point x_{k_ν} that is chosen from this box is feasible as well, which concludes the proof. □

Revisiting our discussion on feasibility verification from Sect. 3, we observe that the proof of Lemma 2 allows to check feasibility for sets $M(X_k^j, \bar{\delta})$ with $0 < \bar{\delta} < \delta$ in Step 4 of Algorithm 1 without compromising the result.

Importantly, it can be shown that the assumptions of Lemma 2 are always satisfied after a finite number of steps.

Lemma 3 *Assume that Assumption 1 and Assumption 2 are satisfied, and that the possibly infinite branch-and-bound procedure corresponding to $\varepsilon_v = 0$ in Algorithm 1 does not terminate. Let $(\delta_k)_{k\in\mathbb{N}}$ denote the sequence of δ obtained in Algorithm 1. Then, there exists some $\bar{k} \in \mathbb{N}$ such that the first condition from Lemma 2 is satisfied for all $k \ge \bar{k}$.*

Proof First, we note that due to the existence of a Slater point $\tilde{x} \in M(B)$ satisfying $g_i(\tilde{x}) < 0$ for all $i \in I$ (Assumption 2), for some sufficiently small $\bar{\delta} > 0$, the first condition from Lemma 2 is guaranteed to be satisfied. By construction, it is also satisfied for all $\delta < \bar{\delta}$ then. We now prove that such a $\bar{\delta}$ is reached in Algorithm 1 after a finite number of steps.

We assume that this does not hold and derive a contradiction. Hence, we assume that for all $k \in \mathbb{N}$, we have $g_i(x) > -\delta_k$ for all $x \in M(B)$ and at least one $i \in I$.

We now consider an exhaustive subsequence of boxes $(X_{k_\nu})_{\nu\in\mathbb{N}}$ obtained in Algorithm 1, with $\bar{x} \in X_{k_\nu}$ for all $\nu \in \mathbb{N}$. We have $\bar{x} \in M(B)$, as by Lemma 4.1 in [30], otherwise the boxes $(X_{k_\nu})_{\nu\in\mathbb{N}}$ would no longer be selected and refined in Algorithm 1 after a finite number of steps.

As the M-independent lower bounding procedure ℓ_{g_i}, $i \in I$, is convergent according to Assumption 1, for at least one $i \in I$ we obtain $\lim_{\nu\to\infty} \ell_{g_i}(X_{k_\nu}) = g_i(\bar{x}) > -\delta_{k_\nu}$. From continuity it follows that there exists some $\bar{\nu} \in \mathbb{N}$ such that also $\max_{i\in I} \ell_{g_i}(X_{k_\nu}) > -\delta_{k_\nu}$ for all $\nu \ge \bar{\nu}$.

However, this means that for all $\nu \geq \bar{\nu}$ the list $\bar{\mathcal{L}}$ is empty and the parameter δ_k is decreased in Step 1 of Algorithm 1. The same argument can be repeated, so we obtain $\lim_{\nu\to\infty} \delta_{k_\nu} = 0$, and thus also $\lim_{k\to\infty} \delta_k = 0$. This is a contradiction to $\bar{\delta}$ not being reached after a finite number of steps. □

Remark 1 Lemma 3 in combination with Lemma 2 implies that after finitely many steps in Algorithm 1 a feasible point is identified, and thus the global upper bound u_k is updated. In particular, there exists a *non-empty* subsequence $(u_{k_\nu})_{\nu\in\mathbb{N}}$ of upper bounds related to iterations k_ν, $\nu \in \mathbb{N}$, where the upper bound is updated.

Remark 2 From Lemma 3 it also follows that there exists some $\tilde{k} \in \mathbb{N}$ such that for all $k \geq \tilde{k}$ we have $M(B, \delta_{\tilde{k}}) \neq \emptyset$. Therefore, by restricting to a subsequence $(\delta_{k_\nu})_{\nu\in\mathbb{N}}$ with $k_\nu \geq \tilde{k}$ for all $\nu \in \mathbb{N}$, all conditions of Lemma 1 are satisfied.

With these results, we are now able to prove convergence of the sequence of upper bounds generated by Algorithm 1.

Theorem 4 *Consider problem $P(B)$ with non-empty feasible set $M(B)$. Let Assumptions 1 and 2 hold. Then, if the possibly infinite branch-and-bound procedure corresponding to $\varepsilon_v = 0$ in Algorithm 1 does not terminate, the sequence of upper bounds $(u_k)_{k\in\mathbb{N}}$ converges to the globally optimal value v^*.*

Proof The sequence $(u_k)_{k\in\mathbb{N}}$ of upper bounds generated by Alorithm 1 is monotonically decreasing and bounded from below by the optimal value v^* of the original problem $P(B)$. Thus, the sequence $(u_k)_{k\in\mathbb{N}}$ is convergent.

However, it remains to be shown that this sequence really converges to the globally optimal value v^* of $P(B)$. To this end, we consider the non-empty subsequence $(u_{k_\nu})_{\nu\in\mathbb{N}}$ where there is an update of the upper bound in iteration k_ν. According to Lemma 3 and Remark 2 such a sequence exists. The values u_{k_ν} are computed by evaluating the objective function at feasible points $(x_{k_\nu})_{\nu\in\mathbb{N}}$. Since all these points are feasible and since the feasible set is compact, there must be a cluster point $\overline{x}$ of this sequence of points with $\overline{x} \in M(B)$.

Next, we show that $\overline{x}$ is, in fact, a globally optimal point of problem $P(B)$. To achieve this, we assume that this is not the case and derive a contradiction. Hence, if $\overline{x}$ is feasible and *not* optimal, then we have $f(\overline{x}) > v^*$. Moreover, in view of the box selection rule of Algorithm 1 there must be an exhaustive sequence of boxes $(X_\mu)_{\mu\in\mathbb{N}}$ created by the algorithm such that $\overline{x} \in X_\mu$ for all $\mu \in \mathbb{N}$. By Assumption 1 our lower bounding procedures are convergent, so we have

$$\lim_{\mu\to\infty} \ell(X_\mu) \quad = \quad f(\overline{x}) \quad > \quad v^*.$$

In particular, there is some $\overline{\mu}$ such that

$$\ell(X_\mu) > \frac{1}{2}\Big(f(\overline{x}) + v^*\Big) \qquad \text{for all} \qquad \mu \geq \overline{\mu}. \tag{1}$$

We now continue to show that this is actually not possible. To this end, we consider an arbitrary iteration k together with the corresponding δ_k. After a finite number of iterations we either have that the list $\overline{\mathcal{L}}$ is empty or otherwise, as shown in Lemma 3, that a feasible point is found. In both cases, in these iterations k the value δ_k is decreased by letting $\delta_{k+1} = \gamma\delta_k$ (either in Step 1 or in Step 4). Inductively, for that reason, we have $\lim_{k\to\infty}\delta_k = 0$.

Additionally, by Lemma 3 and Remark 1, for sufficiently large k, all conditions of Lemma 1 are satisfied. Hence, we conclude that the sequence of optimal values v_k^* of the problems $P(B, \delta_k)$ converges to v^*. In particular, there is some $\overline{k} \in \mathbb{N}$ such that

$$v_k^* \le \frac{1}{2}\Big(f(\overline{x}) + v^*\Big) \qquad \text{for all} \qquad k \ge \overline{k}.$$

Then, however, for $k \ge \overline{k}$, neither in iterations with an even iteration number k nor in iterations with an odd iteration number k we select a box in Step 1 of Algorithm 1 with a lower bound larger than $v_k^* \le \frac{1}{2}(f(\overline{x}) + v^*)$, because there are always boxes with a smaller lower bound contained in the list $\mathcal{L}$ and $\overline{\mathcal{L}}$, respectively. In particular, that means that inequality (1) is violated for μ sufficiently large, such that the sequence $(X_\mu)_{\mu\in\mathbb{N}}$ is not created by the algorithm at all. This contradicts our assumption and, hence, the point $\overline{x}$ must be a globally optimal point of the problem $P(B)$.

Since our upper bounds are created by evaluating the objective function at the points $(x_{k_\nu})_{\nu\in\mathbb{N}}$ and since this sequence possesses the feasible point $\overline{x}$ as a cluster point, the assertion now follows from continuity of the objective function f. □

4.2 Convergence of lower bounds and optimal points

In this section we establish convergence of the sequence of the overall lower bounds $\widehat{v}_k$ of Algorithm 1. Together with the convergent upper bounding procedure this ensures finite termination of the branch-and-bound algorithm. Similarly, one can establish a convergence result for optimal points.

Proposition 5 *Let $M(B) \neq \emptyset$ and let Assumption 1 hold. Then, if the possibly infinite branch-and-bound procedure of Algorithm 1 corresponding to $\varepsilon_v = 0$ does not terminate, for the overall lower bounds $\widehat{v}_k$ in every iteration k the limit satisfies $\lim_{k\to\infty}\widehat{v}_k = v^*$ where v^* denotes the optimal value of $P(B)$.*

Proof We apply Theorem IV.3. from [23]. To this end, two main properties need to be ensured, i.e. the bounding operation needs to be *bound improving* and *consistent*. The former means that after a finite number of iterations a tesselation element needs to be selected where the lowest bound is attained. Clearly, in Algorithm 1 in every second iteration such a box is chosen.

The latter, i.e. consistency of the bounding operation, is a bit more involved. In view of the box division rule in Algorithm 1 for our setting this means that for every exhaustive sequence of boxes $(X_{k_\nu})_{\nu\in\mathbb{N}}$ created by the algorithm we have

$$\lim_{\nu\to\infty}\Big(u_{k_\nu} - \ell(X_{k_\nu})\Big) = 0.$$

Clearly, this expression cannot be negative, since boxes X where the lower bound $\ell(X)$ is strictly larger than the current upper bound u_k are fathomed in Step 6 of the algorithm. Moreover, for any threshold $\tau > 0$ the difference $u_{k_\nu} - \ell(X_{k_\nu})$ cannot remain larger than τ, as we show now. From Theorem 4 we know that for the subsequence of upper bounds $(u_{k_\nu})_{\nu\in\mathbb{N}}$ we have

$$\lim_{\nu\to\infty} u_{k_\nu} = v^*.$$

Furthermore, by monotonicity of the lower bounding procedure the sequence of lower bounds $(\ell(X_{k_\nu}))_{\nu\in\mathbb{N}}$ is monotonically increasing. As it is also bounded from above by v^*, it is convergent. Let us now assume for a moment that

$$\lim_{\nu\to\infty} \ell(X_{k_\nu}) = \overline{v} < v^*.$$

Then, there exists a feasible point $\overline{x} \in X_{k_\nu}$ for all ν with $f(\overline{x}) = \overline{v}$, see Lemma 4.1 in [30]. However, that means that v^* could not be the optimal value of the problem, which contradicts our assumption. Hence, we have

$$0 \leq \left(u_{k_\nu} - \ell(X_{k_\nu})\right) = \left(u_{k_\nu} - v^*\right) + \left(v^* - \ell(X_{k_\nu})\right)$$

and the assertion follows by sandwiching the expression in the middle. □

In classical branch-and-bound algorithms in global optimization one can easily consider convergent subsequences of points in boxes and usually these subsequences converge to globally optimal points of the original problem. Due to the fact that not always boxes with the most promising lower bounds are selected in Step 1, in Algorithm 1 one could think that this might not necessarily be the case. However, in the next result we show that this result still holds under this modification.

Proposition 6 *Let $M(B) \neq \emptyset$, let Assumption 1 and Assumption 2 hold and let us assume that the possibly infinite branch-and-bound procedure of Algorithm 1 corresponding to $\varepsilon_v = 0$ does not terminate. Moreover, we consider a subsequence of boxes $(X_k)_{k\in\mathbb{N}}$ chosen in Step 1, and let $x_k \in X_k$ for all $k \in \mathbb{N}$. Then, $(x_k)_{k\in\mathbb{N}}$ possesses a cluster point and any such cluster point is a globally optimal point of $P(B)$.*

Proof Due to $x_k \in B$ for all $k \in \mathbb{N}$ and due to the fact that B is bounded, the sequence $(x_k)_{k\in\mathbb{N}}$ possesses a cluster point. We now consider a convergent subsequence $(x_{k_\nu})_{\nu\in\mathbb{N}}$ of $(x_k)_{k\in\mathbb{N}}$ and put $\overline{x} := \lim_{\nu\to\infty} x_{k_\nu}$.

Next, we assume that $\overline{x}$ is not a globally optimal point and derive a contradiction by distinguishing two different cases.

Case 1: $\overline{x} \in M(B)$ and $f(\overline{x}) > v^*$. Then, there exists an exhaustive sequence of boxes $(X_{k_\mu})_{\mu\in\mathbb{N}}$ generated by the algorithm with $\overline{x} \in X_{k_\mu}$ for all $\mu \in \mathbb{N}$. According to Assumption 1 in Step 3 an M-dependent lower bounding procedure is used, so we have $\lim_{\mu\to\infty} \ell(X_{k_\mu}) = f(\overline{x})$. Using arguments from the proof of Theorem 4 we can show that after a finite number of iterations no box with $\ell(X_{k_\mu}) > \frac{1}{2}(f(\overline{x}) + v^*)$ is selected and, hence, the exhaustive sequence of boxes $(X_{k_\mu})_{\mu\in\mathbb{N}}$ is not created at all, contradicting the assumption.

Case 2: $\overline{x} \notin M(B)$. Then, by Assumption 1 we have $\lim_{\mu\to\infty} \ell(X_{k_\mu}) = +\infty$. Again, this can be ruled out by using the same line of arguments as in the proof of Theorem 4. □

According to Theorem 4, the overall upper bounds generated by Algorithm 1 converge to the globally optimal value if the assumptions of Theorem 4 hold. Moreover, in view of Proposition 5 lower bounds are ensured to converge to the globally optimal value as well. Therefore, the branch-and-bound method terminates after a finite number of iterations for $\varepsilon_v > 0$. This immediate consequence is stated formally in the following result.

Corollary 7 *Consider problem $P(B)$ with non-empty feasible set $M(B)$ and let a positive termination tolerance $\varepsilon_v > 0$ be given. Further assume that Assumption 1 and Assumption 2 hold. Then, if a convergent lower bounding procedure is used, Algorithm 1 terminates after a finite number of iterations.*

Proof Since the lower bounding procedure is assumed to be convergent we have from Proposition 5 the limit $\lim_{k\to\infty} \widehat{v}_k = v^*$. Moreover, according to Theorem 4 we have $\lim_{k\to\infty} u_k = v^*$. Thus, we obtain

$$\lim_{k\to\infty} (u_k - \widehat{v}_k) = \lim_{k\to\infty} \Big((u_k - v^*) + (v^* - \widehat{v}_k) \Big) = 0.$$

Therefore, after a finite number of iterations we have $u_k - \widehat{v}_k \leq \varepsilon_v$, and the assertion must hold. □

Let us remark that we have not addressed the case of an empty feasible set yet. Typically, spatial branch-and-bound algorithms in global optimization are able to recognize this by stopping with a certificate of infeasibility. In this case, no box can be selected in Step 1 anymore, since the list $\mathcal{L}$ becomes empty. In principle, this also holds for Algorithm 1. In order to show this formally, it is required that the lower bounding procedure is not only convergent, but also recognizes empty boxes X by evaluating to the extended real value $\ell(X) = +\infty$, provided that the box X is sufficiently small. This is also fulfilled for common lower bounding procedures (see, e.g., [3, 4, 7]).

5 Discussion of the assumptions

In the following we briefly discuss our main requirements that need to be fulfilled so that the concept of restriction of the right-hand side in Algorithm 1 is sufficient to ensure termination of spatial branch-and-bound algorithms. We start with Assumption 1 that is straightforward in this regard, as it comprises a common requirement regarding the lower bounding procedure in global optimization. In fact, lower bounding procedures that fulfill this are widely used and typically required for basically all algorithms in that domain (see, e.g., [23, 30]).

In contrast, Assumption 2 is rarely used in global optimization, although very common in local optimization. Moreover, under the additional assumption that all defining functions are differentiable, it can even be shown to be a straightforward consequence of other (weak) constraint qualifications that are also common in nonlinear local optimization. This holds, for instance, for the *Mangasarian-Fromovitz constraint qualification (MFCQ)*. In order to formulate this for problem $P(B)$ we explicitly rewrite the box constraints as additional constraints, see [30]:

$$\begin{aligned} h_i(x) &= \underline{b}_i - x_i \leq 0 \qquad i = 1, \ldots, n \\ h_j(x) &= x_i - \overline{b}_i \leq 0 \qquad j = n+1, \ldots, 2n. \end{aligned}$$

Then, MFCQ is said to hold at a feasible point $\overline{x} \in M(B)$, if there is a direction $d \in \mathbb{R}^n$ such that

$$\begin{aligned} Dg_i(\overline{x})d &< 0 \qquad \text{for all active indices, i.e. indices } i \text{ with } g_i(\overline{x}) = 0 \\ Dh_j(\overline{x})d &< 0 \qquad \text{for all active indices, i.e. indices } j \text{ with } h_j(\overline{x}) = 0. \end{aligned}$$

Hence, if MFCQ holds at every globally optimal point x^*, then standard arguments show that for $\lambda > 0$ sufficiently close to zero we have

$$\begin{aligned} g_i(x^* + \lambda d) &< 0 \qquad \text{for all } i \in I \\ \text{and} \quad h_j(x^* + \lambda d) &< 0 \qquad \text{for all } j = 1, \ldots, 2n, \end{aligned}$$

and thus a Slater point exists in every neighbourhood $N(x^*)$ around a point x^*. In particular, that means that Assumption 2 holds.

Hence, Assumption 2 is mild in the sense that it is a direct consequence of MFCQ, which, in turn, is implied by the *Linear Independence Constraint Qualification (LICQ)*. This, however, is already mild in the sense that it is proven to generically hold everywhere in the feasible set $M(B)$ (see [27]). The latter means, in particular, that in case of its violation it may be expected to hold at least under small perturbations of the problem data. We briefly state the aforementioned consideration in the next remark.

Remark 3 Let MFCQ or LICQ hold in every globally optimal point of problem $P(B)$. Then, Assumption 2 is fulfilled.

Furthermore, we point out that in nonlinear local optimization the assumption of LICQ even in all locally minimal points is standard for convergence proofs. This means, in particular, that common upper bounding procedures in spatial branch-and-bound methods which rely on the local solution of NLP subproblems implicitly use this or related assumptions as well and, hence, Assumption 2 is not restrictive in this regard.

Finally, let us stress that even in case that Assumption 2 is violated, our upper bounding procedure may still provide valid upper bounds, although it may happen that these upper bounds do not converge to the globally optimal value v^*. As becomes clear from the proof of Lemma 1 it is not possible, though, that a value is computed that is strictly smaller than v^*, in contrast to many other approaches such as acceptance of ε_f-feasible points.

Table 1 Test problems and dimensions

name	n	p
booth	2	1
hs011	2	1
simpllpa	2	2
zecevic4	2	2
hs030	3	1
congigmz	3	5
hs044	4	6
hs268	5	5
ex3_1_2	5	6
hs098	6	4
hs113	10	8

6 Illustrative computational results

In this section, we provide illustrative computational results for 11 small test problems from the COCONUT benchmark library [50], which are summarized in Table 1 with their name, variable dimension and number of inequality constraints.

We should emphasize that we do not present results for extensive computational tests and large problem instances, and this is for the following reason: The merit of restriction of the right-hand side is that it is a very simple, but powerful concept that allows to ensure deterministic and finite convergence of spatial branch-and-bound methods with marginal additional computational effort. For most of the implementations of such algorithms, this is not guaranteed. However, solvers like `ANTIGONE`, `BARON`, `LINDOGlobal`, `Octeract` or `SCIP` are highly-tuned and perform extremely well for many test problems. Therefore, we do not expect a simple implementation using restriction of the right-hand side to be computationally competitive with these solvers. We rather advocate to incorporate our proposed approach into existing methods as an *additional* way to obtain upper bounds and as a convergence guarantee. The results in this section are included for illustrative reasons and as a proof-of-concept.

6.1 Implementation and test details

Our implementation is based on Algorithm 1 with only slight modifications. The algorithm is implemented in *Python* 3.7. For numeric operations *Numpy* and *Scipy* are used, and for interval arithmetic operations we apply the *IntvalPy* package [6]. The lower bounds are computed using centered forms [31]. Moreover, as indicated in Sect. 3, for each k, the value $\delta(X_k) = \max_{i \in I} \ell_{g_i}(X_k)$ is computed once when the box X_k is constructed. We do not use bounds tightening or similar acceleration techniques.

In Step 4 of Algorithm 1, we use three different strategies to choose candidate points x_k^j:

(a) `Mid`: Here, we choose $x_k^j := \text{mid}(X_k^j)$, and then check for feasibility.

(b) `Mid + Loc`: Here, in addition to (a), if $\mathrm{mid}(X_k^j) \notin M(X_k^j)$, we run `IPOPT` [58] to obtain a local solution of $P(X_k^j)$ as an alternative candidate for x_k^j. This candidate is then checked for feasibility.

(c) `Mid + Loc-Res`: Same approach as (b), but for k being even, we solve the restricted problem $P(X_k^j, \delta_k)$ locally; see also our discussion at the end of Sect. 3.

In each case, we do not allow for ε_f-feasibility when checking whether $x_k^j \in M(X_k^j)$.

To access `IPOPT` as a local solver, we use the default function *minimize_ipopt* from the Python package *cyipopt* [5]. We use the default feasibility tolerance of 10^{-4}, but do not provide information on gradients or Hessians manually.

For restriction of the right-hand side, we choose $\delta_0 = 1, \gamma = 0.95$ and perform a restricted selection step in every κ-th iteration with $\kappa \in \{2, 10, 100\}$. We compare this to a branch-and-bound method with only normal selection steps (no restriction). Note that for case `Mid + Loc-Res` we only consider $\kappa \in \{2, 10\}$, since solving the restricted problems locally only seems worthwhile if restricted problems occur sufficiently often.

For the termination criterion we set $\varepsilon_v = 10^{-3}$ and additionally introduce a time limit of 7,200 s and an iteration limit of 10,000 iterations. The experiments are executed on a Windows machine with 2.5GHz Intel Core i5-6300U CPU and 12GB of RAM.

6.2 Discussion of results

The results are summarized in Tables 2. The columns compare the results for different values of κ (with "-" indicating that no restriction is applied) and with different selection strategies for x_k^j. In each column, the total number of iterations until termination (it_{term}), the first iteration in which a feasible point is found and an upper bound is computed (it_{feas}), the total time in seconds (time) and the solution status (status) are reported. If the problem is not solved to optimality, we report the remaining relative optimality gap $\frac{UB-LB}{|UB|}$ as the status.

The results show that the computational performance of using restriction of the right-hand side and of using only normal selection steps do not differ by much (for all strategies, `Mid`, `Mid + Loc` and `Mid + Loc-Res`). On first sight, this seems detrimental, as we do not see a clear performance advantage for our proposed method. However, the results highlight that restriction of the right-hand side provides a convergence guarantee at low computational overhead, i.e. without compromising the performance of the spatial branch-and-bound method by much. For several problems, the number of iterations and total time required until termination are a bit lower if only normal selection steps are applied, but this is not always the case.

In fact, we see that in one case, for problem *congigmz*, restriction of the right-hand side with a sufficiently small κ and $x_k^j = \mathrm{mid}(X_k^j)$ leads to an early identification of a feasible point, and thus computation of a valid upper bound for v^*. In contrast, without restriction of the right-hand side no finite upper bound can be computed within the given time and iteration limit.

Table 2 Computational results for Algorithm 1 (part 1)

	Mid				Mid + Loc				Mid + Loc-Res	
κ	2	10	100	-	2	10	100	-	2	10
booth										
it_t	527	527	2	527	2	2	2	2	2	2
it_f	526	526	1	526	1	1	1	1	1	1
time	18	21	4	17	3	2	2	2	2	2
status*	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**
hs011										
it_t	1683	1673	1672	1672	1672	1672	1672	1672	1671	1672
it_f	1	1	1	1	1	1	1	1	1	1
time	57	56	52	56	673	613	627	599	1004	710
status*	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**
simpllpa										
it_t	106	92	89	89	106	92	89	89	100	92
it_f	1	1	1	1	1	1	1	1	1	1
time	3	3	6	4	20	19	18	18	27	20
status*	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**
zecevic4										
it_t	609	579	572	570	603	579	572	572	570	579
it_f	4	4	4	4	4	4	4	4	2	4
time	27	29	29	47	532	506	542	509	521	667
status*	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**
hs030										
it_t	max	max	max	max	400	400	400	400	414	400
it_f	-	-	-	-	1	1	1	1	1	1
time	334	292	386	345	354	349	356	338	494	413
status*	-	-	-	-	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**
congigmz										
it_t	max	max	max	max	852	839	728	920	594	511
it_f	964	1580	-	-	-	-	-	-	-	-
time	1042	1017	1345	1049	max	max	max	max	max	max
status*	0.02	0.21	-	-	-	-	-	-	-	-
hs044										
it_t	398	283	268	268	398	283	269	269	384	283
it_f	47	47	47	47	47	47	47	47	47	47
time	46	43	36	40	2194	1870	1614	1577	2195	1746
status*	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**	**opt**

Table 2 continued

κ	Mid 2	10	100	-	Mid + Loc 2	10	100	-	Mid + Loc-Res 2	10
hs268										
it_t	max	max	max	max	902	839	738	912	862	780
it_f	-	-	-	-	-	-	-	-	-	-
time	1536	1926	1914	1466	max	max	max	max	max	max
status*	-	-	-	-	-	-	-	-	-	-
ex3_1_2										
it_t	max	max	max	max	613	710	652	714	732	221
it_f	2	2	2	2	2	2	2	2	4	2
time	2793	2871	2619	2384	max	max	max	max	max	max
status*	0.01	0.01	0.01	0.01	5.65	1.97	2.01	1.90	5.65	6.59
hs098										
it_t	max	max	max	max	525	529	493	529	522	350
it_f	-	-	-	-	299	223	210	210	4	20
time	2102	2418	2323	2518	max	max	max	max	max	max
status*	-	-	-	-	71.83	68.02	68.05	67.15	71.88	72.70
hs113										
it_t	max	max	6725	max	79	73	62	82	69	74
it_f	-	-	-	-	-	-	-	-	-	-
time	3456	4360	max	6995	max	max	max	max	max	max
status*	-	-	-	-	-	-	-	-	-	-

* **opt**: an ε-optimal solution has been determined;
otherwise the remaining relative gap $\frac{UB-LB}{|UB|}$ is reported ("-" if no UB has been found)

Whereas the additional usage of basic IPOPT does not lead to significantly more instances being solved (only *hs030*), and also slows down the solution process, the results illustrate that local solvers can be easily combined with restriction of the right-hand side and leveraged to find feasible points more quickly (see *booth, hs030, hs098*).

For 2 out of 12 test problems, even with restriction of the right-hand side, no valid upper bound is determined within the given time and iteration limit. Additionally, for 5 problems, finite upper bounds are determined, but no convergence is achieved in the predefined iteration and time limit, even if for *congigmz* and *ex3_1_2* at least very small *relative* optimality gaps are reached.

This lack of performance can be assumed to be explained by the very simple branch-and-bound implementation. In practical applications, a branch-and-bound method using restriction of the right-hand side could be tuned by introducing improved lower bounding procedures, bounds tightening, additional strategies to identify feasible points and several more acceleration techniques. Also IPOPT could be tuned, for instance by providing derivative information.

Finally, let us note that even in cases where we only examine box midpoints for feasibility, for 6 out of 12 test problems, the branch-and-bound method terminates successfully in the predefined iteration limit. This is even true without using restricted selection steps. Importantly, in contrast to the proven results for our proposed approach, using only normal selection steps this is not guaranteed in general, though. We provide an illustrative example in Appendix A for which only boxes with infeasible midpoints are selected as long as the selection step is not modified.

7 Conclusions

In this paper we proposed a new convergent upper bounding procedure that is straightforward to incorporate into spatial branch-and-bound algorithms in global optimization with little overhead. A proof of convergence is provided so that termination of classical branch-and-bound algorithms is ensured. Moreover, some illustrative numerical examples are presented. However, there are still a few issues we would like to mention.

First, we point out that our aim is not to present an entire new branch-and-bound algorithm. Instead, we focus on the upper bounding procedure itself that can be applied in a wide number of real solvers. The assumptions are mild, as discussed in detail. For this reason, our own implementation is meant as a proof-of-concept rather than a solver that is meant for production. For this reason, we did not incorporate important acceleration steps such as bound tightening. Moreover, the lower bounding procedure is convergent, but weaker than many of the standard approaches that are commonly used. In addition, none of the steps is performed in parallel, which is standard in real solvers in this domain. This is also reflected in our numerical results where we only solved some small problems. However, from the discussion of the assumptions it becomes clear that the method works under mild assumptions, and thus can be integrated into a wide range of solvers in that area.

Furthermore, in this paper we assumed boxes as partition elements, although we believe that the approach can be extended to other partition elements as well. We also assumed that a box is always divided along the midpoint of the longest edge. Although this is common, it should be possible to use other strategies here as well, as long as they guarantee exhaustive sequences. The same holds for acceleration strategies such as bound tightening, which are common in state-of-the-art implementations that are used in practice.

Moreover, as mentioned before, in our theoretical results we assume that a restricted selection step based on restriction of the right-hand side is performed in every second iteration for simplicity. For our proofs of convergence it is only important that such step is always made after a finite number of iterations. Hence, we have some flexibility regarding the choice of iterations with restricted selection steps, which also allows the combination with different search patterns.

Finally, in this paper we discussed a very specific strategy to drive the feasibility parameter to zero by letting $\delta_{k+1} = \gamma\delta_k$ for some $\gamma \in (0, 1)$. However, we point out that other possibilities can be used here as well.

Appendix A Illustrative example

The following example illustrates that sufficiency of considering box midpoints to find feasible points is a special feature of our proposed approach, while not guaranteed in general.

Example 2 Consider the nonlinear problem

$$\begin{aligned} P(B): \min_{x\in\mathbb{R}^2} \quad & f(x) := x_1 \\ \text{s.t.} \quad & g(x) := -x_1^2 - (x_2-5)^2 + \sqrt{2} + 25 \leq 0, \\ & x \in B := [1,2] \times [0,1], \end{aligned}$$

which is borrowed and slightly adapted from Example 3.1 in [30]. The globally minimal point is $x^* = (\sqrt[4]{2}, 0)^\top$ with globally minimal value $f(x^*) = \sqrt[4]{2}$. The feasible set is depicted in Fig. 3.

We show that for any box X that is chosen within a simple spatial branch-and-bound method with only *normal selection* and division along the longest edge (cf. Algorithm 1),

the midpoint $\widehat{x} := \mathrm{mid}(X)$ is infeasible. For the initial box B and any sub-box $X = [\underline{x}, \overline{x}]$ with $\widehat{x}_1 < \sqrt[4]{2}$, this can be easily see from Fig. 3. The initial box B has rational bounds, so that all other boxes obtained by division of the longest edge have rational bounds and midpoints as well. Hence, the case $\widehat{x}_1 = \sqrt[4]{2}$ can be excluded. The assertion remains to be proven for boxes X with $\widehat{x}_1 > \sqrt[4]{2}$. We proceed with a case distinction.

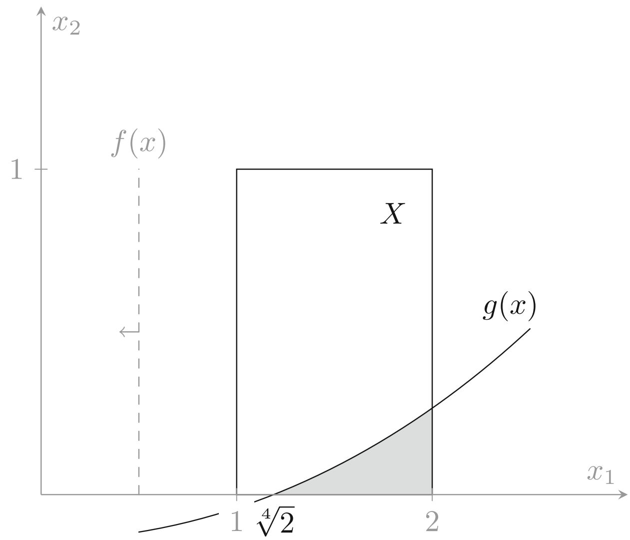

Fig. 3 Box X and feasible set for Example 2

Case 1: $\widehat{x}_1 > \underline{x}_1 > \sqrt[4]{2}$. Let $\tilde{X} = [\underline{\alpha}, \overline{\alpha}]$ be a box in $\mathcal{L}$ with $\underline{\alpha}_1 < \sqrt[4]{2} < \overline{\alpha}_1, \overline{\alpha}_1 \leq \underline{x}_1$ and $x^* \in \tilde{X}$, i.e., a box located left of X and containing x^*. Since x^* is the globally optimal point, such a box always has to exist.

Assume that in Step 3 of the branch-and-bound method we use some lower bounding procedure, which uses a convex relaxation $\widehat{M}(X)$ of $M(X)$. The objective function f is linear and therefore not relaxed. We obtain

$$\ell(\tilde{X}) \leq \min_{x \in M(\tilde{X})} f(x) = f(x^*) = \sqrt[4]{2},$$

but also

$$\ell(X) = \min_{x \in \widehat{M}(X)} \geq \min_{x \in X} f(x) > \sqrt[4]{2}$$

by definition of X. However, this yields a contradiction, since box X would have never been selected by the selection rule.

Case 2: $\widehat{x}_1 > \sqrt[4]{2} > \underline{x}_1$. In this case, there exist two possible sub-cases. In the first one, the selected box X is a rectangle with $\overline{x}_2 - \underline{x}_2 = 2(\overline{x}_1 - \underline{x}_1)$. By division by the longest edge, two quadratic boxes X', X'' are constructed with midpoints $\text{mid}(X') = (\widehat{x}_1, \frac{3}{4}\underline{x}_2 + \frac{1}{4}\overline{x}_2)$ and $\text{mid}(X'') = (\widehat{x}_1, \frac{1}{4}\underline{x}_2 + \frac{3}{4}\overline{x}_2)$. If $\text{mid}(X')$ is infeasible, then $\text{mid}(X'')$ is clearly infeasible as well, so we only examine the first one.

Looking at Fig. 3, we see that the level curve of g to level 0 can be interpreted as a convex function in x_1, and thus can be overestimated by a linear function between $(\sqrt[4]{2}, 0)$ and $(2, 5 - \sqrt{21 + \sqrt{2}})$. Let

$$m = \frac{5 - \sqrt{21 + \sqrt{2}}}{2 - \sqrt[4]{2}} \approx 0.3276,$$

then this linear function is given by

$$\phi(x_1) := m x_1 - \sqrt[4]{2} m. \tag{A1}$$

On the other hand, the midpoint $\text{mid}(X')$ is located on a line with slope 1 and starting at $\underline{x}$. More formally, this line is described by the linear function

$$\gamma(x_1) := x_1 + \underline{x}_2 - \underline{x}_1. \tag{A2}$$

Note that $\phi(x_1)$ by construction has a zero at $\sqrt[4]{2}$, while

$$\gamma(\sqrt[4]{2}) = \sqrt[4]{2} + \underline{x}_2 - \underline{x}_1 > \underline{x}_2 \geq 0.$$

Hence, $\gamma(\sqrt[4]{2}) > \phi(\sqrt[4]{2})$. Since γ also has a larger slope than ϕ, it overestimates the latter on $(\sqrt[4]{2}, 2]$. This means that $\text{mid}(X')$ lies above ϕ, by that above the level curve of g to level 0, and thus is infeasible.

In the second sub-case, the selected box X is quadratic. We can prove the assertion using a similar reasoning as before.

The main difference in applying restriction of the right-hand side is that Case 1 cannot be excluded by the adapted selection rule. Therefore, it is guaranteed that after finitely many steps for some selected box X, the midpoint $\mathrm{mid}(X)$ will be feasible. In fact, if we apply Algorithm 1 with $\delta_0 = 1$ and $\gamma = 0.95$, in iteration 16 a box with feasible midpoint is selected, and thus a valid upper bound for v^* is computed.

Acknowledgements The authors are grateful to Hendrik Otto on whose source code they based their implementation and the computational illustrations. Moreover, the authors thank the anonymous referee for his or her precise remarks, which helped to significantly improve the paper.

Funding No funding was received for conducting this study.

Declarations

Conflict of interest The authors have no Conflict of interest to declare.

References

1. Achterberg, T.: SCIP: solving constraint integer programs. Math. Program. Comput. **1**, 1–41 (2009)
2. Adjiman, C.S., Floudas, C.A.: Rigorous convex underestimators for general twice-differentiable problems. J. Global. Optimiz. **9**, 23–40 (1996)
3. Adjiman, C.S., Androulakis, I.P., Floudas, C.A.: A global optimization method, αbb, for general twice-differentiable NLPs: II. Implementation and computational results. Comput. Chem. Eng. **22**, 1159–1179 (1998)
4. Adjiman, C.S., Dallwig, S., Floudas, C.A., Neumaier, A.: A global optimization method, αbb, for general twice-differentiable NLPs: I. Theoret. Adv. Comput. Chem. Eng. **22**, 1137–1158 (1998)
5. Aides, A., Kümmerer, M.: Cyipopt. Code released on url-https://cyipopt.readthedocs.io/en/stable/index.html, (2024)
6. Androsov, A.: IntvalPy. Code released on GitHub https://github.com/AndrosovAS/intvalpy/, (2021)
7. Androulakis, I.P., Maranas, C.D., Floudas, C.A.: αbb: a global optimization method for general constrained nonconvex problems. J. Global. Optimiz. **7**, 337–363 (1995)
8. Baumann, E.: Optimal centered forms. BIT Numeric. Math. **28**, 80–87 (1988)
9. Belotti, P., Lee, J., Liberti, L., Margot, F., Wächter, A.: Branching and bounds tightening techniques for non-convex minlp. Optimizat. Methods Software. **24**, 597–634 (2009)
10. Berthold, T., Gleixner, A.M.: Undercover: a primal MINLP heuristic exploring a largest sub-MIP. Mathematical Programming. Ser. A **144**, 315–346 (2014)
11. Bonami, P., Cornuéjols, G., Lodi, A., Margot, F.: A feasibility pump for mixed integer nonlinear programs. Mathematical Programming. Ser. A **119**, 331–352 (2009)
12. Dickinson, P.J.C.: On the exhaustivity of simplicial partitioning. J. Global. Optimizat. **58**, 189–203 (2014)
13. Domes, F., Neumaier, A.: Rigorous verification of feasibility. J. Global. Optimizat. **61**, 255–278 (2015)
14. Dür, M.: Dual bounding procedures lead to convergent branch-and-bound algorithms. Math. Program. **91**, 117–125 (2001)

15. Dür, M.: A class of problems where dual bounds beat underestimation bounds. J. Global. Optimizat. **22**, 49–57 (2002)
16. Falk, J.E., Soland, R.M.: An algorithm for separable nonconvex programming problems. Manag. Sci. **15**(9), 550–569 (1969)
17. Floudas, C. A.: Deterministic global optimization: theory, algorithms and applications, volume 37 of Nonconvex Optimization and its Applications. Kluwer Academic Publishers, 309–554 (2000)
18. Füllner, C., Kirst, P., Otto, H., Rebennack, S.: Feasibility verication and upper bound determination in global minimization using approximate active index sets. INFORMS Journal on Computing, (2024)
19. Füllner, C., Kirst, P., Stein, O.: Convergent upper bounds in global minimization with nonlinear equality constraints. Math. Program. **187**(1), 617–651 (2021)
20. Geißler, B., Martin, A., Morsi, A., Schewe, L.: Using piecewise linear functions for solving MINLPs. In J. Lee and S. Leyffer, editors, Mixed integer nonlinear programming, volume 154 of The IMA volumes in mathematics and its applications, pages 287–314. Springer, (2012)
21. Gounaris, C.E., Floudas, C.A.: Tight convex underestimators for $\mathcal{C}^2$-continuous problems: I. univariate functions. J. Global. Optimizat. **42**, 51–67 (2008)
22. He, T., Tawarmalani, M.: A new framework to relax composite functions in nonlinear programs. Math. Program. **190**, 427–466 (2021)
23. Horst, R., Tuy, H.: Global Optimization: Deterministic Approaches. Springer, Berlin (1996)
24. Hübner, T., Gupte, A., Rebennack, S.: Spatial branch-and-bound for nonconvex separable piecewise linear optimization. Preprint, available online at https://optimization-online.org/?p=26328
25. Jones, D.R., Martins, J.R.R.A.: The DIRECT algorithm: 25 years later. J. Global. Optimizat. **79**, 521–566 (2021)
26. Jones, D.R., Perttunen, C.D., Stuckman, B.E.: Lipschitzian optimization without the Lipschitz constant. J. Optimizat. Theory. Appl. **79**(1), 157–181 (1993)
27. Jongen, HTh., Jonker, P., Twilt, F.: Nonlinear optimization in finite dimensions. Kluwer, Dordrcht (2000)
28. Kearfott, R.B.: On proving existence of feasible points in equality constrained optimization problems. Math. Program. **83**, 89–100 (1998)
29. Kearfott, R.B.: On rigorous upper bounds to a global optimum. J. Global. Optimizat. **59**, 459–476 (2014)
30. Kirst, P., Stein, O., Steuermann, P.: Deterministic upper bounds for spatial branch-and-bound methods in global minimization with nonconvex constraints. TOP **23**, 591–616 (2015)
31. Krawczyk, R., Nickel, K.: Die zentrische Form in der Intervallarithmetik, ihre quadratische Konvergenz und Inklusionsisotonie. Computing **28**, 117–137 (1982)
32. Lasserre, J.B., Thanh, T.P.: Convex underestimators of polynomials. J. Global. Optimizat. **56**, 1–25 (2013)
33. Lin, Y., Schrage, L.: The global solver in the LINDO API. Optimizat. Method. Software. **24**, 657–668 (2009)
34. Locatelli, M., Schoen, F.: Global optimization: Theory, algorithms and applications. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, Philadelphia, PA (2013)
35. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: part I - convex underestimating problems. Math. Program. **10**(1), 147–175 (1976)
36. Meyer, C.A., Floudas, C.A.: Convex underestimation of twice continuously differentiable functions by piecewise quadratic perturbation: spline αBB underestimators. J. Global Optimizat. **32**, 221–258 (2005)
37. Miranda, C.: Un' osservazione su una teorema di Brouwer. Bollettino dell'Unione Matematica Italiana **3**, 527 (1940)
38. Misener, R., Floudas, C.A.: Piecewise-linear approximations of multidimensional functions. J. Optimizat. Theory Appl. **145**, 120–147 (2010)
39. Misener, R., Floudas, C.A.: Antigone: algorithms for continuous / integer global optimization of nonlinear equations. J. Global. Optimizat. **59**(2–3), 503–526 (2014)
40. Mitsos, A.: Global optimization of semi-infinite programs via restriction of the right-hand side. Optimization **60**(10–11), 1291–1308 (2011)
41. Mitsos, A., Tsoukalas, A.: Global optimization of generalized semi-infinite programs via restriction of the right hand side. J. Global. Optimizat. **61**(1), 1–17 (2015)
42. Neumaier, A.: Interval methods for systems of equations. Cambridge University Press, Cambridge (1990)

43. Pintér, J.: Branch and bound algorithms for solving global optimization problems with Lipschitzian structure. Optimization **19**, 101–110 (1988)
44. Puranik, Y., Sahinidis, N.V.: Domain reduction techniques for global NLP and MINLP optimization. Constraints **22**, 338–376 (2017)
45. Rebennack, S., Kallrath, J.: Continuous piecewise linear delta-approximations for bivariate and multivariate functions. J. Optimizat. Theory. Appl. **167**(1), 102–117 (2015)
46. Rebennack, S., Kallrath, J.: Continuous piecewise linear delta-approximations for univariate functions: computing minimal breakpoint systems. J. Optimizat. Theory. Appl. **167**(2), 617–643 (2015)
47. Ryoo, H.S., Sahinidis, N.V.: Global optimization of nonconvex NLPs and MINLPs with applications in process design. Comput. Chem. Eng. **19**, 551–566 (1995)
48. Ryoo, H.S., Sahinidis, N.V.: A branch-and-reduce approach to global optimization. J. Global. Optimizat. **8**, 107–138 (1996)
49. Sahinidis, N.V.: BARON: A general purpose global optimization software package. J. Global. Optimizat. **8**, 201–205 (1996)
50. Shcherbina, O., Neumaier, A., Sam-Haroud, D., Vu, X.-H., Nguyen, T.-V.: Benchmarking global optimization and constraint satisfaction codes. In Global Optimization and Constraint Satisfaction, pages 211–222. Springer Berlin Heidelberg, (2003)
51. Smith, E.M.B., Pantelides, C.C.: Global optimisation of non-convex MINLPs. Comput. Chem. Eng. **21**, S791–S796 (1997)
52. Smith, E.M.B., Pantelides, C.C.: A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex minlps. Comput. Chem. Eng. **23**, 457–478 (1999)
53. Tawarmalani, M., Sahinidis, N.: Convex extensions and envelopes of lower semi-continuous functions. Math. Program. **93**, 247–263 (2002)
54. Tawarmalani, M., Sahinidis, N.V.: Global optimization of mixed-integer nonlinear programs: a theoretical and computational study. Math. Program. **99**(3), 563–591 (2004)
55. Tawarmalani, M., Sahinidis, N.V.: A polyhedral branch-and-cut approach to global optimization. Math. Program. **103**, 225–249 (2005)
56. Tuy, H.: $\mathcal{D}(\mathcal{C})$-optimization and robust global optimization. J. Global. Optimizat. **47**, 485–501 (2010)
57. Vigerske, S., Gleixner, A.: SCIP: global optimization of mixed-integer nonlinear programs in a branch-and-cut framework. Optimiz. Method. Software. **33**(3), 563–593 (2018)
58. Wäechter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. **106**, 25–57 (2006)
59. Zamora, J.M., Grossmann, I.E.: A branch and contract algorithm for problems with concave univariate, bilinear and linear fractional terms. J. Global. Optimizat. **14**, 217–249 (1999)