Journal of Global Optimization manuscript No.
(will be inserted by the editor)

Extended Reverse Convex Programming: An Active-Set Approach to Global Optimization

Gene A. Bunin

Received: date / Accepted: date

Abstract Reverse convex programming (RCP) represents an important class of global optimization problems consisting of concave cost and inequality constraint functions. While useful in many practical scenarios due to the frequent appearance of concave models, a more powerful, though somewhat abstractly recognized, characteristic of the RCP problem is its ability to approximate a very general class of nonconvex nonlinear programming (NLP) problems to arbitrary precision. The goal of the present work is to make this abstract idea concrete by formalizing an extended RCP framework with a nearly algorithmic procedure to approximate the general NLP problem by an RCP one. Furthermore, an active-set RCP algorithm, which may be seen as an improved and modernized version of Ueing's method [39], is proposed and described in detail. Some preliminary results are presented for several NLP problems to demonstrate the potential of the proposed framework together with its shortcomings.

Keywords Reverse convex programming · Concave programming · Piecewise-concave approximation · Complete global optimization methods · Active-set methods

1 Introduction

The following reverse convex programming (RCP) problem is considered:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & c^T x \\ \text{subject to} & g_i(x) \leq 0, \quad i = 1,...,n_g \\ & Cx = d \end{array} , \tag{1}$$

with $x \in \mathbb{R}^{n \times 1}$ the vector of variables, $g : \mathbb{R}^n \to \mathbb{R}$ a set of n_g *concave* constraint functions, and $C \in \mathbb{R}^{n_C \times n}$, $d \in \mathbb{R}^{n_C \times 1}$ the matrix and vector defining the linear equality constraints. A linear cost, $c \in \mathbb{R}^{n \times 1}$, is chosen here as it eases the presentation and makes the theoretical

Gene A. Bunin
Automatic Control Laboratory, Ecole Polytechnique Fédérale de Lausanne, Lausanne, CH-1015
Tel.: +41-21-693-2564
E-mail: gene.bunin@epfl.ch

discussions (a bit) simpler – the nonlinear version being equivalent to (1) via a simple epigraph transformation (see, e.g., [21]). Problem (1) will be referred to as the "standard" RCP form in all the discussion that follows.

The following assumptions are made throughout this work:

A1. $g_i(x)$ is concave and differentiable for all i.
A2. rank $C = n_C$ and $n_C < n$.
A3. The feasible space defined by $g_i(x) \leq 0$ and $Cx = d$ is compact.
A4. $\|c\| > 0$.

Assumption A1 is mostly a convenience assumption – it is expected that all of the theory in this work generalizes easily, though perhaps with a bit more effort, to quasiconcave and nondifferentiable cases. A2 ensures that the linear equalities are not redundant and the problem not trivial (or trivially infeasible), while A3 ensures that the problem admits a global minimum. A4 limits the discussion (though not the applicability) of the methods presented to strictly optimization (rather than optimization and feasibility) problems.

Judging from the literature, formal study of RCP problems appears to be nearly fifty years of age, with the 1966 paper by Rosen [29] usually credited as the first work where an RCP problem is solved to local optimality. The term "reverse convex" is credited to Meyer [24] in a slightly later work, and similar contemporary developments in the context of geometric programs (with the term "complementary convex programming") are due to Avriel [2]. First attempts to solve RCP problems to global optimality may be simultaneously credited to Ueing [39], in a general mathematical context, and to Rozvany [31, 32], in the context of structural optimization (under the name of "concave programming"). The key 1980 paper by Hillestad and Jacobsen [14] represents the first attempt to formalize the RCP problem – reviewing its global optimality properties and proposing an edge search, cutting-plane algorithm that would be refined in later works [4, 13, 16]. The work by Tuy [36] represents the first concrete link between RCP and canonical D.C. (CDC) programming [15, 37, 38], as it shows how Problem (1) can be cast into an equivalent CDC form. Finally, some branch-and-bound methods have treated RCP problems indirectly, as the concave function (especially if it is separable) is one of those with an easily defined convex underestimator – see, e.g., [9] or, more recently, [40].

Practical interest in RCP problems is justified by the common appearance of a concave cost and/or constraints in application. Some examples include:

– integer constraints in MINLP problems, as these constraints may be reformulated into continuous concave ones [34, 15, 10],
– the minimal-excitation condition in real-time optimization schemes [20, 28],
– non-overlap constraints or a distance maximization criterion in packing and placement problems [23, 18],
– the cost in concave minimization problems [22, 27, 15].

Although these examples constitute an important argument for why "RCP matters", it is the author's belief that the real potential of the RCP problem goes much further than this, and as such the main message and contribution of this work is in showing that the general NLP problem

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f_0(x) \\ \text{subject to} & g_{0,i}(x) \leq 0, \quad i = 1, ..., n_{g0} \\ & h_{0,i}(x) = 0, \quad i = 1, ..., n_{h0} \end{array} , \qquad (2)$$

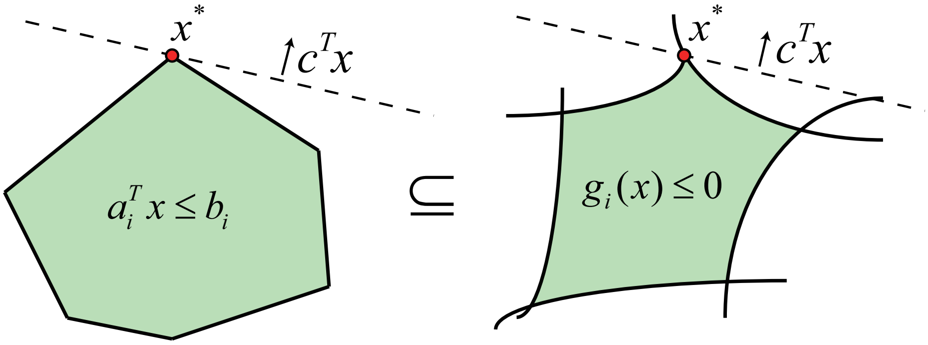

Fig. 1 The conceptual geometric link between linear (left) and reverse convex (right) programming, with the former being a special case of the latter. In both cases, n locally linearly-independent constraints are required to define a local minimum for the linear cost.

may be approximated to arbitrary precision by an RCP problem and solved to global optimality in the RCP framework, provided that the functions f_0, g_0, and h_0 (a) satisfy a certain Lipschitz criterion and (b) satisfy a factorability condition (see Section 3). To the best of the author's knowledge, only Rozvany [31,32] has made explicit mention of this link, and it is indeed Rozvany's – and, in part, Zangwill's [41] – work on piecewise-concave approximations that largely motivates the results presented in this paper.

The other major contribution of the present work lies in the development of an algorithm tailored specifically to RCP problems. This is, largely, an extension on the work of Ueing [39], who used the property that a local (and thereby global) minimum of an RCP problem must lie on the intersection of n active and linearly-independent constraints to propose a combinatorial search that is, essentially, an active-set method (this property, a generalization of what is well known for the linear case, is illustrated qualitatively in Fig. 1). The two major criticisms of Ueing's approach [14], i.e. that it requires *strict* concavity to avoid theoretical complications and that it has unfavorable complexity, are treated and remedied to a great extent in this work. Although, as already stated, there exist several ways to approach Problem (1) (e.g. via CDC programming or a branch-and-bound search), the active-set approach is chosen here on both academic and practical grounds. With regard to the former, the method is scientifically fascinating as it presents a fundamentally different approach to solving NLP problems by searching for the solution on the space of active constraints rather than on the space of problem variables, thereby replacing the curse of dimensionality with a different sort of curse that may, in some instances, be more mild. Practically, this approach is of some value as it (a) allows for a guaranteed upper bound on the total computational effort, (b) does not require an initial point, (c) does not suffer from disjoint feasible regions, and (d) can be shown to break the curse of dimensionality for certain (though perhaps academic) examples.

The organization of this paper is as follows. As the RCP problem acts as the major theoretical backbone, all of the relevant theory regarding the characterization of a global minimum and how it may be calculated is provided in Section 2. This is, in many ways, a review of the concepts previously presented in [39] and [14], but additional care is taken to define a "regular" RCP problem that allows for the application of Ueing's method to problems with nonstrict concavity, and some results are derived on how the regularity assumption may either be verified or enforced. Section 3 then presents the extended RCP methodology and the semi-algorithmic steps to approximate (2) by (1). The RCP algorithm is described in Section

4 – particularly, a number of RCP-specific fathoming techniques, together with well-known domain reduction techniques, are presented as a means of speeding up the convergence of the active-set search. A number of numerical examples are then presented in Section 5 so as to illustrate the noticeable strengths and shortcomings of the proposed methodology. Finally, Section 6 serves to conclude the paper with an outlook on how further improvements based on the idea of homotopy could eventually culminate in an RCP-based general-purpose NLP solver.

2 Reverse Convex Programming Theory

As detailed in both [39] and [14], an RCP problem defined by strictly concave inequality constraints benefits from the following two necessary conditions that every local minimum, x^*_{loc}, must satisfy:

- x^*_{loc} is defined completely by the intersection of n active constraints that are linearly independent at x^*_{loc}.
- x^*_{loc} may be found by solving the (convex) reverse problem where the cost $c^T x$ is *maximized* subject to the complements of the n constraints in question:

$$x^*_{loc} = \arg \begin{array}{ll} \underset{x}{\text{maximize}} & c^T x \\ \text{subject to} & -G_a(x) \leq \mathbf{0} \end{array}, \tag{3}$$

  with $G_a(x)$ representing the n active constraints defining the local minimum.

These two properties may be exploited to calculate the global minimum in finite time by simply solving (3) for all possible $\binom{n_g}{n}$ active sets and declaring the feasible point with the lowest cost as the global minimum, x^*. However, as was pointed out in [14], that such an approach would not be innately promising for two major reasons.

First, the number of active sets to be checked will almost always be unacceptable, from a computational point of view, for practical problems – methods to overcome this are treated in Section 4. The second issue, and the one addressed in this section, is that the properties above do not always extend to cases when the constraints are concave but not strictly concave, thereby making the theory inapplicable to – or, at least, not rigorous for – most pertinent problems. This latter challenge may be dealt with by introducing of the notion of "RCP regularity", which, in this work, will refer to the minimalist set of conditions needed to ensure that the global minimum (though not necessarily every local minimum) will retain the two properties above.

In the two-part discussion that follows, the idea of RCP regularity is defined first, followed by the proof of the two necessary conditions for RCP problems that satisfy the regularity assumption.

2.1 Regular RCP Problems

In simplest terms, saying that an RCP problem is "regular" by the convention chosen here is equivalent to saying that its global minimum (or one of its global minima, if nonunique) is *locally unique*. The formal definition is given as follows:

Definition 1 (RCP Regularity)

Let x^* be a (possibly nonunique) global minimum of Problem (1). Problem (1) is said to be regular if x^* is a locally unique minimum:

$$\begin{array}{c} \nexists \delta x^* \in \mathbb{R}^n : \|\delta x^*\| > 0,\ c^T(x^* + t\delta x^*) = c^T x^*,\ g_i(x^* + t\delta x^*) \leq 0\ (\forall i = 1,...,n_g) \\ C(x^* + t\delta x^*) = d,\ \forall t \in [0,1] \end{array}, \quad (4)$$

i.e. there does not exist a direction, δx^*, that locally is both feasible and yields the same cost as x^*.

The following theorem presents a sufficient condition for which the RCP problem of form (1) may be proven regular:

Theorem 1 (Sufficient Condition for RCP Regularity) *Let x^* be a global minimum of Problem (1) and let δx^* denote a (non-zero) direction. Index by $i_{L0} = \{i : g_i(x^* + t\delta x^*) = 0, \forall t \in [0,1]\}$ those inequality constraints that are active at x^* and null in δx^*. Problem (1) is regular, and x^* a locally unique global minimum, if:*

$$\forall \delta x^* \in N\left(\begin{bmatrix} c^T \\ C \end{bmatrix}\right),\ \nexists \lambda \in \mathbb{R}_+^{n_g}, \mu \in \mathbb{R}^{n_C} : c^T + \sum_{i \in i_{L0}} \lambda_i \nabla g_i(x^*)^T + \sum_{i=1}^{n_C} \mu_i C_i = \mathbf{0}. \quad (5)$$

Proof Suppose that x^* is not a locally unique global minimum. This implies that there exists a direction δx^* with $t \in [0,1]$ such that:

$$\begin{array}{l} c^T(x^* + t\delta x^*) = c^T x^*,\ g_i(x^* + t\delta x^*) \leq 0\ (\forall i = 1,...,n_g) \\ C(x^* + t\delta x^*) = Cx^* = d,\ \forall t \in [0,1] \end{array}. \quad (6)$$

The immediate conclusion is that this can only hold if δx^* lies in the null space of both the cost and equality constraints, i.e. $\delta x^* \in N\left(\begin{bmatrix} c^T \\ C \end{bmatrix}\right)$, thereby limiting the directions in question to those that satisfy this condition.

Consider now the stationarity properties of $x^* + t\delta x^*$. This is done by first dividing the constraints $g_i(x)$ into four categories:

- constraints that are active at x^* and are null in δx^* (as indexed by i_{L0}),
- constraints that are active at x^* and are linear but not null in δx^*, as indexed by $i_L = \{i : g_i(x^*) = 0,\ g_i(x^* + t\delta x^*) \neq g_i(x^*)\ (\forall t \in (0,1]),\ \nabla g_i(x^* + t\delta x^*) = \nabla g_i(x^*)\ (\forall t \in [0,1])\}$,
- constraints that are active at x^* and are nonlinear in δx^*, as indexed by $i_{NL} = \{i : g_i(x^*) = 0,\ \exists t \in (0,1] : \nabla g_i(x^* + t\delta x^*) \neq \nabla g_i(x^*)\}$,
- constraints that are inactive at x^*, as indexed by $i_I = \{i : g_i(x^*) < 0\}$.

For a small enough $t > 0$, it is seen that:

- $g_i(x^* + t\delta x^*) < 0,\ \forall i \in i_I$.
- $g_i(x^* + t\delta x^*) < 0,\ \forall i \in i_{NL}$ by the nonlinearity of these constraints in δx^* and the fact that the direction δx^* must be locally feasible for these constraints.
- $g_i(x^* + t\delta x^*) < 0,\ \forall i \in i_L$ by the virtue of these constraints being non-null in δx^* and the fact that the direction δx^* must be globally feasible for these constraints.

As all of these constraints are inactive at $x^* + t\delta x^*$, it follows that they cannot affect the stationarity properties of this point and may be ignored there. However, as the point is a global minimum, it is nevertheless stationary, i.e. the set of locally feasible, strict descent directions is empty:

$$\begin{array}{c} \{x : c^T(x - x^* - t\delta x^*) < 0,\ C(x - x^* - t\delta x^*) = 0, \\ \nabla g_i(x^* + t\delta x^*)^T(x - x^* - t\delta x^*) \leq 0,\ \forall i \in i_{L0}\} = \varnothing \end{array}, \tag{7}$$

which, by the Lemma of Farkas [19], is equivalent to:

$$\exists \lambda \in \mathbb{R}_+^{n_g}, \mu \in \mathbb{R}^{n_C} : c^T + \sum_{i \in i_{L0}} \lambda_i \nabla g_i(x^* + t\delta x^*)^T + \sum_{i=1}^{n_C} \mu_i C_i = \mathbf{0}, \tag{8}$$

and simplifies to (5) as $\nabla g_i(x^* + t\delta x^*) = \nabla g_i(x^*),\ \forall i \in i_{L0}$ by the virtue of these constraints being null (and therefore linear) in δx^*. As this condition is necessary for a locally nonunique global minimum x^*, it follows that the failure to meet it for all possible $\delta x^* \in N\left(\begin{bmatrix} c^T \\ C \end{bmatrix}\right)$ proves that x^* must be locally unique. □

While there is no readily apparent *algorithmic* way to verify this condition (as one cannot, in general, enumerate all possible δx^* or know x^* in advance), the preliminary results in Section 5 will illustrate that (5) can be proven to hold quite easily for at least some problems. Furthermore, certain types of NLP problems (e.g. concave minimization problems with a stricty concave cost) naturally lead to regular RCP problems upon transformation. The following proposition is nevertheless given as a general-purpose means of ensuring regularity for problems where (5) cannot be verified.

Proposition 1 (Enforcing RCP Regularity for Any RCP Problem)
Consider the following RCP problem:

$$\begin{array}{ll} \underset{x,s}{\text{minimize}} & s \\ \text{subject to} & c^T x - \left(\alpha_r \sum_{i=1}^{n} x_i^2\right) - s \leq 0 \\ & g_i(x) \leq 0, \quad i = 1, ..., n_g \\ & Cx = d \end{array}, \tag{9}$$

with $s \in \mathbb{R}$ a slack variable and $\alpha_r > 0$ a regularizing scalar. Problem (9) becomes equivalent to Problem (1) as $\alpha_r \to 0$, and is a regular RCP for any $\alpha_r > 0$.

Proof The equivalence is easily seen as the added constraint, which clearly must be active at the minimum, approaches $c^T x^* = s^*$ as $\alpha_r \to 0$.

Consider now Problem (9) in standard RCP form by making the following substitutions:

$$\hat{c}^T = [\mathbf{0}_{1 \times n}\ \ 1],\ \hat{x} = \begin{bmatrix} x \\ s \end{bmatrix}, \tag{10}$$

so that (9) becomes:

$$\begin{array}{ll} \underset{\hat{x}}{\text{minimize}} & \hat{c}^T \hat{x} \\ \text{subject to} & c^T x - \left(\alpha_r \sum_{i=1}^{n} x_i^2\right) - s \leq 0 \\ & g_i(x) \leq 0, \quad i = 1, ..., n_g \\ & Cx = d \end{array}. \tag{11}$$

To verify regularity, consider the "worst-case" scenario of $n_C = 0$, since this allows for more potential $\delta\hat{x}^*$. Without any additional assumptions on c, it is clear that $\delta\hat{x}^*$ could be any direction *except* $[\mathbf{0}_{1\times n}\;\; 1]^T$, i.e. it must include a step in at least one of the x variables to satisfy $\delta\hat{x}^* \in N(\hat{c}^T)$. However, the constraint $c^T x - \alpha_r \Sigma_{i=1}^n x_i^2 - s \leq 0$ is nonlinear in any such $\delta\hat{x}^*$ when $\alpha_r > 0$, which allows the following implication:

$$\begin{aligned} c^T x^* - \left(\alpha_r \sum_{i=1}^n (x_i^*)^2\right) - s^* = 0 \Rightarrow \\ c^T(x^* + t\delta x^*) - \left(\alpha_r \sum_{i=1}^n (x_i^* + t\delta x_i^*)^2\right) - (s^* + t\delta s^*) < 0 \end{aligned} \tag{12}$$

for any feasible $\hat{x}^* + t\delta\hat{x}^*$ and $t > 0$ sufficiently small. Since the above constraint must be active at any minimum, it follows that any minimum, and thus any global minimum, must be locally unique, since a small perturbation in any direction $\delta\hat{x}^*$ will render the constraint inactive and thereby lead to a point that cannot be a minimum. $\square$

Proposition 1 thereby offers a simple way to ensure regularity with the addition of an auxiliary variable and constraint. The scalar α_r represents a potentially unwanted tradeoff, however, as setting $\alpha_r \approx 0$ leads to negligible approximation error but may lead to numerical issues in practice (due to rounding errors), thereby losing regularity. Setting α_r as sufficiently large would avoid this but could introduce significant approximation error, which may not be tolerable. As such, Proposition 1 is only advocated as a last resort to ensure regularity, though it may work very well in certain cases.

Some examples illustrating RCP regularity are now given.

Example 1 (Non-regular RCP Problems)

As perhaps the simplest example of a non-regular RCP problem, consider the linear programming problem of minimizing a single variable over a hypercube:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & x_1 \\ \text{subject to} & \mathbf{0} \leq x \leq \mathbf{1} \end{array} , \tag{13}$$

where the global minimum is not unique as any of the points in the connected space corresponding to $x_1 = 0$ are globally optimal. To show this using Theorem 1, it is sufficient to consider $\delta x^* = [0\;\; 1\;\; \mathbf{0}_{1\times(n-2)}]^T$, the constraint $-x_1 \leq 0$, and the corresponding multiplier $\lambda = 1$, as this provides a counterexample to (5).

A similar example appears in [14]:

$$\begin{array}{ll} \underset{x_1,x_2}{\text{minimize}} & x_1 \\ \text{subject to} & -x_1^2 + x_2 - 1 \leq 0 \\ & -x_1 \leq 0 \\ & -x_2 \leq 0 \end{array} , \tag{14}$$

and fails to satisfy the regularity condition by the same counterexample.

Example 2 (RCP Regularity of Strict Concave Minimization Problems)

Consider the following concave minimization problem and its equivalent reformulation:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & -\sum_{i=1}^n (x_i - 0.5)^2 \\ \text{subject to} & \mathbf{0} \leq x \leq \mathbf{1} \end{array} \Leftrightarrow \begin{array}{ll} \underset{x,s}{\text{minimize}} & s \\ \text{subject to} & -\sum_{i=1}^n (x_i - 0.5)^2 - s \leq 0 \\ & \mathbf{0} \leq x \leq \mathbf{1} \end{array} , \tag{15}$$

where the latter is easily proven regular by the same arguments as in Proposition 1. This is expected since the problem clearly admits 2^n nonunique (but locally unique) global minima in the corners of the hypercube.

2.2 Necessary Conditions of Global Optimality for Regular RCP Problems

Assuming RCP regularity, the two necessary conditions may now be proven. The following lemma is presented first:

Lemma 1 (Uniqueness of an Optimum of a Cone Linear Program)
Let $J_a \in \mathbb{R}^{n_a \times n}$ and consider the following LP:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & c^T x \\ \text{subject to} & J_a(x-x^*) \leq \mathbf{0} \\ & Cx = d \end{array} \quad , \tag{16}$$

with a minimum x^*_{cone} that satisfies $J_a(x^*_{cone} - x^*) = \mathbf{0}$. A necessary and sufficient condition for x^*_{cone} to be the unique solution to (16) is that:

$$\text{rank} \begin{bmatrix} C \\ J_a \end{bmatrix} = n. \tag{17}$$

Proof To prove necessity, suppose that the minimum, x^*_{cone}, is unique, but that:

$$\text{rank} \begin{bmatrix} C \\ J_a \end{bmatrix} < n. \tag{18}$$

Since x^*_{cone} is stationary, it follows that the gradients of the cost and the constraints are linearly dependent, and that, by the rank-nullity theorem:

$$\text{rank} \begin{bmatrix} C \\ J_a \end{bmatrix} = \text{rank} \begin{bmatrix} c^T \\ C \\ J_a \end{bmatrix} < n \Leftrightarrow N\left(\begin{bmatrix} c^T \\ C \\ J_a \end{bmatrix} \right) \neq \varnothing, \tag{19}$$

i.e. that there exists a direction satisfying all of the active constraints while maintaining the same cost, which contradicts the uniqueness of x^*_{cone}.

To prove sufficiency, it is enough to note that the linear system

$$\begin{array}{l} J_a(x-x^*) = 0 \\ Cx = d \end{array} \tag{20}$$

is solved uniquely by x^*_{cone} when (17) holds. □

The two necessary conditions follow.

Theorem 2 (First Necessary Condition of Global Optimality for a Regular RCP)
Let x^* be a locally unique global minimum of RCP problem (1), and let J_a be the Jacobian matrix of the active set of inequality constraints at x^*. The condition (17) must hold and represents a necessary condition for x^*.

Proof Consider the version of Problem (1) that has been linearized around x^*:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & c^T x \\ \text{subject to} & g_i(x^*) + \nabla g_i(x^*)^T (x - x^*) \leq 0, \;\; i = 1, ..., n_g \\ & Cx = d \end{array} \quad . \tag{21}$$

This problem has a tighter feasible region than the original due to the concavity of the inequality constraints, and must admit x^* as a single and locally unique global minimum due to the convexity of the problem and the fact that the local uniqueness of x^* cannot be lost by tightening the feasible space while retaining x^* as a feasible point. Noting that ignoring the inactive constraints in (21) does not change the solution, and that $g_i(x^*) = 0$ for all active constraints, (21) may be rewritten as (16). x^* being the unique mininum, (17) follows directly from Lemma 1. □

Theorem 3 (Second Necessary Condition of Global Optimality for a Regular RCP)
Let x^* be a locally unique global minimum of RCP problem (1), and let $G_a(x)$ denote the set of active inequality constraints at x^*, i.e. $G_a(x^*) = 0$. It follows that:

$$\begin{array}{rl} x^* = \arg \underset{x}{\text{maximize}} & c^T x \\ \text{subject to} & -G_a(x) \leq \mathbf{0} \\ & Cx = d \end{array} \quad . \tag{22}$$

Proof With J_a the Jacobian of the active set at x^*, consider this time the reverse linearized problem:

$$\begin{array}{ll} \underset{x}{\text{maximize}} & c^T x \\ \text{subject to} & -J_a(x - x^*) \leq \mathbf{0} \\ & Cx = d \end{array} \quad . \tag{23}$$

By the argument of Theorem 2, the local of uniqueness of x^* for (1) ensures that x^* is the unique minimum for (21). Since (21) and (23) have the same stationarity condition, it follows that x^* also solves (23) uniquely. Due to the convexity of $-G_a(x)$, it is also clear that (23) has a greater feasible region than (22). As such, x^* must also solve (22) uniquely, as tightening the feasible space cannot compromise the local uniqueness of x^*, x^* being feasible for both cases. □

The proof of Theorem 3 may be interpreted geometrically as shown in Fig. 2. It is important to emphasize that the linearized problems used in the proofs are purely conceptual – there is no need to solve them in implementation.

Theorems 2 and 3 create the foundations of a combinatorial, active-set procedure. The former makes it clear that there exists an active set of the n_C equality constraints and $n - n_C$ inequality constraints that are locally linearly independent at x^* and define x^* by their active manifolds. The latter, in turn, provides a tractable approach to calculate x^* given its corresponding active set. Together, the two theorems provide a guarantee that checking all of the possible active sets and solving for the corresponding x^* candidates is bound to generate a set of points that includes the global minimum (provided, of course, that the problem is regular and feasible).

To finish, one notes that not having a regular RCP does not automatically guarantee the failure of such a method, but only loses the guarantee of its success. The main reason for this, as already pointed out in [14], is the inability to guarantee that the solution to (22) is unique, which may result in (22) yielding an x^* candidate that is infeasible for (1), even when a solution that satisfies the constraints of (1) exists.

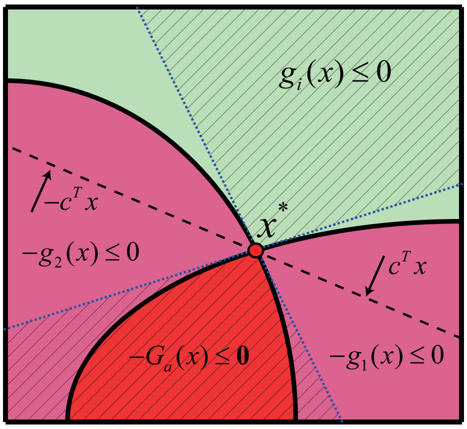

Fig. 2 Geometrical illustration of Theorem 3 and its proof, with the light and dark areas denoting the feasible and infeasible spaces, respectively, of an RCP problem whose global minimum is defined by two concave inequality constraints. The shaded regions represent the feasible spaces of the linearized problems. Clearly, as x^* is a locally unique global minimum for the original problem, it is also the unique minimum for the original problem linearized around x^*. It is also clear that the linearized reverse problem has the same solution as the linearized original problem, and, as the infeasible-side linearized set is a superset of $-G_a(x) \leq \mathbf{0}$, it follows that solving the reverse problem over the latter also yields x^*.

3 The Piecewise-Concave Approximation and Extended RCP

To the best of the author's knowledge, solving a general NLP problem (2) via an RCP approximation is an idea that has only been mentioned in passing by Rozvany [31,32], with no formal systematic procedure of when and how such an approximation could be carried out ever being proposed. This is the essential goal of this section. First, the piecewise-concave function is defined and it is shown how any Lipschitz-continuous univariate function may be approximated arbitrarily well by a piecewise-concave one. Using the piecewise-concave approximation as a link, it is then shown how the general NLP (2) meeting a fairly weak factorability assumption may be approximated as an RCP. Finally, a procedure for carrying out the NLP $\rightarrow$ RCP approximation is proposed and demonstrated on some example problems. As this essentially allows for non-RCP problems to be solved as RCP ones, the term "extended reverse convex programming" is used to describe the proposed methodology.

3.1 Piecewise-Concave Functions as Approximators

The function $p(x)$ is called "piecewise-concave" if:

$$p(x) = \max_{i=1,\ldots,n_p} p_i(x), \tag{24}$$

with each of the n_p functions $p_i(x)$ concave. The first formal presentation of this function and its properties are due to Zangwill [41], with multiple relevant works by Rozvany [30–32] appearing at around the same time. Some later work has also looked at the use of these functions in economics problems [3,7], as well as in a more general mathematical context [11].

The ability of a piecewise-concave function to approximate a general function arbitrarily well has been previously mentioned without proof [41,32], but may be proven, with a little work, under fairly weak assumptions for different choices of $p_i(x)$. Of particular interest here is the case where:

- $p(x) : \mathbb{R} \to \mathbb{R}$ is univariate, as this allows for a number of algorithmic conveniences.
- $p_i(x) = \beta_{2,i}x^2 + \beta_{1,i}x + \beta_{0,i}$, i.e. each piece is quadratic, as this will later lead to convex/concave quadratic constraints in the extended RCP problem, which are easier to treat both theoretically and numerically than general convex/concave constraints.

The following theorem proves the existence of a $p(x)$ that meets the above criteria and approximates a Lipschitz-continuous univariate function to arbitrary precision.

Theorem 4 (Approximation by a Piecewise Maximum of Concave Quadratics) *Let $f(x) : \mathbb{R} \to \mathbb{R}$ be a κ-Lipschitz function on a closed interval $x \in [\underline{x}, \overline{x}]$, so that:*

$$-\kappa < \left.\frac{df}{dx}\right|_x < \kappa, \;\; \forall x \in [\underline{x}, \overline{x}], \tag{25}$$

and consider the discretization $\underline{x}, \underline{x} + \Delta x, ..., \overline{x} - \Delta x, \overline{x}$ with $\Delta x = \frac{\overline{x} - \underline{x}}{n_{app} - 1}$, i.e. the discretization of the interval into n_{app} evenly spaced points. Letting

$$p(x) = \max_{i=1,...,n_{app}-1} (\beta_{2,i}x^2 + \beta_{1,i}x + \beta_{0,i}), \tag{26}$$

it follows that there exist sets of coefficients $\beta_2 \in \mathbb{R}_-^{n_{app}-1}$ and $\beta_1, \beta_0 \in \mathbb{R}^{n_{app}-1}$ such that:

$$\lim_{n_{app} \to \infty} \max_{x \in [\underline{x}, \overline{x}]} |f(x) - p(x)| = 0. \tag{27}$$

Proof The theorem is proven by providing a single choice of quadratic functions that possess the desired properties. Consider the interval $[x_i, x_{i+1}]$, where x_i and x_{i+1} are neighboring discretization points (with $x_{i+1} = x_i + \Delta x$), and define the function $p_i(x)$ as $p_i(x) = \beta_{2,i}x^2 + \beta_{1,i}x + \beta_{0,i}$. Let $p_i(x)$ satisfy the following criteria:

$$\begin{array}{l} p_i(x_i + 0.5\Delta x) = f(x_i + 0.5\Delta x) \\ \left.\dfrac{dp_i}{dx}\right|_{x_i} = 2\kappa \\ \left.\dfrac{dp_i}{dx}\right|_{x_{i+1}} = -2\kappa \end{array}, \tag{28}$$

which translate into the following linear system:

$$\begin{bmatrix} \beta_{2,i} \\ \beta_{1,i} \\ \beta_{0,i} \end{bmatrix} = \begin{bmatrix} (x_i + 0.5\Delta x)^2 & x_i + 0.5\Delta x & 1 \\ 2x_i & 1 & 0 \\ 2x_{i+1} & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} f(x_i + 0.5\Delta x) \\ 2\kappa \\ -2\kappa \end{bmatrix}. \tag{29}$$

This system has a solution as long as $x_i \neq x_{i+1}$ (as this ensures that the inverted matrix is full rank and therefore invertible), with the resulting $p_i(x) = -\frac{2\kappa}{\Delta x}x^2 - 2\kappa(1 - \frac{2x_{i+1}}{\Delta x})x - \frac{2\kappa x_i^2}{\Delta x} - 0.5\kappa\Delta x - 2\kappa x_i + f(x_i + 0.5\Delta x)$. By enforcing the three conditions of (28), the following properties are guaranteed:

1. $p_i(x)$ is quadratic and concave, with $\beta_{2,i} = -\frac{2\kappa}{\Delta x} < 0$.

2. $p_i(x)$ is a strict underestimator of $f(x)$ at all points outside the interval (x_i, x_{i+1}). This may be proven as follows. First, consider the function $L_i(x) = f(x_i + 0.5\Delta x) - \kappa|x - x_i - 0.5\Delta x|$, which is nothing but the Lipschitz "sawtooth" underestimator of $f(x)$, generated around $x = x_i + 0.5\Delta x$. It follows from the definition of the Lipschitz constant that $L_i(x) < f(x),\ \forall x \setminus \{x_i + 0.5\Delta x\}$. Analyzing the intersection of $L_i(x)$ and $p_i(x)$, one sees that $L_i(x) = p_i(x)$ at $x = x_i, x_{i+1}$. Consider now the function $\overline{p}_i(x) = 2\kappa x + f(x_i + 0.5\Delta x) - 2\kappa x_i - 0.5\kappa\Delta x$, which is the linearization of $p_i(x)$ at $x = x_i$. It is evident that $\overline{p}_i(x) \leq L_i(x),\ \forall x \in [\underline{x}, x_i]$, as both are linear and intersect at x_i, with $\overline{p}_i(x)$ having a greater positive slope. From the concavity of $p_i(x)$, it is also true that $p_i(x) \leq \overline{p}_i(x),\ \forall x$. It follows that:

$$p_i(x) \leq \overline{p}_i(x) \leq L_i(x) < f(x),\ \forall x \in [\underline{x}, x_i]. \tag{30}$$

A symmetrical analysis around x_{i+1} yields a symmetrical result, which then leads to the following statement:

$$p_i(x) < f(x),\ \forall x \in [\underline{x}, x_i] \cup [x_{i+1}, \overline{x}]. \tag{31}$$

3. $p_i(x)$ approximates $f(x)$ perfectly (with zero error) at $x = x_i + 0.5\Delta x$.
4. The interval for which $p_i(x) = p(x)$ is a strict subinterval of $[x_i - 0.5\Delta x, x_{i+1} + 0.5\Delta x]$, i.e. $p_i(x)$ can only be a "piece" of the piecewise-maximum function in the interior of this interval. This may be proven as follows. Let $p_{i-1}(x)$ denote a corresponding concave quadratic function for the neighboring interval $[x_{i-1}, x_i]$, and consider the difference:

$$p_{i-1}(x) - p_i(x) = -4\kappa(x - x_i) + f(x_i - 0.5\Delta x) - f(x_i + 0.5\Delta x). \tag{32}$$

For $x = x_i - 0.5\Delta x$, one may build on the result of (31), which states that $p_i(x_i - 0.5\Delta x) < f(x_i - 0.5\Delta x)$, and Property 3, which states that $p_{i-1}(x_i - 0.5\Delta x) = f(x_i - 0.5\Delta x)$, to arrive at the following:

$$\begin{array}{l} -p_i(x_i - 0.5\Delta x) > -f(x_i - 0.5\Delta x) \\ p_{i-1}(x_i - 0.5\Delta x) = f(x_i - 0.5\Delta x) \\ \Rightarrow p_{i-1}(x_i - 0.5\Delta x) - p_i(x_i - 0.5\Delta x) > 0 \end{array}, \tag{33}$$

which shows that the piece $p_{i-1}(x)$ must be strictly greater than $p_i(x)$ at $x = x_i - 0.5\Delta x$. From examining (32), it is clear that the derivative of this difference with respect to x is negative, i.e. the difference increases with decreasing x. This, in turn, implies that $p_{i-1}(x) - p_i(x) > 0$ remains true on the interval $x \in [\underline{x}, x_i - 0.5\Delta x]$, i.e. $p_i(x)$ cannot be the maximal piece on this interval. A symmetrical analysis shows that $p_{i+1}(x) - p_i(x) > 0$ for $x \in [x_{i+1} + 0.5\Delta x, \overline{x}]$, i.e. that $p_i(x)$ cannot be the maximal piece on this interval either. The overall result is thus summarized as:

$$p_i(x) < p(x),\ \forall x \in [\underline{x}, x_i - 0.5\Delta x] \cup [x_{i+1} + 0.5\Delta x, \overline{x}]. \tag{34}$$

The characteristics of $p(x)$ as defined by the $p_i(x)$ piece functions are now stated. First, by Property 3, $p(x)$ has one member that approximates $f(x)$ with zero error at the midpoint of each discretization interval $[x_i, x_{i+1}]$. By Property 2, every other member must underestimate this point. Together, these two properties guarantee that $p(x) = f(x)$ at the midpoint of every discretization interval, with Property 1 guaranteeing that all of the members of $p(x)$ are concave quadratic.

It now remains to consider the approximation error between the midpoints of the discretization intervals, for which the first step requires the identification of the Lipschitz constant of $p(x)$. By Property 4, every piece $p_i(x)$ is limited to $(x_i - 0.5\Delta x, x_{i+1} + 0.5\Delta x)$, from which it follows that the Lipschitz constant of $p(x)$ cannot exceed the Lipschitz constant of one of these pieces over the relevant interval:

$$\sup_{x \in (x_i - 0.5\Delta x, x_{i+1} + 0.5\Delta x)} \left| \frac{dp_i}{dx}\Big|_x \right| = \sup_{x \in (x_i - 0.5\Delta x, x_{i+1} + 0.5\Delta x)} \left| \frac{4\kappa}{\Delta x}(x_i - x) - 2\kappa \right| = 4\kappa. \tag{35}$$

This allows for the approximation error to be bounded with respect to any discretization interval midpoint $x_i + 0.5\Delta x$ as follows:

$$\begin{array}{l} f(x_i + 0.5\Delta x) - \kappa|x - x_i - 0.5\Delta x| \leq f(x) \\ \qquad \leq f(x_i + 0.5\Delta x) + \kappa|x - x_i - 0.5\Delta x| \\ p(x_i + 0.5\Delta x) - 4\kappa|x - x_i - 0.5\Delta x| \leq p(x) \\ \qquad \leq p(x_i + 0.5\Delta x) + 4\kappa|x - x_i - 0.5\Delta x| \Rightarrow \\ -p(x_i + 0.5\Delta x) - 4\kappa|x - x_i - 0.5\Delta x| \leq -p(x) \\ \qquad \leq -p(x_i + 0.5\Delta x) + 4\kappa|x - x_i - \Delta x| \\ \Rightarrow f(x_i + 0.5\Delta x) - p(x_i + 0.5\Delta x) - 5\kappa|x - x_i - 0.5\Delta x| \leq f(x) - p(x) \\ \qquad \leq f(x_i + 0.5\Delta x) - p(x_i + 0.5\Delta x) + 5\kappa|x - x_i - 0.5\Delta x| \\ \Rightarrow |f(x) - p(x)| \leq 5\kappa|x - x_i - 0.5\Delta x|, \quad \forall x \in [\underline{x}, \overline{x}] \end{array} . \tag{36}$$

As this is true for any discretization interval midpoint, it is also true for $x_{i+1} + 0.5\Delta x$, and so the following bound is also valid:

$$|f(x) - p(x)| \leq 5\kappa|x - x_{i+1} - 0.5\Delta x|, \quad \forall x \in [\underline{x}, \overline{x}]. \tag{37}$$

Since these two discretization points can be any two consecutive discretization points, one may, without loss of generality, assume x to lie between them, with $x = \theta(x_i + 0.5\Delta x) + (1-\theta)(x_{i+1} + 0.5\Delta x),\ \theta \in [0,1]$. Substituting this into both bounds yields:

$$\begin{array}{l} |f(x) - p(x)| \leq 5\kappa\Delta x(1-\theta) \\ |f(x) - p(x)| \leq 5\kappa\Delta x\theta \\ \Rightarrow |f(x) - p(x)| \leq 5\kappa\Delta x \max\limits_{\theta} \min(1-\theta, \theta) = 2.5\kappa\Delta x \end{array} , \tag{38}$$

and so:

$$\begin{array}{l} |f(x) - p(x)| \leq 2.5\kappa\Delta x, \ \forall x \in [\underline{x}, \overline{x}] \\ \Rightarrow \max\limits_{x \in [\underline{x}, \overline{x}]} |f(x) - p(x)| \leq 2.5\kappa\Delta x \end{array} , \tag{39}$$

which clearly goes to 0 as $\Delta x \to 0$ or, equivalently, as $n_{app} \to \infty$. □

Theorem 4 and its proof are more theoretically reassuring than constructive, as they basically suggest to take a quadratic for each pair of discretization points, to make each one so steep that it does not interfere with the others, and to then decrease Δx (increase n_{app}) until the original function is approximated arbitrarily well by $n_{app} - 1$ (possibly needle-like) parabolas. Clearly, such a scheme would not only be inefficient but also numerically unstable as $\Delta x \to 0$, due to the loss in rank in (29).

A more practical approximation technique is proposed below.

Algorithm 1 (A Piecewise-Concave Approximation by Quadratics)
User input: $f(x)$, κ, n_{app}, n_{ori} (the number of points used to represent $f(x)$, with $n_{ori} \geq n_{app}$), ρ (equal to 1 if an over-approximation is desired and -1 for an under-approximation), $\underline{x}$, $\bar{x}$.
Output: $\beta_2, \beta_1, \beta_0$.

1. Discretize the relevant variable space $x \in [\underline{x}, \bar{x}]$ into $x_{app} \in \mathbb{R}^{n_{app}}$, with x_{app} a set of n_{app} evenly spaced points between $\underline{x}$ and $\bar{x}$. Discretize the variable space into $x_{ori} \in \mathbb{R}^{n_{ori}}$ – a set of n_{ori} evenly spaced points – and define the vector $f_{ori} \in \mathbb{R}^{n_{ori}}$ as the discrete function values of $f(x)$ at these points. Set $X^r := \varnothing$.
2. Solve the following LP problem:

$$
\begin{aligned}
\underset{\beta_2,\beta_1,\beta_0,\varepsilon_a}{\text{minimize}} \quad & \rho \sum_{i=1}^{n_{ori}} \varepsilon_{a,i} \\
\text{subject to} \quad & \varepsilon_{a,i} = \beta_{2,j} x_{ori,i}^2 + \beta_{1,j} x_{ori,i} + \beta_{0,j} - f_{ori,i}, \\
& \quad i = 1,...,n_{ori}, \;\; j = 1,...,n_{app}-1 : x_{app,j} \leq x_{ori,i} \leq x_{app,j+1} \\
& \rho \varepsilon_{a,i} \geq 0, \;\; i = 1,...,n_{ori} \\
& \beta_{2,j} x_{ori,i}^2 + \beta_{1,j} x_{ori,i} + \beta_{0,j} \geq \beta_{2,k} x_{ori,i}^2 + \beta_{1,k} x_{ori,i} + \beta_{0,k} \\
& \quad i = 1,...,n_{ori}, \; k = 1,...,n_{app}-1 \\
& \quad j \neq k : x_{app,j} \leq x_{ori,i} \leq x_{app,j+1} \\
& \beta_{2,j} \leq 0, \;\; j = 1,...,n_{app}-1
\end{aligned}
\quad , \qquad (40)
$$

   where $\beta_2, \beta_1, \beta_0 \in \mathbb{R}^{n_{app}-1}$ are, as before, the coefficients of the $n_{app}-1$ quadratic approximating functions and $\varepsilon_a \in \mathbb{R}^{n_{ori}}$ are the approximation errors on the finer grid defined by x_{ori}.
3. For every combination of $i = 1,...,n_{app}-1$ and $j = 1,...,n_{app}-1$ ($i \neq j$):
   (a) Calculate $x_{r,1}$ and $x_{r,2}$ as the solutions of $\beta_{2,i}x^2 + \beta_{1,i}x + \beta_{0,i} = \beta_{2,j}x^2 + \beta_{1,j}x + \beta_{0,j}$ (or only $x_{r,1}$ if $\beta_{2,i} = \beta_{2,j}$).
   (b) If $\max\limits_{k=1,...,n_{app}-1} (\beta_{2,k}x_{r,1}^2 + \beta_{1,k}x_{r,1} + \beta_{0,k}) \leq \beta_{2,i}x_{r,1}^2 + \beta_{1,i}x_{r,1} + \beta_{0,i}$ and $\underline{x} \leq x_{r,1} \leq \bar{x}$, then append $[x_{r,1} \;\; i \;\; j]$ to X^r. Likewise, if $\max\limits_{k=1,...,n_{app}-1} (\beta_{2,k}x_{r,2}^2 + \beta_{1,k}x_{r,2} + \beta_{0,k}) \leq \beta_{2,i}x_{r,2}^2 + \beta_{1,i}x_{r,2} + \beta_{0,i}$ and $\underline{x} \leq x_{r,2} \leq \bar{x}$, then append $[x_{r,2} \;\; i \;\; j]$ to X^r.
4. Sort the matrix X^r so that its rows are in increasing order with respect to the x_r values in the first column. Define $R := [\underline{x} \;\; X^r_{1,1} \;\; 1]$. Denote by n_r the number of rows in X^r. For $k := 1,...,n_r - 1$:
   (a) If $R_{k,3} = X^r_{k,2}$, append $[X^r_{k,1} \;\; X^r_{k+1,1} \;\; X^r_{k,3}]$ to R. Otherwise, if $R_{k,3} = X^r_{k,3}$, append $[X^r_{k,1} \;\; X^r_{k+1,1} \;\; X^r_{k,2}]$ to R.
5. Append $[X^r_{n_r,1} \;\; \bar{x} \;\; n_{app}-1]$ to R.
6. Define $\kappa_p := 0$. For $k := 1,...,n_r + 1$:
   (a) If $|2\beta_{2,R_{k,3}} R_{k,1} + \beta_{1,R_{k,3}}| > \kappa_p$, set $\kappa_p := |2\beta_{2,R_{k,3}} R_{k,1} + \beta_{1,R_{k,3}}|$.
   (b) If $|2\beta_{2,R_{k,3}} R_{k,2} + \beta_{1,R_{k,3}}| > \kappa_p$, set $\kappa_p := |2\beta_{2,R_{k,3}} R_{k,2} + \beta_{1,R_{k,3}}|$.
7. Shift $\beta_{0,i} := \beta_{0,i} + 0.5\rho(\kappa + \kappa_p)\Delta x_{ori}$ for all $i = 1,...,n_{app}-1$.

Algorithm 1 builds an under- or over-approximation of the original function by breaking the relevant interval into $n_{app}-1$ subintervals and ensuring that a single concave function be the maximal element for each. This is done via the third set of constraints in (40), while the second set is used to ensure an under- or over-approximation. The discretization, x_{ori}, used for ensuring these properties, as well as for fitting the approximation to $f(x)$ via the

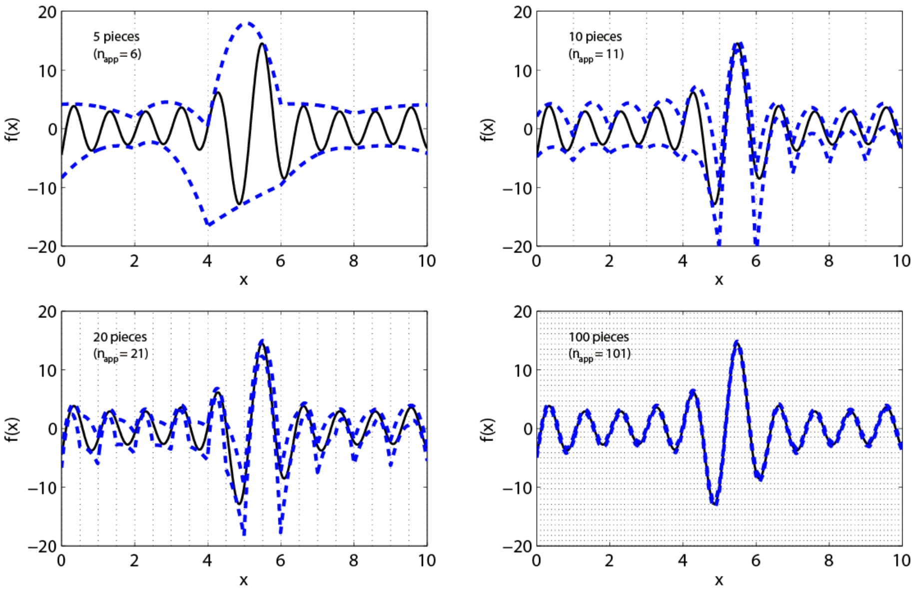

Fig. 3 Under- and over-approximations (dashed lines) of $f(x) = \Sigma_{i=1}^{5} i\cos((i+1)x+i)$ (solid line) on the interval $x \in [0,10]$ as constructed by Algorithm 1. n_{ori} is fixed at 1,000, n_{app} is varied, and the Lipschitz constant is taken as $\kappa = 70$. The dotted vertical lines show the discretization instants x_{app}.

objective, is assumed to be sufficiently fine so as to describe $f(x)$ accurately, while the discretization x_{app} will generally be much coarser so as to avoid inefficiency in the approximation. However, one still needs to account for the intergrid behavior on x_{ori} so as to ensure the robustness of the under- or over-approximation. This is done in Steps 3-7, which essentially identify which piece is maximal over which interval (the matrix R) and then use this information to calculate the Lipschitz constants of each piece and, thereby, the Lipschitz constant κ_p of the entire function $p(x)$ over $[\underline{x}, \overline{x}]$. By the same argument as in the end of the proof of Theorem 4, the calculated $p(x)$ is then shifted up/down to account for worst-case intergrid deviations.

While lacking the theoretical rigor of Theorem 4, Algorithm 1 has been found to work well in practice, with $p(x) \approx f(x)$ as both n_{app} and n_{ori} are increased (the application of Algorithm 1 to a sinuisoidal function is demonstrated in Fig. 3).

3.2 Extended RCP

The results of the previous subsection may be used to approximate the NLP problem (2) by an RCP problem provided that the functions $f_0(x)$, $g_0(x)$, $h_0(x)$ meet a certain factorability assumption. This is now formally defined.

Definition 2 (A Concave-Factorable Function)
The function $f(x) : \mathbb{R}^n \to \mathbb{R}$ is "concave factorable" if it may be expressed as a sum of a concave function, $f_{ccv}(x) : \mathbb{R}^n \to \mathbb{R}$, with the products of factorable functions $f_{ij}^0(x) : \mathbb{R}^n \to \mathbb{R}$:

$$f(x) = f_{ccv}(x) + \sum_j \prod_i f_{ij}^0(x), \tag{41}$$

with each $f_{ij}^0(x)$ in turn equal to the sum of products of other factorable functions, $f_{ij}^1(x)$:

$$f_{ij}^0(x) = \sum_j \prod_i f_{ij}^1(x), \tag{42}$$

and so forth until[1]:

$$f_{ij}^{m-1}(x) = \sum_j \prod_i f_{ij}^m((a_{ij}^F)^T x + b_{ij}^F), \tag{43}$$

i.e. sequential factoring eventually reduces the function to component functions that are univariate in affine combinations of the variables, with m finite.

This definition is particularly suitable to the needs of this paper, but is not different in essence from the decomposition or factoring methods described in, e.g., [21] or [17], where the goal is also to reduce a potentially involved function to a series of univariate expressions. The sole key difference lies in allowing the concave component, $f_{ccv}(x)$, on which no factorability/decomposition assumption is required.

The following theorem now builds on this definition to link it to the RCP problem.

Theorem 5 (Approximation of a Concave-Factorable Constraint by an RCP Set)
Let $f(x): \mathbb{R}^n \to \mathbb{R}$ be concave-factorable and consider the constraint $f(x) \leq 0$. It follows that there is an equivalent RCP constraint set that approximates the set $\{x : f(x) \leq 0\}$ to arbitrary precision, provided that the base component functions $f_{ij}^m((a_{ij}^F)^T x + b_{ij}^F)$ are Lipschitz-continuous in $(a_{ij}^F)^T x + b_{ij}^F$ over the box $X = \{x : \underline{x} \leq x \leq \overline{x}\}$ and that $f(x)$ is bounded on X.

Proof The constraint $f(x) \leq 0$ is rewritten as:

$$f_{ccv}(x) + \sum_j \prod_i f_{ij}^0(x) \leq 0. \tag{44}$$

Adding the auxiliary variable s_{ccv} and employing the epigraph transformation yields the equivalent set:

$$f_{ccv}(x) + s_{ccv} \leq 0, \;\; \sum_j \prod_i f_{ij}^0(x) - s_{ccv} \leq 0, \tag{45}$$

of which the first constraint is RCP and is lumped into the RCP set. Assume the summation in the second constraint to have n_j elements. Introducing the auxiliary variables $s_1^0, ..., s_{n_j}^0$ and applying the epigraph again yields the equivalent set:

$$\begin{array}{l} \sum_{j=1}^{n_j} s_j^0 - s_{ccv} \leq 0 \\ \prod_i f_{ij}^0(x) - s_j^0 \leq 0, \;\; j = 1, ..., n_j \end{array}, \tag{46}$$

[1] The indices i and j are not used rigorously in these definitions, since doing so would complicate the notation significantly – e.g. $f_{ij}^m(x)$ by the notation used here is ambiguous, since it could have multiple definitions depending on which "factoring path" it was derived from. For simplicity, assume that only a single path is considered.

of which the first constraint is linear and lumped into the RCP set. Assuming the product term to have n_i terms, reformulate the second set with the following equality constraints and the auxiliary variables z:

$$\begin{array}{l} \prod_{i=1}^{n_i} z_{ij}^0 - s_j^0 \leq 0, \;\; j = 1, ..., n_j \\ f_{ij}^0(x) - z_{ij}^0 = 0, \;\; i = 1, ..., n_i \end{array} . \tag{47}$$

It is now shown that both of these sets may be approximated by equivalent RCP sets. Consider the first constraint type

$$\prod_{i=1}^{n_i} z_{ij}^0 - s_j^0 \leq 0 \tag{48}$$

and note that continuous additions of auxiliary variables u and the substitution:

$$z_1 z_2 - u = 0 \tag{49}$$

allows for (48) to be reduced to the constraint set defined by one constraint of the type:

$$u_a u_b - s_j^0 \leq 0, \tag{50}$$

and some finite number of constraints of type (49).

The constraint type (50) is approximated by an RCP set as follows:

$$\begin{array}{l} u_a u_b - s_j^0 \leq 0 \Leftrightarrow 0.5(u_a + u_b)^2 - 0.5u_a^2 - 0.5u_b^2 - s_j^0 \leq 0 \\ \Leftrightarrow 0.5u_c^2 - 0.5u_a^2 - 0.5u_b^2 - s_j^0 \leq 0, \; u_c = u_a + u_b \\ \approx 0.5p(u_c) - 0.5u_a^2 - 0.5u_b^2 - s_j^0 \leq 0, \; u_c = u_a + u_b \\ \Leftrightarrow 0.5p_i(u_c) - 0.5u_a^2 - 0.5u_b^2 - s_j^0 \leq 0 \;\; (i = 1, ..., n_p), \; u_c = u_a + u_b \end{array} , \tag{51}$$

with $p(u_c) \approx u_c^2$ guaranteed to exist as u_c^2 is Lipschitz-continuous on a bounded domain of u_c (boundedness of $f(x)$ implying boundedness of $f_{ij}^0(x)$ implying boundedness of z_{ij}^0 implying boundedness of u_a and u_b, and thus u_c). (49) is approximated by first breaking it into $z_1 z_2 - u \leq 0$ and $-z_1 z_2 + u \leq 0$, and then applying the procedure of (51) to both.

The second constraint type of (47) may be approximated by an RCP set by first breaking it into inequalities:

$$f_{ij}^0(x) - z_{ij}^0 = 0 \Leftrightarrow f_{ij}^0(x) - z_{ij}^0 \leq 0, \;\; -f_{ij}^0(x) + z_{ij}^0 \leq 0, \tag{52}$$

which, by factorability, may be written as:

$$\sum_j \prod_i f_{ij}^1(x) - z_{ij}^0 \leq 0, \;\; -\sum_j \prod_i f_{ij}^1(x) + z_{ij}^0 \leq 0. \tag{53}$$

This establishes a cyclic procedure, since both of these constraints may be treated by the steps of (47)-(53) to yield more RCP approximations together with the constraints of (53) where the factoring level has been augmented by one. This can be continued until the factoring level reaches m and, lumping the z variable into the summation of products, one is left with an RCP set and the constraints:

$$\sum_j \prod_i f_{ij}^m((a_{ij}^F)^T x + b_{ij}^F) \leq 0, \;\; -\sum_j \prod_i f_{ij}^m((a_{ij}^F)^T x + b_{ij}^F) \leq 0. \tag{54}$$

The summations may be "removed" by the same procedure as in (46) and, introducing a final set of auxiliary z variables, one has:

$$\prod_i z_{ij}^m \le 0,\ -\prod_i z_{ij}^m \le 0,\ f_{ij}^m((a_{ij}^F)^T x + b_{ij}^F) - z_{ij}^m = 0, \tag{55}$$

where the first two sets may be put through the procedure of (48)-(51) to yield RCP approximations. With the addition of auxiliary variables $x_{ij} = (a_{ij}^F)^T x + b_{ij}^F$, the last constraint type can now be expressed and reformulated as:

$$\begin{array}{l} f_{ij}^m(x_{ij}) - z_{ij}^m = 0 \Leftrightarrow f_{ij}^m(x_{ij}) - z_{ij}^m \le 0,\ -f_{ij}^m(x_{ij}) + z_{ij}^m \le 0 \\ \approx p^+(x_{ij}) - z_{ij}^m \le 0,\ p^-(x_{ij}) + z_{ij}^m \le 0 \\ \Leftrightarrow p_k^+(x_{ij}) - z_{ij}^m \le 0\ (k = 1,...,n_p^+),\ p_k^-(x_{ij}) + z_{ij}^m \le 0\ (k = 1,...,n_p^-) \end{array}, \tag{56}$$

which is an RCP set.

Since all of the RCP approximations used in steps (51) and (56) may be arbitrarily precise, it follows that the overall approximation of $f(x) \le 0$, which includes a finite number of such approximations, may be arbitrarily precise as well. □

Although somewhat busy, it is worth noting that the proof of Theorem 5 is *nearly* algorithmic in nature, as the steps could, in principle, be automated. The one part that may prove difficult for automation is the factoring, as this may require either symbolic logic or additional input from the user.

The following corollary completes the link between Problems (1) and (2).

Corollary 1 (Approximation of NLP Problem (2) by RCP Problem (1))
Let $f_0(x)$, $g_0(x)$, $h_0(x)$, and $-h_0(x)$ be concave-factorable, bounded over X, and their base component functions Lipschitz-continuous in the relevant affine combinations over X. It follows that Problem (2) may be approximated, to arbitrary precision, by Problem (1).

Proof The addition of an auxiliary variable s, the epigraph transformation, and the splitting of equality constraints allows the following equivalence:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f_0(x) \\ \text{subject to} & g_{0,i}(x) \le 0, \quad i = 1,...,n_{g0} \\ & h_{0,i}(x) = 0, \quad i = 1,...,n_{h0} \end{array} \Leftrightarrow \begin{array}{ll} \underset{x,s}{\text{minimize}} & s \\ \text{subject to} & f_0(x) - s \le 0 \\ & g_{0,i}(x) \le 0, \quad i = 1,...,n_{g0} \\ & h_{0,i}(x) \le 0, \quad i = 1,...,n_{h0} \\ & -h_{0,i}(x) \le 0, \quad i = 1,...,n_{h0} \end{array}. \tag{57}$$

Clearly, $f_0(x) - s$ is concave-factorable if $f_0(x)$ is concave-factorable. It then follows from Theorem 5 that the feasible set of the right-hand side of (57) may be approximated arbitrarily well by an RCP set. As the cost of the latter is also linear, equivalence to Problem (1) is established. □

Some examples to illustrate the NLP → RCP approximation are now given.

Example 3 (NLP → RCP Approximation)

Consider the following problem of Al-Khayyal and Falk [1]:

$$\begin{array}{ll} \underset{x_1,x_2}{\text{minimize}} & -x_1 + x_1x_2 - x_2 \\ \text{subject to} & -6x_1 + 8x_2 \le 3 \\ & 3x_1 - x_2 \le 3 \\ & x_1, x_2 \in [0,5] \end{array}, \tag{58}$$

where the constraints are already in RCP form and only the cost needs reformulation.

Applying the epigraph to the nonlinear portion of the cost and using the equality $x_1x_2 = 0.5(x_1+x_2)^2 - 0.5x_1^2 - 0.5x_2^2$ yields the equivalent problem:

$$\begin{array}{ll} \underset{x_1,x_2,x_3}{\text{minimize}} & -x_1 - x_2 + 0.5x_3 \\ \text{subject to} & (x_1+x_2)^2 - x_1^2 - x_2^2 - x_3 \leq 0 \\ & -6x_1 + 8x_2 \leq 3 \\ & 3x_1 - x_2 \leq 3 \\ & x_1, x_2 \in [0,5], x_3 \in [0,50] \end{array} . \tag{59}$$

Then, introducing the auxiliary x_4:

$$\begin{array}{ll} \underset{x_1,x_2,x_3,x_4}{\text{minimize}} & -x_1 - x_2 + 0.5x_3 \\ \text{subject to} & x_1 + x_2 - x_4 = 0 \\ & x_4^2 - x_1^2 - x_2^2 - x_3 \leq 0 \\ & -6x_1 + 8x_2 \leq 3 \\ & 3x_1 - x_2 \leq 3 \\ & x_1, x_2 \in [0,5], x_3 \in [0,50], x_4 \in [0,10] \end{array} . \tag{60}$$

Finally, one may define $p(x_4) \approx x_4^2$ and break the piecewise-maximum to obtain:

$$\begin{array}{ll} \underset{x_1,x_2,x_3,x_4}{\text{minimize}} & -x_1 - x_2 + 0.5x_3 \\ \text{subject to} & x_1 + x_2 - x_4 = 0 \\ & p_i(x_4) - x_1^2 - x_2^2 - x_3 \leq 0, \;\; i = 1,...,n_p \\ & -6x_1 + 8x_2 \leq 3 \\ & 3x_1 - x_2 \leq 3 \\ & x_1, x_2 \in [0,5], x_3 \in [\underline{\varepsilon}_a, 50 + \overline{\varepsilon}_a], x_4 \in [0,10] \end{array} , \tag{61}$$

which is easily rearranged into standard RCP form.

The box constraints $X = \{x : \underline{x} \leq x \leq \overline{x}\}$ deserve a few words. As will become apparent in Sections 4 and 5, domain reduction techniques represent a crucial aspect of the proposed method and as such it is important that lower and upper bounds on all variables, including the auxiliary ones, be provided[2]. In many cases, as in this example, it is easy to calculate these bounds (e.g. one simply minimizes or maximizes $x_4^2 - x_1^2 - x_2^2$ over X to obtain the bounds on x_3, since x_3 must equal this quantity at the optimum). One should, however, keep in mind that approximation error may affect these bounds, i.e. when x_4^2 is replaced by $p(x_4)$, it is important to add appropriate slacks for the lower and upper bounds on the approximation error ($\underline{\varepsilon}_a$ and $\overline{\varepsilon}_a$, respectively). A practical observation, however, is that gross under- and over-estimates will often suffice, as the domain reduction techniques discussed in Section 4 are often able to refine these.

Example 4 (NLP → RCP Approximation)

Consider the following problem:

$$\begin{array}{ll} \underset{x_1,x_2}{\text{minimize}} & (\sin x_1)(-x_1 + 0.3x_2) \\ \text{subject to} & x_1 \in [-2,2], x_2 \in [-5,5] \end{array} . \tag{62}$$

Noting that the cost is already written as a product of base component functions, one may define the auxiliary variables $x_3 = \sin x_1$ and $x_4 = -x_1 + 0.3x_2$, with the implicit bounds $x_3 \in [-1,1]$ and $x_4 \in [-3.5, 3.5]$:

[2] Note that doing so also ensures the compactness of the feasible space.

$$\begin{array}{ll} \underset{x_1,x_2,x_3,x_4}{\text{minimize}} & x_3x_4 \\ \text{subject to} & -x_1+0.3x_2-x_4=0 \\ & x_3=\sin x_1 \\ & x_1\in[-2,2], x_2\in[-5,5], x_3\in[-1,1], x_4\in[-3.5,3.5] \end{array} . \quad (63)$$

Breaking the equality constraint and rearranging leads to $\sin x_1 \le x_3$ and $-\sin x_1 \le -x_3$. Using the piecewise-concave approximations $p_a(x_1) \approx \sin x_1$ and $p_b(x_1) \approx -\sin x_1$, and then splitting the components into individual constraints yields:

$$\begin{array}{ll} \underset{x_1,x_2,x_3,x_4}{\text{minimize}} & x_3x_4 \\ \text{subject to} & -x_1+0.3x_2-x_4=0 \\ & p_{a,i}(x_1)-x_3\le 0,\ \ i=1,...,n_a \\ & p_{b,i}(x_1)+x_3\le 0,\ \ i=1,...,n_b \\ & x_1\in[-2,2], x_2\in[-5,5], x_3\in[-1+\underline{\varepsilon}_{a,1}, 1+\overline{\varepsilon}_{a,1}], x_4\in[-3.5,3.5] \end{array} . \quad (64)$$

The only remaining non-concave part is the objective, which may again be transformed by the equality $x_3x_4 = 0.5(x_3+x_4)^2 - 0.5x_3^2 - 0.5x_4^2$ and the use of the epigraph:

$$\begin{array}{ll} \underset{x_1,x_2,x_3,x_4,x_5}{\text{minimize}} & 0.5x_5 \\ \text{subject to} & -x_1+0.3x_2-x_4=0 \\ & p_{a,i}(x_1)-x_3\le 0,\ \ i=1,...,n_a \\ & p_{b,i}(x_1)+x_3\le 0,\ \ i=1,...,n_b \\ & (x_3+x_4)^2-x_3^2-x_4^2-x_5\le 0 \\ & x_1\in[-2,2], x_2\in[-5,5], x_3\in[-1+\underline{\varepsilon}_{a,1}, 1+\overline{\varepsilon}_{a,1}] \\ & x_4\in[-3.5,3.5], x_5\in[-7,7] \end{array} . \quad (65)$$

Following the same steps as in Example 3 then gives:

$$\begin{array}{ll} \underset{x_1,x_2,x_3,x_4,x_5,x_6}{\text{minimize}} & 0.5x_5 \\ \text{subject to} & -x_1+0.3x_2-x_4=0 \\ & x_3+x_4-x_6=0 \\ & p_{a,i}(x_1)-x_3\le 0,\ \ i=1,...,n_a \\ & p_{b,i}(x_1)+x_3\le 0,\ \ i=1,...,n_b \\ & p_{c,i}(x_6)-x_3^2-x_4^2-x_5\le 0,\ \ i=1,...,n_c \\ & x_1\in[-2,2], x_2\in[-5,5], x_3\in[-1+\underline{\varepsilon}_{a,1}, 1+\overline{\varepsilon}_{a,1}], x_4\in[-3.5,3.5] \\ & x_5\in[-7+\underline{\varepsilon}_{a,2}, 7+\overline{\varepsilon}_{a,2}], x_6\in[-4.5+\underline{\varepsilon}_{a,1}, 4.5+\overline{\varepsilon}_{a,1}] \end{array} . \quad (66)$$

4 An Extended RCP Solution Method

The basic prototype of an extended RCP solution method, comprised of the problem set-up and an active-set algorithm, is given here in three parts. The active-set-search scheme is outlined first, as this constitutes the base of the method and guarantees its finite-time convergence to a global minimum of (1). The second part then goes through a number of techniques that may dramatically speed up the algorithm by reducing the number of active sets to be checked. These may be divided into (a) techniques particular to (extended) RCP and (b) domain reduction techniques that are common to existing global optimization solvers. Finally, the third part joins these ideas to provide the full active-set RCP algorithm as it was coded in this work.

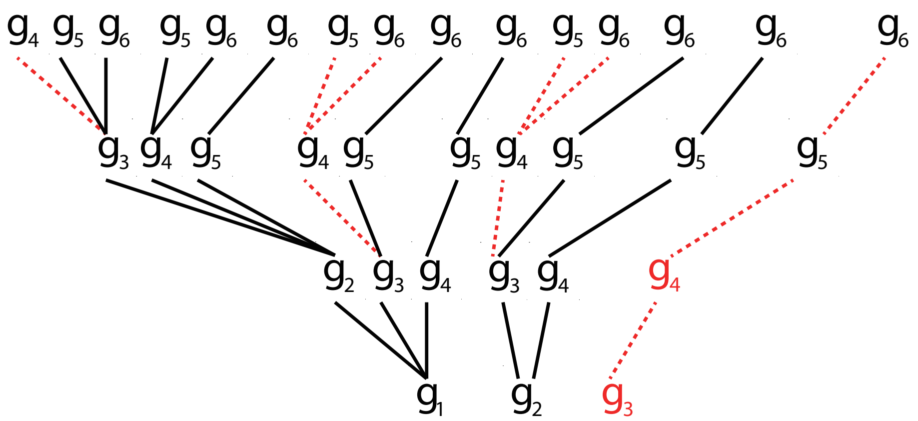

Fig. 4 The branching scheme of possible active sets for a problem with $n-n_C=4$ and $n_g=6$. The effect of fathoming an active subset ($g_3(x)$ and $g_4(x)$) is demonstrated, with all the resulting fathomed active sets shown via dotted branches.

4.1 Active-Set Search: General Procedure

As discussed at the end of Section 2, the global minimum of (1), provided that the problem is regular, may be found by simply checking all of the possible $\binom{n_g}{n-n_C}$ active sets, solving the corresponding convex problems (22), and then comparing the solutions. However, much like the brute vertex enumeration techniques of concave minimization [22,27], such an approach, though guaranteed to solve the problem in finite time, is not computationally enviable when $\binom{n_g}{n-n_C}$ is large. The work by Ueing [39] has proposed considering active *subsets* as a means of bypassing this difficulty, and this is essentially the approach pursued here.

Envisioning the entire set of possible active sets as a tree (Fig. 4), the basic idea of checking active subsets involves starting at the base level of the tree, with a single member (a single inequality constraint), and building up to the final level of $n-n_C$ inequality constraints. The justification for doing this is formalized in the following lemma.

Lemma 2 (Fathoming of Active Subsets)
Let $\tilde{G}_a(x)$ denote a set of $\tilde{n}<n-n_C$ inequality constraints of Problem (1), assumed regular, and consider the following problem:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & c^T x \\ \text{subject to} & -\tilde{G}_a(x) \leq \mathbf{0} \\ & x \in \mathscr{C} \end{array} , \qquad (67)$$

where $\mathscr{C}$ is defined as any convex set that contains x^* (nominally, the convex relaxation of the feasible set of (1), together with the linear equality constraints). It follows that:

(i) if (67) is infeasible, then the $\tilde{n}$ constraints defining $\tilde{G}_a(x)$ cannot be a subset of the active set defining x^*.
(ii) if (67) is feasible, the optimal value $c^T\tilde{x}^*$ defines a lower bound on the potential global minimal value attained for the active set $G_a(x)\supset\tilde{G}_a(x)$.
(iii) denoting by $\bar{i}$ the inequality constraints that are active or violated at $\tilde{x}^*$, with $\bar{i}=\{i: g_i(\tilde{x}^*)\geq 0\}$, Problem (67) will be feasible for all $\tilde{G}_a(x)$ that are defined by subsets of $\bar{i}$.

Proof (i) Let $G_a(x)$ denote the full active set of $n - n_C$ inequality constraints defining the global minimum x^*. By contradiction, suppose that $\tilde{G}_a(x) \subset G_a(x)$ despite Problem (67) being infeasible. $G_a(x^*) = \mathbf{0}$ implies $\tilde{G}_a(x^*) = -\tilde{G}_a(x^*) = \mathbf{0}$. By definition, $x^* \in \mathscr{C}$. However, this contradicts the infeasibility of (67), since x^* is clearly feasible for this problem.

(ii) Since $x^* \in \mathscr{C}$ and solves (22), the following equivalence must hold:

$$x^* = \begin{array}{ll} \arg\underset{x}{\text{maximize}} & c^T x \\ \text{subject to} & -G_a(x) \leq \mathbf{0} \\ & Cx = d \end{array} = \begin{array}{ll} \arg\underset{x}{\text{maximize}} & c^T x \\ \text{subject to} & -G_a(x) \leq \mathbf{0} \\ & Cx = d \\ & x \in \mathscr{C} \end{array}. \tag{68}$$

Consider now the reverse:

$$\underline{x}^* = \begin{array}{ll} \arg\underset{x}{\text{minimize}} & c^T x \\ \text{subject to} & -G_a(x) \leq \mathbf{0} \\ & Cx = d \\ & x \in \mathscr{C} \end{array}, \tag{69}$$

where it is obvious that $c^T \underline{x}^* \leq c^T x^*$. It follows that $c^T \tilde{x}^* \leq c^T \underline{x}^*$ since (67) is the same as (69) but with some constraints removed.

(iii) Since $\tilde{x}^* \in \mathscr{C}$ and satisfies $-\tilde{G}_a(\tilde{x}^*) \leq \mathbf{0}$ for all constraint sets $\tilde{G}_a(x)$ that are subsets of $\bar{i}$, it follows that $\tilde{x}^*$ will be a feasible point for all problems (67) where $\tilde{G}_a(x)$ is defined by subsets of $\bar{i}$. □

One sees that solving Problem (67) is very useful as it essentially does triple duty by:

(a) fathoming subsets that cannot define the active set for the global minimum (when the problem is infeasible), thereby saving the computational effort of checking the active sets that include the fathomed combinations as subsets,
(b) providing a lower bound on the globally minimal cost value attained by certain active sets, thereby allowing for those sets to be fathomed if a feasible point with a lower cost value is found,
(c) allowing for computational savings by foregoing Problem (67) when it is known to be feasible.

Of these three points, (a) is the most crucial as it allows for entire branches to be removed from the active-set search (see Fig. 4 for an illustration, where fathoming a single subset invalidates 6 of the 15 possible active sets).

On an implementation level, the active-set search proceeds by considering all the subsets at $\tilde{n} = 1$ and solving (67) for these subsets. This is repeated for increased $\tilde{n}$ until $\tilde{n}$ reaches $n - n_C$, at which point the reverse problems (22) are solved to obtain the global minimum candidates. Once more using the problem in Fig. 4 as an example, the active-set search in this case would proceed by solving (67), if needed, for the following subsets: $\{g_1\}$, $\{g_2\}$, $\{g_3\}$, $\{g_1,g_2\}$, $\{g_1,g_3\}$, $\{g_1,g_4\}$, $\{g_2,g_3\}$, $\{g_2,g_4\}$, $\{g_3,g_4\}$, $\{g_1,g_2,g_3\}$, $\{g_1,g_2,g_4\}$, $\{g_1,g_2,g_5\}$, $\{g_1,g_3,g_4\}$, $\{g_1,g_3,g_5\}$, $\{g_1,g_4,g_5\}$, $\{g_2,g_3,g_4\}$, $\{g_2,g_3,g_5\}$, $\{g_2,g_4,g_5\}$, $\{g_3,g_4,g_5\}$, before proceeding to solve (22) for the full active sets.

Throughout this procedure, a *fathoming* and a *validation* basis are built to cut out certain branches entirely and to skip solving (67) for certain others, respectively. Deferring the actual algorithmic details to Section 4.3, it should be noted that the worst-case complexity of such a scheme is actually worse than the $\binom{n_g}{n-n_C}$ convex problems of a brute enumeration.

However, its observed performance is unequivocally better (often orders of magnitude so) than the brute approach.

4.2 Fathoming and Domain Reduction Techniques

A natural property of the active-set search is that the fathoming of many low-cardinality subsets is bound to lead to the solution much quicker, since low-cardinality subsets will cut out greater parts of the search tree. A more general statement is that more fathoming will automatically lead to faster convergence, and so a number of fathoming techniques are now considered in detail. Some of these, as will be seen, take direct advantage of the concavity, the active-set nature, and the approximation steps that characterize the extended RCP framework. Others, however, employ standard domain reduction techniques common to modern global optimization solvers [25] so as to shrink $\mathscr{C}$ as much as possible, thereby increasing the number of subsets for which (67) is infeasible.

4.2.1 Fathoming Techniques Particular to Extended RCP

Innate Fathoming

Once the RCP problem has been defined, there are certain constraints or constraint combinations that are innately known to either never intersect or to not define the global minimum. These are stated in the following lemma.

Lemma 3 (Innate Fathoming Rules)

The following inequality constraint subsets are not subsets of the active set defining x^* and may be fathomed:

(i) *Any set or subset of $n - n_C$ constraints or less that is linearly dependent at x^*.*

(ii) *Lower and upper bound constraints, $\underline{s} \le s$ and $s \le \overline{s}$, on auxiliary variables added through the epigraph transformation:*

$$f_1(x) + f_2(x) \le 0 \Leftrightarrow \begin{array}{l} f_1(x) + s \le 0 \\ f_2(x) - s \le 0 \,, \\ \underline{s} \le s \le \overline{s} \end{array} \tag{70}$$

with $\underline{s} \le \inf_{x \in X} f_2(x)$ and $\overline{s} \ge \sup_{x \in X} f_2(x)$.

(iii) *Any pair of constraints $p_i(x_1) + f(x) \le 0$ and $p_j(x_1) + f(x) \le 0$, with $i \ne j$ denoting different pieces of the piecewise-concave function $p(x_1) = \max_{k=1,\ldots,n_p} p_k(x_1)$, assumed (without loss of generality) to be univariate in x_1, where $p_i(x_1) = p_j(x_1) \Rightarrow p_k(x_1) + f(x) > 0$ for some $k = 1,\ldots,n_p$ or $p_i(x_1) = p_j(x_1) \Rightarrow x_1 \notin [\underline{x}_1, \overline{x}_1]$.*

(iv) *Any pair of constraints $p_i^+(x) + f_1(x) \le 0$ and $p_j^-(x) - f_1(x) \le 0$, where $\max_{i=1,\ldots,n_p^+} p_i^+(x) < -f_2(x)$ and $\max_{j=1,\ldots,n_p^-} p_j^-(x) < f_2(x)$.*

Proof (i) This follows directly from Theorem 2.

(ii) It is first proven that the constraints $f_1(x) + s \le 0$ and $f_2(x) - s \le 0$ must be active simultaneously at the global minimum (x^*, s^*). Suppose the alternatives where at least one of the two is inactive and consider the implication from summing the two inequalities:

$$\begin{array}{c} f_1(x^*) + s^* = 0 \\ f_2(x^*) - s^* < 0 \end{array} \text{ or } \begin{array}{c} f_1(x^*) + s^* < 0 \\ f_2(x^*) - s^* = 0 \end{array} \text{ or } \begin{array}{c} f_1(x^*) + s^* < 0 \\ f_2(x^*) - s^* < 0 \end{array} \Rightarrow f_1(x^*) + f_2(x^*) < 0, \quad (71)$$

i.e. that the original constraint is strictly inactive at x^* and may therefore be dropped from consideration. As such, only the case where $f_1(x^*) + s^* = 0$ and $f_2(x^*) - s^* = 0$ is of interest. Since this fixes the value of s as $s^* = f_2(x^*)$, the box constraints $\underline{s} \leq s \leq \overline{s}$ can only influence this value if they do not admit $s^* = f_2(x^*)$ as feasible. This is, however, ruled out by the definition of $\underline{s}$ and $\overline{s}$. Since leaving these box constraints out of the RCP problem does not affect the solution, they may be fathomed from consideration.

(iii) Both constraints being active at x^* implies $p_i(x_1^*) = p_j(x_1^*)$, which in turn either implies $p_k(x_1^*) + f(x^*) > 0$ or $x_1^* \notin [\underline{x}_1, \overline{x}_1]$, both of which contradict the feasibility of x^*.

(iv) Suppose, by contradiction, that both $p_i^+(x) + f_1(x) \leq 0$ and $p_j^-(x) - f_1(x) \leq 0$ are active at the global minimum:

$$\begin{array}{c} p_i^+(x^*) + f_1(x^*) = 0 \\ p_j^-(x^*) - f_1(x^*) = 0 \end{array} \Rightarrow p_i^+(x^*) + p_j^-(x^*) = 0. \quad (72)$$

From $\max\limits_{i=1,\ldots,n_p^+} p_i^+(x) < -f_2(x)$ and $\max\limits_{j=1,\ldots,n_p^-} p_j^-(x) < f_2(x)$, it follows that:

$$\begin{array}{c} p_i^+(x^*) < -f_2(x^*) \\ p_j^-(x^*) < f_2(x^*) \end{array} \Rightarrow p_i^+(x^*) + p_j^-(x^*) < 0, \quad (73)$$

which contradicts (72). □

In extended RCP language, the results of Lemma 3 mean the following:

(i) any combination of $n - n_C$ or less linear constraints that are linearly dependent may be fathomed – this is particularly relevant for box constraint pairs (e.g. $\underline{x}_1 \leq x_1$ and $x_1 \leq \overline{x}_1$),
(ii) box constraints for introduced auxiliary variables may be fathomed,
(iii) all pairs of constraints corresponding to non-adjacent pieces in a piecewise-concave approximation may be fathomed (see Fig. 5 for an illustration),
(iv) when a univariate nonlinear equality constraint (e.g. $x_3 = \sin x_1$ in (63)) is split with both parts strictly under-approximated, the pairs coming from different approximations cannot define together the optimal active set and may be fathomed (this is equivalent to saying that a strict over-approximation of a function cannot intersect a strict under-approximation of the same function).

Fathoming Box Constraint Combinations

All points corresponding to a given active subset of box constraints may be proven infeasible (and the constraint combination thereby fathomed) via a cheap computational procedure when $\mathscr{C}$ is a polytope:

$$\mathscr{C} = \{x : (a_i^{\mathscr{C}})^T x \leq b_i^{\mathscr{C}},\ i = 1, \ldots, n_{\mathscr{C}}\}, \quad (74)$$

which, for the methods proposed in this work, will always be the case.

This procedure is possible since choosing a subset of box constraints fixes the corresponding variables and allows for a simple minimization of any linear constraint over the rest. Without loss of generality, let $x_1, \ldots, x_{\tilde{n}}$ denote the variables whose box constraints have

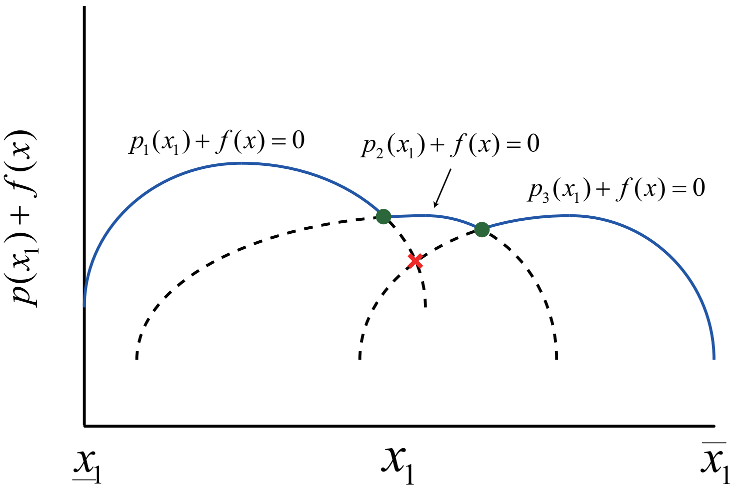

Fig. 5 Illustration of how a crossing between non-adjacent concave constraints of a piecewise-concave function is bound to lie in the infeasible region. Here, the cross designates the intersection between $p_1(x_1) + f(x) = 0$ and $p_3(x_1) + f(x) = 0$ (seen to be infeasible), while the two round points indicate the feasible intersections of adjacent constraints $p_1(x_1) + f(x) = p_2(x_1) + f(x)$ and $p_2(x_1) + f(x) = p_3(x_1) + f(x)$.

been fixed and $x_{\tilde{n}+1}, ..., x_n$ the others. If, for example, the constraint set in question consists of the lower bounds on $x_1, ..., x_{\tilde{n}}$, then the minimal value of each $\mathscr{C}$ constraint value over X (and thus over $\mathscr{C}$) may be calculated as:

$$\min_{x_{\tilde{n}+1},...,x_n \in X} (a_i^{\mathscr{C}})^T x = \sum_{j=1}^{\tilde{n}} a_{ij}^{\mathscr{C}} \underline{x}_j + \sum_{j=\tilde{n}+1}^{n} \min \left(a_{ij}^{\mathscr{C}} \underline{x}_j, a_{ij}^{\mathscr{C}} \overline{x}_j \right). \tag{75}$$

It then follows that if this value is strictly superior to $b_i^{\mathscr{C}}$ for any $i = 1, ..., n_{\mathscr{C}}$, then the constraint set corresponding to the lower bounds on $x_1, ..., x_{\tilde{n}}$ may be fathomed since there cannot exist any point in $\mathscr{C}$ for which these constraints would be active. Note that the equality constraints $Cx = d$ may be easily incorporated by simply being split into $Cx \leq d$ and $-Cx \leq -d$.

While this method can provide useful fathoming information at a very low price, it may nevertheless be computationally expensive to run this check for all possible active sets and subsets generated by the box constraints. The following scheme is proposed to check the different sets in a branching manner that is somewhat similar in nature to the overall active-set search, but which is set to terminate if the number of nodes grows too large.

Subroutine A (Fathoming Box Constraints)

User input: $\mathscr{C}$, X, and F. F is the fathoming basis – a Boolean[3] matrix of n_g columns, with 0 denoting the absence of a constraint from the considered set and 1 its presence. The final $2n$ columns of F correspond, by the convention set here, to the box constraints.
Output: F (updated).

1. Set $B := \mathbf{0}_{1 \times 2n}$.

[3] Boolean vectors and matrices are used throughout the main algorithm and subroutines as this provides an easy way to check membership (e.g. of a Boolean vector in a Boolean matrix) through a series of multiplications and additions.

2. If $B = \varnothing$, terminate. Otherwise, go to Step 3.
3. Define $\underline{B}_c^1$ as the first n elements of the first row of B and $\overline{B}_c^1$ as the last n elements of the first row of B. Remove the first row of B and define $B_c^1 = \underline{B}_c^1 + \overline{B}_c^1$. Let $\tilde{n}$ denote the index of the last non-zero element of B_c^1 (0 if there are none) augmented by 1. If $\tilde{n} > n$, return to Step 2. Otherwise, proceed to Step 4.
4. For $k := \tilde{n}, ..., n$:
   (a) Define $\underline{B}_c$ and $\overline{B}_c$ as the vectors $\underline{B}_c^1$ and $\overline{B}_c^1$ with their k^{th} indices set to 1.
   (b) If

$$\exists i \in 1,...,n_{\mathscr{C}} : \sum_{j:\underline{B}_{c,j}=1} a_{ij}^{\mathscr{C}} \underline{x}_j + \sum_{j:\overline{B}_{c,j}^1=1} a_{ij}^{\mathscr{C}} \overline{x}_j + \sum_{j:\underline{B}_{c,j},\overline{B}_{c,j}^1=0} \min\left(a_{ij}^{\mathscr{C}} \underline{x}_j, a_{ij}^{\mathscr{C}} \overline{x}_j\right) > b_i^{\mathscr{C}} \tag{76}$$

   and $\nexists z : F_z \in [\mathbf{0}_{1\times(n_g-2n)}\ \underline{B}_c\ \overline{B}_c^1]$, then append $[\mathbf{0}_{1\times(n_g-2n)}\ \underline{B}_c\ \overline{B}_c^1]$ to F[4]. If neither is true, then append $[\underline{B}_c\ \overline{B}_c^1]$ to B. Likewise, if

$$\exists i \in 1,...,n_{\mathscr{C}} : \sum_{j:\underline{B}_{c,j}^1=1} a_{ij}^{\mathscr{C}} \underline{x}_j + \sum_{j:\overline{B}_{c,j}=1} a_{ij}^{\mathscr{C}} \overline{x}_j + \sum_{j:\underline{B}_{c,j}^1,\overline{B}_{c,j}=0} \min\left(a_{ij}^{\mathscr{C}} \underline{x}_j, a_{ij}^{\mathscr{C}} \overline{x}_j\right) > b_i^{\mathscr{C}} \tag{77}$$

   and $\nexists z : F_z \in [\mathbf{0}_{1\times(n_g-2n)}\ \underline{B}_c^1\ \overline{B}_c]$, then append $[\mathbf{0}_{1\times(n_g-2n)}\ \underline{B}_c^1\ \overline{B}_c]$ to F. If neither is true, then append $[\underline{B}_c^1\ \overline{B}_c]$ to B.
5. If the number of rows in B is superior to 100, terminate. Otherwise, return to Step 2.

Here, the choice of 100 in the final step of the routine is heuristic.

Fathoming Separable Concave Constraints

One may also prove the inactivity of a given separable concave constraint over X (and thereby $\mathscr{C}$) by calculating its maximum value on X and showing that it is strictly inferior to 0. This may be done in the general nonseparable case by solving a single convex optimization problem, and indeed this is what happens in (67) when $\tilde{G}_a(x)$ consists of a single member. However, a faster check may be performed for the separable case since, for a given $g_i(x)$:

$$\max_{x\in X} g_i(x) = \sum_{j=1}^{n} \max_{x_j\in[\underline{x}_j,\overline{x}_j]} g_{ij}(x_j), \tag{78}$$

where each of the separate components must reach their maximum at either (a) the lower boundary, (b) the upper boundary, or (c) the stationary zero-derivative point. As checking these cases n times is significantly cheaper than maximizing $g_i(x)$ over $\mathscr{C}$, (78) offers an easy way to quickly fathom $g_i(x)$ if it is irrelevant.

Subroutine B (Fathoming Separable Concave Constraints)

User input: X, $g_i(x)$ (with $i = 1,...,n_g - 2n$), F. By the convention set here, the first $n_g - 2n$ columns of F correspond to the concave constraints (box constraints excluded).
Output: F (updated).

1. For $i = \{1,...,n_g - 2n\} \setminus \{i : g_i(x) \text{ not separable}\}$:

[4] F_z denotes the z^{th} row of F.

(a) For $j = 1, ..., n$:
  (i) Calculate:

$$\begin{aligned} &\max_{x_j \in [\underline{x}_j, \overline{x}_j]} g_{ij}(x_j) = \max\left(g_{ij}(\underline{x}_j), g_{ij}(\overline{x}_j), g_{ij}(\hat{x}_j)\right) \\ &\hat{x}_j = \{\hat{x}_j : \left.\tfrac{dg_{ij}}{dx_j}\right|_{\hat{x}_j} = 0\} \text{ if } \underline{x}_j \leq \hat{x}_j \leq \overline{x}_j \ (g_{ij}(\hat{x}_j) = \varnothing \text{ otherwise}) \end{aligned} . \tag{79}$$

(b) If:

$$\sum_{j=1}^{n} \max_{x_j \in [\underline{x}_j, \overline{x}_j]} g_{ij}(x_j) < 0, \tag{80}$$

then append $[\mathbf{0}_{1\times(i-1)}\ 1\ \mathbf{0}_{1\times(n_g-2n-i)}\ \mathbf{0}_{1\times 2n}]$ to F if $\nexists z : F_z \in [\mathbf{0}_{1\times(i-1)}\ 1\ \mathbf{0}_{1\times(n_g-2n-i)}\ \mathbf{0}_{1\times 2n}]$.

Mandatory Constraints and Linked Constraint Sets

A characteristic of the NLP $\rightarrow$ RCP transformation is the addition of auxiliary variables via the epigraph transformation. This naturally leads to inequality constraints that are linked by the auxiliary variables and which may be shown to be necessarily active together if one of them is active at the solution. Additionally, applying the epigraph to any portion of the cost function, or recognizing innate epigraph forms in a given problem, leads to "mandatory" constraints that must be active at the optimum.

For the former case, consider Point (ii) of Lemma 3 and its proof, which shows that "distributing" a constraint over two (or more) constraints in the epigraph transformation leads to a linked set – if the original constraint is relevant and active at the global minimum, the entire equivalent set of multiple inequality constraints must be as well. Although this information does not yield useful members for the fathoming basis F, it may nevertheless be exploited through conditional statements to fathom out those constraint combinations that preclude linked constraints being active together.

With regard to the latter point, consider the following transformation of the (bounded) problem:

$$\underset{x}{\text{minimize}} \quad \sum_{i=1}^{n_f} f_i(x) \Leftrightarrow \begin{array}{ll} \underset{x}{\text{minimize}} & \sum_{i=1}^{n_f} s_i \\ \text{subject to} & f_i(x) - s_i \leq 0, \ \ i = 1, ..., n_f \end{array} , \tag{81}$$

where it is clear that all of the resulting constraints must be active at a global (or local) minimum. Furthermore, transforming any of the resulting constraints via an additional epigraph implies that all of the constraints that result from the second transformation must be active as well. The impact of mandatory constraints is to reduce the number of levels in the active-set search. By the convention proposed here, mandatory constraints are given the first indices and are included in every subset and set considered.

4.2.2 General Domain Reduction Techniques

Domain reduction techniques [6] are standard in several of the currently available global optimization solvers [26], and have been credited for reducing the computational effort of a complete global search significantly [33,40]. This is no different in extended RCP, where the use of domain reduction techniques to shrink X (and thereby $\mathscr{C}$) can make solution times orders of magnitude faster. The particular characteristic of the RCP active-set search with respect to domain reduction techniques is that domain reduction allows for more subsets to be fathomed earlier in the search.

The general domain reduction routine, with a slight modification, is presented first.

Subroutine C (Domain Reduction)

User input: X, $\mathscr{C}$, and $X_\mathscr{C}$. $X_\mathscr{C} \in \mathbb{R}^{n\times 2n}$ is a matrix of $2n$ coordinates corresponding to the points where the n variables reach their lower and upper bounds on $\mathscr{C}$.
Output: X (updated), $\mathscr{C}$ (updated), $X_\mathscr{C}$ (updated).

1. If $X_\mathscr{C} \neq \varnothing$, skip to Step 3. Otherwise, for $i = 1,...,n$:
   (a) Calculate

$$\begin{aligned} \underline{X}_\mathscr{C} := \arg \underset{x}{\text{minimize}} \quad & x_i \\ \text{subject to} \quad & x \in \mathscr{C} \end{aligned} \tag{82}$$

   and update $\underline{x}_i := \underline{X}_{\mathscr{C},i}$. Likewise, calculate

$$\begin{aligned} \overline{X}_\mathscr{C} := \arg \underset{x}{\text{maximize}} \quad & x_i \\ \text{subject to} \quad & x \in \mathscr{C} \end{aligned} \tag{83}$$

   and update $\bar{x}_i := \overline{X}_{\mathscr{C},i}$. Append $[\underline{X}_\mathscr{C} \;\; \overline{X}_\mathscr{C}]$ to $X_\mathscr{C}$.
2. Use the updated X to update $\mathscr{C}$ accordingly.
3. For $i = 1,...,n$:
   (a) Define $\underline{X}_\mathscr{C}$ as the row vector corresponding to the first n columns of the i^{th} row of $X_\mathscr{C}$. Likewise, define $\overline{X}_\mathscr{C}$ as the row vector corresponding to the last n columns of the i^{th} row of $X_\mathscr{C}$.
   (b) If $\underline{X}_\mathscr{C} \in \mathscr{C}$, proceed to Step 3c. Otherwise, reorder the constraints of $\mathscr{C}$ so that:

$$(a_1^\mathscr{C})^T \underline{X}_\mathscr{C} - b_1^\mathscr{C} \geq (a_2^\mathscr{C})^T \underline{X}_\mathscr{C} - b_2^\mathscr{C} \geq ... \geq (a_{n_\mathscr{C}}^\mathscr{C})^T \underline{X}_\mathscr{C} - b_{n_\mathscr{C}}^\mathscr{C}. \tag{84}$$

      (i) Set $\tilde{n}_\mathscr{C} := n - n_C$.
      (ii) Construct:

$$A_\mathscr{C} := \begin{bmatrix} C \\ (a_1^\mathscr{C})^T \\ \vdots \\ (a_{\tilde{n}_\mathscr{C}}^\mathscr{C})^T \end{bmatrix}, \; b_\mathscr{C} := \begin{bmatrix} d \\ b_1^\mathscr{C} \\ \vdots \\ b_{\tilde{n}_\mathscr{C}}^\mathscr{C} \end{bmatrix}. \tag{85}$$

      (iii) If rank $A_\mathscr{C} < n$, set $\tilde{n}_\mathscr{C} := \tilde{n}_\mathscr{C} + 1$ and return to (ii). Otherwise, define $\underline{X}_\mathscr{C} := A_\mathscr{C}^\dagger b_\mathscr{C}$ († denoting the Moore-Penrose pseudoinverse).
      (iv) If $\underline{X}_\mathscr{C} \in \mathscr{C}$, check that $\underline{X}_\mathscr{C}$ is a KKT point of (82) by verifying that $\exists \lambda \in \mathbb{R}_+^{\tilde{n}_\mathscr{C}}, \mu \in \mathbb{R}^{n_C} : [\mathbf{0}_{1\times(i-1)} \;\; 1 \;\; \mathbf{0}_{1\times(n-i)}] + \sum_{i=1}^{\tilde{n}_\mathscr{C}} \lambda_i (a_i^\mathscr{C})^T + \sum_{i=1}^{n_C} \mu_i C_i = \mathbf{0}$. A cheap way to do this is by taking the pseudoinverse to solve for the Lagrange multipliers. If this cannot be verified, or if $\underline{X}_\mathscr{C} \notin \mathscr{C}$, redefine $\underline{X}_\mathscr{C}$ by solving (82). Update $\underline{x}_i := \underline{X}_{\mathscr{C},i}$.

   (c) If $\overline{X}_\mathscr{C} \in \mathscr{C}$, proceed to Step 3d. Otherwise, reorder the constraints of $\mathscr{C}$ so that:

$$(a_1^\mathscr{C})^T \overline{X}_\mathscr{C} - b_1^\mathscr{C} \geq (a_2^\mathscr{C})^T \overline{X}_\mathscr{C} - b_2^\mathscr{C} \geq ... \geq (a_{n_\mathscr{C}}^\mathscr{C})^T \overline{X}_\mathscr{C} - b_{n_\mathscr{C}}^\mathscr{C}. \tag{86}$$

      (i) Set $\tilde{n}_\mathscr{C} := n - n_C$.
      (ii) Construct $A_\mathscr{C}$ and $b_\mathscr{C}$ as in (85).

(iii) If $\operatorname{rank} A_{\mathscr{C}} < n$, set $\tilde{n}_{\mathscr{C}} := \tilde{n}_{\mathscr{C}} + 1$ and return to (ii). Otherwise, define $\overline{X}_{\mathscr{C}} := A_{\mathscr{C}}^{\dagger} b_{\mathscr{C}}$.

(iv) If $\overline{X}_{\mathscr{C}} \in \mathscr{C}$, check that $\overline{X}_{\mathscr{C}}$ is a KKT point of (83) by verifying that $\exists \lambda \in \mathbb{R}_{+}^{\tilde{n}_{\mathscr{C}}}, \mu \in \mathbb{R}^{n_C} : [\mathbf{0}_{1\times(i-1)} \ \ -1 \ \ \mathbf{0}_{1\times(n-i)}] + \sum_{i=1}^{\tilde{n}_{\mathscr{C}}} \lambda_i (a_i^{\mathscr{C}})^T + \sum_{i=1}^{n_C} \mu_i C_i = \mathbf{0}$. If this cannot be verified, or if $\overline{X}_{\mathscr{C}} \notin \mathscr{C}$, redefine $\overline{X}_{\mathscr{C}}$ by solving (83). Update $\overline{x}_i := \overline{X}_{\mathscr{C},i}$.

(d) Replace the i^{th} row of $X_{\mathscr{C}}$ by the updated $[\underline{X}_{\mathscr{C}} \ \ \overline{X}_{\mathscr{C}}]$.

4. Compare the previous X (prior to Step 3) with the new updated X element by element. If the maximum absolute difference between the old and new elements is less than the specified tolerance ε_X, terminate. Otherwise, return to Step 2.

The above subroutine essentially updates the box X by solving linear programs to calculate the minimal and maximal bounds on the individual variables, which it then uses to redefine and shrink $\mathscr{C}$, after which the bounds are recalculated. The aforementioned "slight modification" comes via storing the old solution points and using them whenever possible to avoid solving (82) and (83), either by (a) verifying that the old point is still inside $\mathscr{C}$ and thus does not need updating, or by (b) projecting the old answer on the most active constraints in hopes of this being the optimal active set that would solve (82) or (83). It should be mentioned that storing $X_{\mathscr{C}}$ may reduce the computational burden of (82) and (83) significantly as well, as it provides a warm start for what are already LP problems. Additional techniques may be possible to reduce the effort here as well (see, e.g., [40]), but have not been considered in this work.

Of crucial interest in Subroutine C is the "update $\mathscr{C}$ accordingly" in Step 2, which is pertinent since the definition of $\mathscr{C}$ will generally be dependent on the definition of X. Noting that $Cx = d$, any linear $g_i(x)$, and the box X may be incorporated into $\mathscr{C}$ directly, of which only the latter is affected by Subroutine C, the other elements defining $\mathscr{C}$ are now discussed in some detail.

Convex Underestimators

Denote by $l_i(x) \leq g_i(x), \forall x \in X$ a linear underestimator of the (nonlinear) concave constraint $g_i(x)$ over X. It is well-known that the efficiency of such underestimators will depend on the degree of nonlinearity of $g_i(x)$ as well as on the size of X. Incorporating $l_i(x)$ into $\mathscr{C}$ thereby allows for the iterative domain reduction as described in Subroutine C, as tightening X makes $l_i(x)$ more efficient (and thereby more restricting), which in turn allows for further tightening of X, and so on. Two particular cases, arguably the two most relevant in the extended RCP methodology, are examined here.

The first corresponds to the case where $g_i(x)$ is separable, as this allows to construct a convex (linear) underestimator of $g_i(x)$ by constructing the convex (linear) underestimators of its univariate components, $g_{ij}(x)$. As all of these components are concave, the convex underestimator of each $g_{ij}(x)$ is simply the line segment joining $(\underline{x}_j, g_{ij}(\underline{x}_j))$ and $(\overline{x}_j, g_{ij}(\overline{x}_j))$. The summation of these underestimators then gives the underestimator of $g_i(x)$, which is, with respect to X, its convex envelope [9].

The second case of interest is that of the univariate piecewise-concave function as generated by Algorithm 1. It is not difficult to show that the convex underestimator of such a function is simply the (piecewise-linear) convex underestimator of the intersection points of the adjacent pieces, together with the points corresponding to the lower and upper boundaries $\underline{x}$ and $\overline{x}$. An algorithm for constructing this underestimator is proposed below.

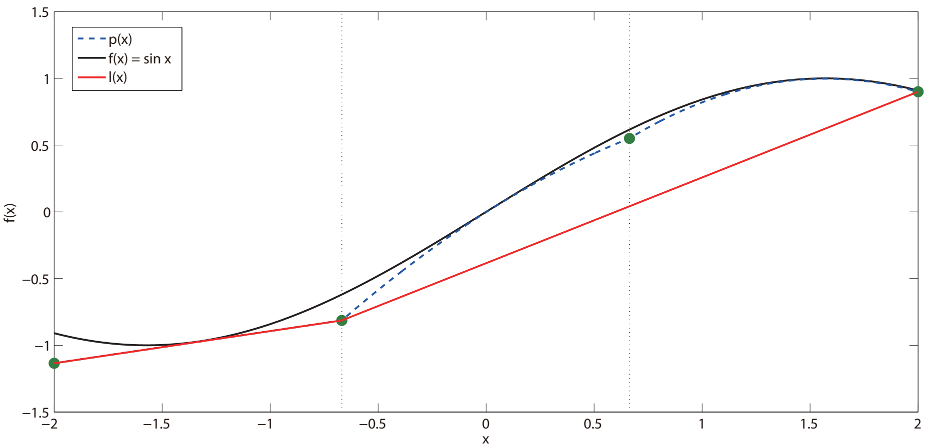

Fig. 6 An example of applying Algorithm 2 to construct a convex underestimator of the 3-piece piecewise-concave under-approximator of the function $f(x) = \sin x$, which is equivalent to the convex underestimator of the boundary and intersection points. The convex underestimator and piecewise-concave approximation are identical on the first interval.

Algorithm 2 (Convex Underestimator for a Univariate Piecewise-Concave Function)
User Input: $\underline{x}$, $\overline{x}$, $p(x)$, and x_c, where x_c may be taken as the first column of X^r in Algorithm 1 and corresponds to the coordinates of all of the feasible intersection points of the pieces of $p(x)$, sorted in increasing order.
Output: P, a two-column matrix giving the slope and y-intercept for the underestimating linear functions.

1. Set $P := \varnothing$. Remove elements of x_c that are either inferior to $\underline{x}$ or superior to $\overline{x}$. Append $\underline{x}$ to the beginning of x_c and $\overline{x}$ to the end. Set $k := 1$. Let $x_{c,k}$ denote the k^{th} element of x_c.
2. If $k = \|x_c\|_0$, terminate. Otherwise, proceed to Step 3.
3. For $j := k+1, ..., \|x_c\|_0$, define $m_{kj} := \frac{p(x_{c,j}) - p(x_{c,k})}{x_{c,j} - x_{c,k}}$.
4. Find $\hat{j} : m_{k\hat{j}} = \min_j m_{kj}$.
5. Calculate $\hat{b} = p(x_{c,k}) - m_{k\hat{j}} x_{c,k}$.
6. Append $[m_{k\hat{j}} \ \hat{b}]$ to P.
7. Set $k := \hat{j}$ and return to Step 2.

Fig. 6 shows the application of this algorithm to a piecewise-concave approximation of the function $f(x) = \sin x$. It should be clear that compressing X here (i.e. raising $\underline{x}$ or lowering $\overline{x}$) would improve the quality of the underestimator as well.

Local Minimization and Cutting Planes

Given some feasible point x_0, it is often reasonable to put in the computational effort for a local optimization so as to bring this point to a local minimum of the RCP problem, x^*_{loc}. The resulting point (in some cases already the global minimum) then gives an upper bound on the global minimum and allows for the cutting plane constraint $c^T x \leq c^T x^*_{loc}$ to be added to $\mathscr{C}$. In the author's experience, this is arguably the most important constraint with respect to the domain reduction, as finding a very good upper bound on the cost tends to lead to drastic reductions in X.

Other cutting plane constraints may be derived from multiplier techniques [25], which are based on the fact that the Lagrangian function must provide a lower bound on the cost function for any feasible point (x^*_{loc} included):

$$c^T x^*_{loc} + \sum_{i=1}^{n_g} \lambda_i g_i(x^*_{loc}) + \sum_{i=1}^{n_C} \mu_i (C_i x^*_{loc} - d_i) \leq c^T x^*_{loc}, \tag{87}$$

with $\lambda \in \mathbb{R}^{n_g}_+$ and $\mu \in \mathbb{R}^{n_C}$ the Lagrange multipliers for the inequality and equality constraints, respectively. Since the Lagrangian at the global minimum must also minimize the Lagrangian, it follows that the constraint

$$c^T x + \sum_{i=1}^{n_g} \lambda_i g_i(x) + \sum_{i=1}^{n_C} \mu_i (C_i x - d_i) \leq c^T x^*_{loc} \tag{88}$$

must be met for the global minimum. Since the constraints $g_i(x)$ are concave, their linear underestimators may be used instead in the definition of $\mathscr{C}$, thereby yielding the linear constraint:

$$c^T x + \sum_{i=1}^{n_g} \lambda_i l_i(x) + \sum_{i=1}^{n_C} \mu_i (C_i x - d_i) \leq c^T x^*_{loc}. \tag{89}$$

Like the linear approximations themselves, these constraints will too be refined as X is reduced.

The following subroutine is defined:

Subroutine D

User input: x_0 (optional), U, X, $\mathscr{C}$, $g_i(x)$, C, d, X^*_{loc}, and Λ^*, where X^*_{loc} and Λ^* are the matrices of coordinates and Lagrange multipliers for any previously found local optima, and U is an upper bound on the random samples used to find x_0 if it is not provided.
Output: $\mathscr{C}$ (updated), X^*_{loc} (updated), Λ^* (updated).

1. If x_0 is not given, randomly sample X until (a) a feasible x_0 is found, or (b) U samples have failed to find a feasible point. In the case of (a), proceed to Step 2. Otherwise, terminate.
2. Initialize a local solver at x_0 and solve (1) to local optimality to obtain x^*_{loc} and the corresponding multipliers λ^*, μ^*.
3. Letting $X^*_{loc,z} \in X^*_{loc}$ denote a row of X^*_{loc}, if $\nexists z : c^T X^*_{loc,z} \leq c^T x^*_{loc}$, replace (or introduce, if $X^*_{loc} = \varnothing$) the cost cutting plane in $\mathscr{C}$ with $c^T x \leq c^T x^*_{loc}$.
4. If $\nexists z : \|X^*_{loc,z} - x^*_{loc}\|_2 \leq \varepsilon_{loc}$, where ε_{loc} is some specified tolerance, add the constraint $c^T x + \sum_{i=1}^{n_g} \lambda^*_i l_i(x) + \sum_{i=1}^{n_C} \mu^*_i (C_i x - d_i) \leq c^T x^*_{loc}$ to $\mathscr{C}$ and append x^*_{loc} to X^*_{loc} and $[\lambda^* \ \mu^*]$ to Λ^*.

It is important to note that Subroutine D may not be very useful for hard feasibility problems where the feasible region is a small subset of X and where random sampling to find a feasible point in less than U samples may be close to impossible.

4.3 Detailed Outline of the Active-Set RCP Algorithm

Bringing together the ideas of the previous two subsections, the entire algorithm is now presented.

Algorithm 3 (Active-Set RCP Algorithm)
User input: c, $g_i(x)$, C, d, $\mathscr{C}$, X, U, and F, where F should be populated according to the innate fathoming rules described in Section 4.2.
Output: X^* (the set of solution candidates).

1. Initialize: $N := 0$ as the counter on the optimization problems solved (the dominant computational effort). $X^* := \varnothing$. $V := \varnothing$ and $X_V := \varnothing$ as the validation basis and the corresponding points for each member of the basis. $S := [\mathbf{1}_{1\times n_m}\ \mathbf{0}_{1\times(n_g-n_m)}]$ as the matrix of candidate active sets and subsets (initially a vector with only the n_m mandatory constraints accounted for, where the convention of listing the n_m mandatory constraints first is chosen). $S_{low} := 0$ as the vector of lower bounds corresponding to the constraint sets in S (initially a scalar with a dummy value of 0 that serves as a place holder). $X^*_{loc} := \varnothing, \Lambda^* := \varnothing, X_{\mathscr{C}} := \varnothing$.
2. Run Subroutine D ($N := N+1$) with no x_0 provided (unless a feasible point is somehow known). Proceed to Step 6 if Subroutine D fails to find a feasible point. Otherwise, denote the resulting local optimum by x_{up} and proceed to Step 3.
3. Run Subroutine C ($N := N+N_C$, where N_C is the number of times that Problems (82) or (83) are solved).
4. Run Subroutines A and B.
5. Solve the relaxed problem ($N := N+1$):
$$x_{low} := \arg \underset{x}{\text{minimize}} \quad c^T x \qquad \text{subject to} \quad x \in \mathscr{C} \quad . \tag{90}$$
If $\max_{i=1,...,n_g} g_i(x_{low}) \leq \varepsilon_g$, where ε_g is the largest acceptable constraint violation, then terminate and declare $X^* := x_{low}$. Alternatively, if $c^T x_{up} - c^T x_{low} < \varepsilon$, where ε is the largest acceptable suboptimality, terminate and declare $X^* := x_{up}$.
6. If $S = \varnothing$, terminate. Otherwise, define $\tilde{S}^1$ as the first row of S. Remove the first row of S and the first element of S_{low}. Denote by $\tilde{n}_g$ the index of the last non-zero element of $\tilde{S}^1$. If $\|\tilde{S}^1\|_1 < n - n_C - 1$, proceed to Step 7. Otherwise, proceed to Step 8.
7. Let $I_k := \{i : \tilde{n}_g + 1 \leq i \leq n_g\}$, ordered in increasing order, and define those indices of I_k for which individual constraints have been fathomed as $I_F := \{i : \exists z : F_z = [\mathbf{0}_{1\times(i-1)}\ 1\ \mathbf{0}_{1\times(n_g-i)}], i \geq \tilde{n}_g + 1\}$. Set $I_k := I_k \setminus I_F$. Remove the last $n - n_C - \|\tilde{S}^1\|_1 - 1$ elements of I_k (this avoids exploring those branches that terminate without being able to reach full cardinality, i.e. $n - n_C$ members). Set k equal to the first element of I_k:
   (a) Define the candidate subset, $\tilde{S}$, as $\tilde{S}^1$ with the k^{th} element set to 1.
   (b) If $\exists z : F_z \in \tilde{S}$, or if the set $\tilde{S}$ implies a full active set where conditionally linked active constraints are not present together, then proceed to Step 7f.
   (c) If $\exists z : \tilde{S} \in V_z$, append $\tilde{S}$ to S and $c^T x_{low}$ to S_{low}. Proceed to Step 7f.
   (d) Define $\tilde{G}_a(x)$ as the constraint set corresponding to $\tilde{S}$ and solve (67) ($N := N+1$). If infeasible, append $\tilde{S}$ to F and proceed to Step 7f. Otherwise, append $\tilde{S}$ to S and, denoting by $\tilde{x}^*$ the solution to (67), append $c^T \tilde{x}^*$ to S_{low}. Define V_c as the Boolean vector where $V_{c,i} = 1, \forall i : g_i(\tilde{x}^*) \geq 0$ (and 0 otherwise). Remove any rows z from V for which $V_z \in V_c$, together with the corresponding rows from X_V. Append V_c to V and $\tilde{x}^*$ to X_V.
   (e) If $g_i(\tilde{x}^*) \leq 0, \forall i = 1,...,n_g$ or if $N > 50$, run Subroutine D ($N := N+1$) with $\tilde{x}^*$ as the initial point in the former case and no initial point in the latter (in this case, reset $N := 0$). Update x_{up} if a better local optimum is found. If Λ^* changes, follow with Subroutines C ($N := N+N_C$), A, and B, and then repeat the procedure of Step 5.

For any rows i of S where $S_{low,i} > c^T x_{up}$, transfer the corresponding rows S_i to F and delete these rows from S_{low}. Find any indices $i : X_{V,i} \notin \mathscr{C}$ and remove these rows from V and from X_V.

(f) If k is the last element of I_k, return to Step 6. Otherwise, set k to the next element of I_k and return to 7a.

8. Let $I_k := \{i : \tilde{n}_g + 1 \leq i \leq n_g\}$, define I_F as in Step 7, and set $I_k := I_k \setminus I_F$. Set k equal to the first element of I_k:
   (a) Define the candidate active set, $\tilde{S}$, as $\tilde{S}^1$ with the k^{th} index set to 1.
   (b) If $\exists z : F_z \in \tilde{S}$, or if the set $\tilde{S}$ is an active set where conditionally linked active constraints are not present together, then proceed to Step 8d.
   (c) Define $G_a(x)$ as the constraint set corresponding to $\tilde{S}$ and solve (22), denoting the solution by x^*_{cand}. If $g_i(x^*_{cand}) \leq \varepsilon_g, \forall i = 1, ..., n_g$, then append x^*_{cand} to X^*.
   (d) If k is the last element of I_k, return to Step 6. Otherwise, set k to the next element of I_k and return to 8a.

Some remarks regarding this algorithm follow:

- There are three ways for Algorithm 3 to terminate. Criteria I and II will be defined as the termination due to a sufficiently tight $\mathscr{C}$, which yields a relaxed solution that either, in the case of I, satisfies the concave constraints with an acceptable tolerance or, in the case of II, yields a lower bound on the cost that is sufficiently close to the value for a feasible local minimum that has already been found. Criterion III indicates that a full active set search has been carried out, in which case the full set of candidates X^* is reported (the member with the lowest cost value corresponding to the global minimum). If X^* is empty, then this implies that the RCP problem is infeasible.
- Note that Termination Criteria I and II do not require the RCP problem to be regular, as both declare a solution by more "traditional" means. Regularity is required for Termination Criterion III to be valid, however. Also note that Criterion III will yield *all* global minima in the case that multiple equally good minima exist, while I and II may terminate as soon as just one of these is found and proven to be globally minimal within a certain tolerance.
- Some care should be taken with respect to the numerical tolerances of the multiple optimization problems and subroutines involved in the algorithm, as failing to do so may lead to a nonrobust implementation with some feasible solutions being fathomed due to slight numerical infeasibility. As just one example of where things could go wrong, consider the case of a local solver finding the globally minimal cost (in Subroutine D) lowered by a numerical error of -10^{-4}, and thereby reporting $c^T x_{up} = c^T x^* - 10^{-4}$. If this is then incorporated into $\mathscr{C}$ as a cutting plane constraint on the cost, which is, in turn, used by a different (presumably more efficient and convex) solver to solve the domain reduction and relaxed problems, it may be that the latter cannot find a feasible solution since the reported upper bound is slightly below what is feasible. Details regarding where all such tolerances should be accounted for would result in a lengthy discussion, which the reader is spared, but it is worth noting that they are quite important nevertheless.
- The counter N adds a heuristic rule by which the RCP solver decides to "take a break" from the active set search to perform a local minimization and hopefully find a new local optimum with which to refine $\mathscr{C}$. This is the only non-deterministic feature of the algorithm, since the initial start point for the local minimization will be randomly generated and, if a very good point is found, may lead to significant reductions in $\mathscr{C}$ and therefore to earlier termination.

- Algorithm 1 builds the active-set tree (e.g. Fig. 4) dimension-by-dimension, which results in Step 7 being exhausted before Step 8 is reached, with the latter corresponding to the solution of the reverse problems (22) for any active sets that have not been fathomed. Since the validation basis V is no longer needed in Step 8, it is no longer updated or used there.
- The choice to order the elements of I_k in increasing order is not mandatory, and other choices could be proposed. Essentially, this affects how the constraints are ordered when growing the branches, and is likely to affect performance. It is difficult to say if this choice could be optimized, however.

5 Illustrative Examples

Several NLP examples are chosen so as to illustrate the different benefits and drawbacks of the extended RCP method. Specifically, the first two problems are those where RCP should have an advantage, as these NLPs are innately RCP and have reasonable $\binom{n_g}{n-n_C}$ values. Example 6 in particular illustrates a case where the curse of dimensionality with respect to the number of variables is broken by the RCP framework. Example 7 extends RCP to the particular class of concave minimization problems for a modest dimension size of $n = 10$, and shows that while the method may work well here, the computation times may differ significantly depending on the available feasible points and on the nonlinearity of the problem. Examples 8 and 9 then consider simple two-dimensional NLPs where piecewise-concave approximations are needed to obtain the RCP form, and demonstrate how one could still get guarantees of ε-optimality in the presence of approximation error. Example 9 also shows a case where the global minimum is not unique (though unique locally). Finally, Example 10 is application flavored and applies the extended RCP method to a maximum-likelihood linear regression where the probability density function of the noise is bimodal.

In each case, the algorithm was coded in MATLAB®, with the CVX-SeDuMi modeling-solver combination [35, 12, 8] used for solving all of the convex problems presented in this paper, while any local (nonconvex) minimizations were done with the MATLAB routine `fmincon`. For all optimizations, it was verified that the solution converged to a local minimum. It should be noted that none of these choices are considered as computationally optimal, but rather as convenient and sufficient for the proof-of-concept message presented here. Since the dominant computational effort lies in the number of optimization problems solved by Algorithm 3, the computational effort for each example is reported in terms of the number of times that each type of optimizer is called, with the following three types being relevant:

- "Convex": Problems (67) and (22), which are general convex NLPs.
- "LP": Given as two numbers, $N_1 + N_2$, with N_1 denoting the LP problems (82) and (83) solved during domain reduction and N_2 denoting the LP relaxation (90) solved over $\mathscr{C}$.
- "Local": Problem (1) solved to local optimality.

The tolerances were defined as $\varepsilon = 10^{-3}, \varepsilon_g = 10^{-6}, \varepsilon_X = 10^{-4}, \varepsilon_{loc} = 10^{-6}$, and $U := 10^6$ was used for Subroutine D. The regularity of each problem was justified individually.

Example 5 (A Low-Dimensional Problem with Concave Constraints)

The following problem is solved:

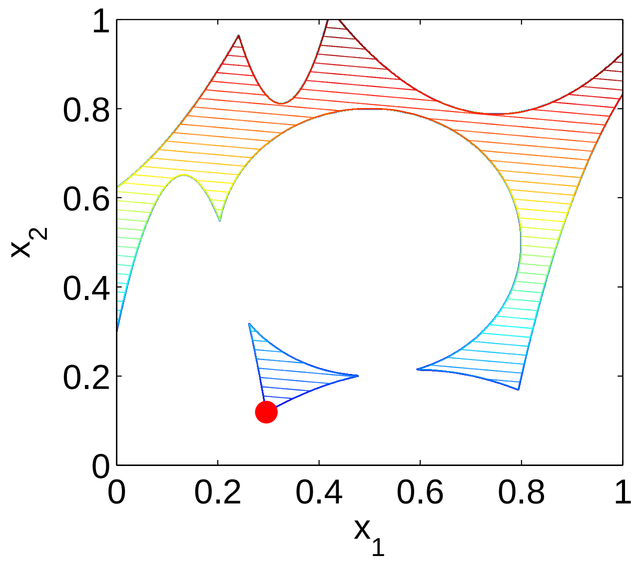

Fig. 7 The feasible region (lined) of (91) with the cost contours and global minimum given for the case of $c_1 = 0.1, c_2 = 1.0$.

$$
\begin{array}{ll}
\underset{x_1,x_2}{\text{minimize}} & c^T x \\
\text{subject to} & -2.42(x_1+0.4)^2 + 1.1x_1 + x_2 - 0.235 \leq 0 \\
& -1.1x_1^2 + 1.3x_1 - x_2 - 0.17 \leq 0 \\
& -e^{-5x_1+4} - x_2 + 1.2 \leq 0 \\
& -(x_1-0.5)^2 - (x_2-0.5)^2 + 0.09 \leq 0 \\
& -22(x_1-0.3)^2 + 1.1x_1 + x_2 - 1.155 \leq 0 \\
& -2.2(x_1-0.5)^2 + 1.1x_1 + x_2 - 1.475 \leq 0 \\
& -20(x_1-0.1)^2 + 1.3x_1 - x_2 + 0.5 \leq 0 \\
& x_1, x_2 \in [0,1]
\end{array}, \tag{91}
$$

which is already in standard RCP form and has a disjoint feasible region as shown in Fig. 7.

The proof of regularity is easy and may be outlined as follows for the case where both $c_1 \neq 0$ and $c_2 \neq 0$:

1. Since $c_1 \neq 0$ and $c_2 \neq 0$, any $\delta x^* \in N(c^T)$ must have both elements non-zero.
2. There are no constraints that are null in any such δx^*.

From Fig. 7, it is clear that the problem is not regular for $c_1 = 1, c_2 = 0$, for $c_1 = -1, c_2 = 0$, or for $c_1 = 0, c_2 = -1$, as all of these realizations have an infinite number of global minima that are not locally unique. Although not able to be proven regular for $c_1 = 0, c_2 = 1$, the problem is effectively so since the constraint $x_2 \geq 0$, which hinders the regularity proof, may be removed without affecting the feasible space.

The computational results for ten randomly generated c are reported in Table 1, and it is seen that the computational burden for this problem is quite light regardless of how the algorithm terminates. It is worth noting that the brute enumeration approach would require

Table 1 Computational effort for Example 5.

c_1	c_2	Convex	LP	Local	Termination
0.1	1.0	3	8 + 2	2	I
1.0	0.4	0	4 + 1	1	I
-0.2	0.7	0	6 + 1	1	II
-0.1	0.1	0	5 + 1	1	II
-0.6	2.2	0	10 + 1	1	II
0.7	1.6	4	16 + 2	2	II
0.6	-0.6	13	4 + 1	1	III
1.1	0.1	0	5 + 1	1	I
-0.1	-0.8	0	4 + 1	1	I
0.3	-1.3	9	4 + 1	1	III

solving $\binom{11}{2} = 55$ convex problems to arrive at the solution, which, though probably acceptable, still requires a lot more computation than Algorithm 3. Finally, one sees that in over half of the cases, domain reduction finds the solution before the active-set search even begins.

Example 6 (High-Dimensional RCPs with Favorable Complexity)

Consider the RCP:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & c^T x \\ \text{subject to} & -\sum_{i=1}^{n} w_i x_i^2 + r \leq 0 \\ & x \geq \mathbf{0} \end{array} , \tag{92}$$

with $c, w \in \mathbb{R}^n_{++}$, and $r = 1$. A qualitative cut of this problem is shown in Fig. 8, from which it is easily seen that the difficulty arises from the ellipse centered at the origin, generated by the single concave constraint. This is, however, an example of an RCP with favorable complexity, as the number of active sets (without any fathoming) is equal to $\binom{n+1}{n} = n+1$ and scales linearly in n. As such, one could always solve this problem by solving n convex optimization problems (the active set corresponding to only $x \geq \mathbf{0}$ may be fathomed as the solution for this set, $x = \mathbf{0}$, is clearly infeasible). The regularity of this problem is easily proven as follows:

1. The constraint $-\Sigma_{i=1}^n w_i x_i^2 + r \leq 0$ is mandatory and must be active at any global minimum (otherwise, $\mathbf{0}$ would be the trivial solution).
2. Since this constraint is nonlinear in all variables, a perturbation in any feasible $\delta x^* \in N(c^T)$ will make this constraint inactive, thereby losing global optimality.

Algorithm 3 is run for various dimension sizes n, with c and w randomly generated in each case. As the algorithm requires upper bounds on the variables as well, an additional set of constraints $x \leq \mathbf{100}$ is provided, although these are quickly fathomed by the algorithm. Table 2 reports the results, where it is seen that the active-set search, although not as efficient as the brute enumeration for this problem, still manages to scale very nicely with dimensionality, solving approximately n convex problems in each case. Significant extra work is introduced by the domain reduction LPs, but these too appear to scale linearly (on average) with n.

Example 7 (Concave Minimization)

The following problem is solved:

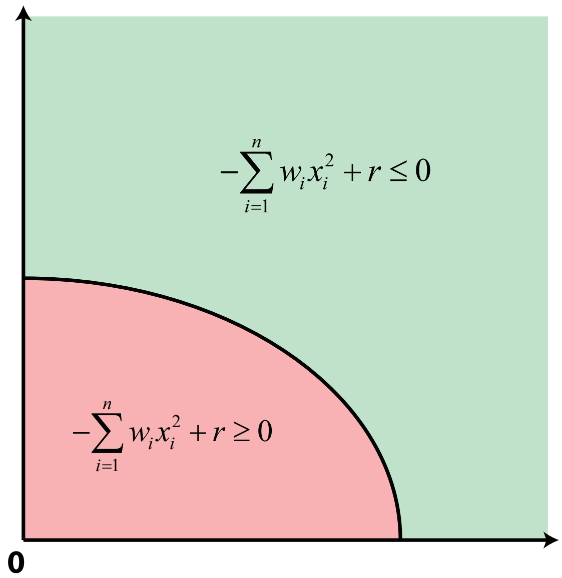

Fig. 8 A two-dimensional cut of the feasible set from the example in (92), where the strict positivity of the cost vector coefficients ensures that the solution always lie on the intersection of the strictly concave constraint and $n-1$ of the linear constraints.

Table 2 Computational effort for Example 6.

n	Convex	LP	Local	Termination
20	22	40 + 1	1	III
40	43	80 + 1	2	III
60	63	240 + 2	3	III
80	82	402 + 3	3	III
100	104	596 + 3	4	III
120	125	360 + 3	3	III
140	143	423 + 3	3	III
160	163	642 + 2	3	III
180	183	1262 + 4	4	III
200	204	603 + 2	3	III

$$\begin{array}{ll} \underset{x_1,\ldots,x_{10}}{\text{minimize}} & x^T Q x + \alpha c^T x \\ \text{subject to} & Ax \leq b \\ & x_i \in [0,1], \;\; i = 1,\ldots,10 \end{array} , \tag{93}$$

where A and b are defined as follows:

$$A = \begin{bmatrix} 2 & -6 & -1 & 0 & -3 & -3 & -2 & -6 & -2 & -2 \\ 6 & -5 & 8 & -3 & 0 & 1 & 3 & 8 & 9 & -3 \\ -5 & 6 & 5 & 3 & 8 & -8 & 9 & 2 & 0 & -9 \\ 9 & 5 & 0 & -9 & 1 & -8 & 3 & -9 & -9 & -3 \\ -8 & 7 & -4 & -5 & -9 & 1 & -7 & -1 & 3 & -2 \end{bmatrix}, \;\; b = \begin{bmatrix} -4 \\ 22 \\ -6 \\ -23 \\ -12 \end{bmatrix}, \tag{94}$$

Table 3 Computational effort for Example 7. The asterisk (*) denotes the case where the algorithm was warm started with a cutting plane constraint $\hat{c}^T x \leq \hat{c}^T x^*$ in $\mathscr{C}$, with $\hat{c}^T x^*$ the cost value at the global minimum.

α	Convex	LP	Local	Termination
-10	4	71 + 1	2	III
-1	3	96 + 1	2	III
-0.1	6	113 + 1	2	III
0.1	356	165 + 3	10	III
1	1005	242 + 14	24	III
1*	3	139 + 2	2	III
10	3	90 + 1	2	III

and where $Q = -50I$, $c = [48\ \ 42\ \ 48\ \ 45\ \ 44\ \ 41\ \ 47\ \ 42\ \ 45\ \ 46]^T$. α is a scalar that is varied for test purposes, with $\alpha := 1$ corresponding to Test Problem 2.6 from [10].

This problem is easily converted into standard RCP form by employing the epigraph transformation:

$$\begin{array}{ll} \underset{x_1,...,x_{11}}{\text{minimize}} & \hat{c}^T x \\ \text{subject to} & x^T \hat{Q} x - x_{11} \leq 0 \\ & \hat{A} x \leq b \\ & x_i \in [0,1],\ \ i = 1,...,10,\ \ x_{11} \in [-500, 0] \end{array} , \tag{95}$$

with:

$$\hat{c} = \begin{bmatrix} \alpha c \\ 1 \end{bmatrix},\ \hat{Q} = \begin{bmatrix} Q & \mathbf{0}_{10\times 1} \\ \mathbf{0}_{1\times 10} & 0 \end{bmatrix},\ \hat{A} = [A\ \ \mathbf{0}_{5\times 1}]. \tag{96}$$

The bounds on the auxiliary variable x_{11} correspond to the minimal and maximum values of $x^T \hat{Q} x$ over X. It is clear that the new constraint is mandatory (i.e. it must be active at the global optimum), and as such must belong to any active set that is considered.

The proof of regularity is similar to that of Example 6:

1. Any direction $\delta x^* \in N(\hat{c}^T)$ must contain at least two non-zero elements.
2. The constraint $x^T \hat{Q} x - x_{11} \leq 0$ is both mandatory and nonlinear in any such direction.
3. Since $x^T \hat{Q} x - x_{11} \leq 0$ must be active at the global minimum, a step in any such δx^* will render it inactive and thereby lose global optimality.

Table 3 presents the computational results for different values of α. As might be expected (see, e.g., the discussion in [22]), problems with relatively small nonlinear effects (i.e. $\alpha = \pm 10$) are solved very quickly due to the local minimization and domain reduction subroutines being very effective at finding a good local minimum and reducing the box accordingly. When the nonlinear effects are more significant ($\alpha = 0.1, 1.0$), the computational effort can increase significantly. This can, however, be overcome by providing a good initial point or "guess" of the globally minimal cost. For $\alpha = \pm 0.1$, it should be noted that the initial point found is much better for the case of $\alpha = -0.1$ than for that of $\alpha = 0.1$ (hence the discrepancy in computational effort between these two). For $\alpha = 1.0$, compare the performance when no initial upper bound is given with the performance when the global minimum value is provided from the onset, allowing major reductions in X due to the constraint $\hat{c}^T x \leq \hat{c}^T x^*$ being incorporated into $\mathscr{C}$. This serves to reinforce the importance of the "good" cutting plane constraints in the RCP scheme.

Example 8 (Extended RCP and Approximation Error Bounds)

Problem (58) is considered and reformulated into an RCP as shown in (59)-(61), with a piecewise-linear function of n_p pieces used to either under-approximate or over-approximate the convex portion of the true constraint. Clearly, using an under-approximation will yield an RCP that has a larger feasible set than the original problem, with the opposite true for an over-approximation. From this it follows that solving the under-RCP will produce a lower bound on the globally minimal cost, while solving the over-RCP will produce an upper bound. The gap between the two may then be seen as a bound on the approximation error and may be used as a sort of feedback on the appropriate value of n_p, as choosing n_p to be very large *a priori* would, though yielding a very accurate answer, introduce more constraints into the RCP problem and thereby increase the computational burden.

Regularity is proven as follows:

1. One of the $p_i(x_4) - x_1^2 - x_2^2 - x_3 \leq 0$ constraints must be active at a global minimum due to the epigraph transformation of the cost.
2. Any $\delta x^* \in N\left(\begin{bmatrix} c^T \\ C \end{bmatrix}\right)$ with non-zero first or second elements is removed from consideration as this constraint is nonlinear in all such directions, making it impossible to preserve global optimality while moving in any such δx^*. Only δx^* where the first two elements are 0 are therefore pertinent.
3. However, no such δx^* exists since $\delta x^* \in N(C)$ implies that the fourth element is 0 as well, but this precludes the existence of any non-zero $\delta x^* \in N(c^T)$.

Noting that the global minimum of this problem lies at $[1.1667, 0.5000]$ with a cost value of -1.0833, the problem is now solved for both under- and over-approximations with increasing n_p values. The results, given in Table 4, show that there is a steady increase in both computational effort and solution accuracy as n_p increases, although there are occasional (pleasant) surprises with respect to the former (see $n_p = 200$ for the over-approximation case). Depending on the user's requirements, the procedure of increasing n_p could be brought to an end once the lower and upper bounds grow sufficiently close – for $n_p = 200$, one sees that the gap is in the fourth digit, for example, which may be sufficiently accurate. It is also worth noting that the upper bounds provided by the over-approximate RCP solution can be further improved upon by a final local minimization – as the solution point here must be a feasible point for the original problem and therefore can only be improved upon by any local descent method.

Example 9 (Nonlinear Equality Constraints and Multiple Global Minima)

Problem (62) is considered and placed into RCP form via the steps detailed in (63)-(66). As mentioned earlier, the equality constraint $x_3 = \sin x_1$ is broken into two inequalities and the nonlinear univariate functions $\sin x_1$ and $-\sin x_1$ are under-approximated by piecewise-concave functions (see also Fig. 6). It is implicit that these must be under-approximations, as over-approximating both would lead to an infeasible set. So as not to mix lower and upper bounds, the convex function is under-approximated as well, and so the resulting RCP problem has a larger feasible set than the original and thereby yields a lower bound on the globally minimal cost upon solution. An upper bound may be obtained by taking the solution of the RCP problem and using it as a starting point for a local minimization of the original problem. Like in the previous example, it is clear that finer and finer approximations may be used until the gap between the lower and upper bounds is sufficiently small[5].

The following reasoning is used to prove regularity:

[5] For simplicity, the number of approximation pieces for each function is the same, with $n_a = n_b = n_c = n_p$.

Table 4 Computational effort for Example 8. Here, $(-)$ and $(+)$ denote under- and over-approximations, respectively.

n_p	Convex	LP	Local	Termination	x^*	$c^T x^*$
3−	0	9 + 1	1	II	[1.4590,1.4692]	-2.9307
3+	11	11 + 1	2	III	[1.5000,1.5000]	-0.2500
5−	0	18 + 1	1	I	[1.1348,0.4045]	-1.5418
5+	9	10 + 1	2	III	[1.2500,0.7500]	-1.0625
10−	13	14 + 1	2	III	[1.1667,0.5000]	-1.2377
10+	26	10 + 1	2	III	[1.2500,0.7500]	-1.0625
15−	27	11 + 1	2	III	[1.1964,0.5893]	-1.1445
15+	42	10 + 1	2	III	[1.0833,0.2500]	-1.0625
20−	34	14 + 1	2	III	[1.2105,0.6316]	-1.1122
20+	46	10 + 1	2	III	[1.1250,0.3750]	-1.0781
30−	36	45 + 2	3	III	[1.1379,0.4138]	-1.0957
30+	40	48 + 2	3	III	[1.1667,0.5000]	-1.0833
50−	45	110 + 3	3	I	[1.1837,0.5510]	-1.0877
50+	52	89 + 2	4	III	[1.1500,0.4500]	-1.0825
100−	87	129 + 3	3	I	[1.1667,0.5000]	-1.0846
100+	87	134 + 3	3	I	[1.1750,0.5250]	-1.0831
200−	172	292 + 4	4	I	[1.1709,0.5126]	-1.0836
200+	19	192 + 3	3	II	[1.1750,0.5250]	-1.0831

1. At least one of the constraints $p_{c,i}(x_6) - x_3^2 - x_4^2 - x_5 \leq 0$ must be active at a global minimum due to the epigraph transformation of the cost.
2. It follows that any $\delta x^* \in N\left(\begin{bmatrix} c^T \\ C \end{bmatrix}\right)$ with the third and fourth elements non-zero may be excluded from consideration since the mandatory constraint is nonlinear in any such δx^* and therefore cannot remain active if perturbed locally in those directions.
3. The equality constraint $x_3 + x_4 - x_6 = 0$ then requires that the sixth element of δx^* be 0 as well.
4. In order for $\delta x^* \in N(c^T)$, it is necessary that the fifth element of δx^* also be 0. Overall, it is now clear that only those δx^* with the first two elements being non-zero are of relevance.
5. From the equality constraint $-x_1 + 0.3x_2 - x_4 = 0$, it follows that both the first and second elements must be non-zero. As such, $\delta x^* = [0.3\delta x \;\; \delta x \;\; 0 \;\; 0 \;\; 0 \;\; 0]^T$, with δx being any non-zero number.
6. However, there are no constraints that are null in this δx^* and that could allow for the satisfaction of (5).

Virtually all of the innate fathoming rules given in Lemma 3 are relevant here. Apart from the “inconvenience” of a nonlinear equality constraint, this problem has an additional difficulty in that the objective function exhibits a symmetry (Fig. 9) and has two global minima (at $[-1.8601, 5.0000]$ and $[1.8601, -5.0000]$), both with a cost value of -3.2205. The consequence of this is that domain reduction is unlikely to be as effective as it may be in certain problems, due to the two global minima being dispersed on nearly opposite corners of the original domain and the impossibility of shrinking the domain without fathoming one of these minima. The computational results, given in Table 5, largely confirm these expectations, with domain reduction only playing a significant role for a poorly approximated problem ($n_p = 3$), where it is likely that there is a significant discrepancy between the two global minima due to approximation error. For better approximated problems, no real reduction occurs and the computational effort increases significantly. A smarter management of

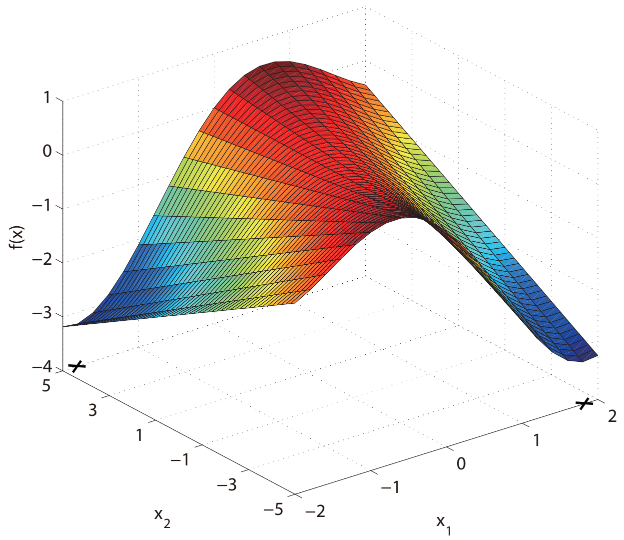

Fig. 9 The function $f(x) = (\sin x_1)(-x_1 + 0.3x_2)$, which has two global minima (marked) over $x_1 \in [-2,2], x_2 \in [-5,5]$ at $[-1.8601, 5.0000]$ and $[1.8601, -5.0000]$.

Table 5 Computational effort for Example 9.

n_p	Convex	LP	Local	Termination	x^*	$c^T x^*$	Upper Bound
3	0	47 + 1	1	I	[2.0000,-5.0000]	-6.4366	-3.2205
5	389	11 + 3	3	III	[-2.0000,5.0000]	-3.5217	-3.2205
10	1525	11 + 2	9	III	[1.5966,-5.0000]	-3.2692	-3.2205
20	6109	12 + 2	27	III	[-1.8246,5.0000]	-3.2623	-3.2205

the approximations is clearly desired for these cases, as it is seen that simply breaking them into finer and finer evenly-dispersed pieces leads to significant rises in the computations but small improvements in the lower bounds (compare the case of $n_p = 10$ with that of $n_p = 20$).

Example 10 (Maximum-Likelihood Regression with a Bimodal Noise Distribution)

A univariate "unknown" affine function, $L_1 u + L_2$, is sampled with noise v at n_y evenly-spaced discrete instants $u_1, ..., u_{n_y}$ over the interval $u \in [0,1]$:

$$y_i = L_1 u_i + L_2 + v_i, \;\; i = 1, ..., n_y, \tag{97}$$

with the problem of interest being to estimate the parameters L_1 and L_2 from the available input-output (u,y) data and a known probability density function (pdf) for v.

When v is additive white Gaussian noise, the standard approach to solving this problem is that of simple least-squares linear regression, with the resulting fit being the one that is the most statistically likely from the maximum-likelihood perspective. This is because the general (log-)likelihood formulation

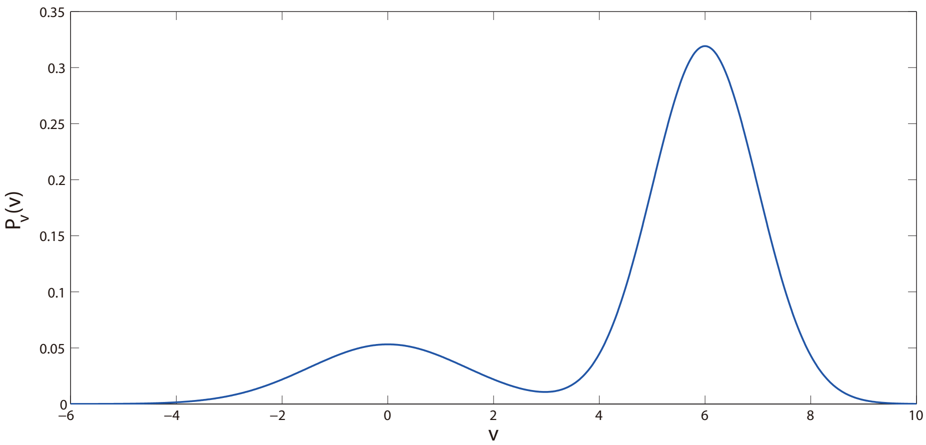

Fig. 10 The probability density function of the noise in Example 10.

$$\begin{array}{ll} \underset{L_1,L_2,\hat{v}}{\text{minimize}} & -\sum_{i=1}^{n_y} \log P_v(\hat{v}_i) \\ \text{subject to} & \hat{v}_i = y_i - L_1 u_i - L_2, \;\; i = 1,...,n_y \end{array}, \tag{98}$$

reduces to a least-squares problem when the pdf $P_v : \mathbb{R} \to \mathbb{R}$ is that of the normal Gaussian distribution [5]. More generally, Problem (98) may be solved efficiently using convex programming whenever the function P_v is log-concave, but requires global optimization to guarantee best solutions otherwise.

The following bimodal pdf is considered in this example (Figure 10):

$$P_v(v) = \frac{1}{7.5\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{v}{1.5}\right)^2} + \frac{1}{1.25\sqrt{2\pi}} e^{-\frac{1}{2}(v-6)^2}. \tag{99}$$

So as to cast this problem as an RCP, the log-likelihood penalty function $-\log P_v(v)$ is over-approximated by a piecewise-concave function $p(v)$ of just three pieces, chosen here in an *ad hoc* manner (rather than being calculated by Algorithm 1) so as to capture the function's main trends without creating too much of a computational burden (Figure 11). This leads to the reformulation:

$$\begin{array}{ll} \underset{L_1,L_2,\hat{v}}{\text{minimize}} & \sum_{i=1}^{n_y} \max \left(p_1(\hat{v}_i), p_2(\hat{v}_i), p_3(\hat{v}_i) \right) \\ \text{subject to} & \hat{v}_i = y_i - L_1 u_i - L_2, \;\; i = 1,...,n_y \end{array}, \tag{100}$$

and, with the addition of auxiliary variables s and the epigraph transformation, to the RCP form:

$$\begin{array}{ll} \underset{L_1,L_2,\hat{v},s}{\text{minimize}} & \sum_{i=1}^{n_y} s_i \\ \text{subject to} & \hat{v}_i = y_i - L_1 u_i - L_2, \;\; i = 1,...,n_y \\ & p_1(\hat{v}_i) - s_i \leq 0, \;\; i = 1,...,n_y \\ & p_2(\hat{v}_i) - s_i \leq 0, \;\; i = 1,...,n_y \\ & p_3(\hat{v}_i) - s_i \leq 0, \;\; i = 1,...,n_y \end{array}, \tag{101}$$

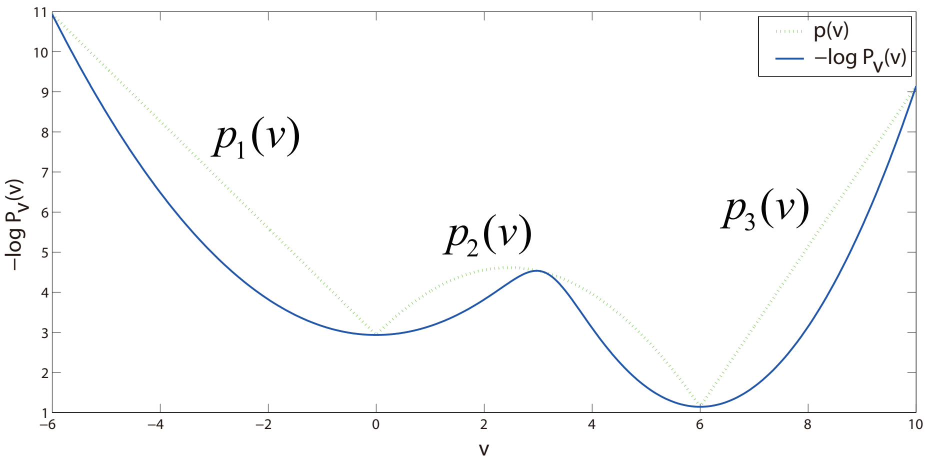

Fig. 11 The negative log-likelihood penalty function and its piecewise-concave approximation.

with the additional box constraints $-6 \leq v_i \leq 10$ and $0 \leq s_i \leq 10$ set based on the knowledge of P_v. The "real" function is defined as having $L_1 = 10$ and $L_2 = 0$, and *a priori* box bounds of $0 \leq L_1 \leq 20$ and $-20 \leq L_2 \leq 20$ are assumed. Rather than attempt to rigorously bound the solution (as in Examples 8 and 9), the answer obtained by solving (101) is taken as an initial point for the local minimization of the original problem, which proves sufficient to find the global minimum in all cases tested here.

RCP regularity is now (loosely) justified:

1. By the epigraph transformation, it follows that either $p_1(\hat{v}_i) - s_i \leq 0$, $p_2(\hat{v}_i) - s_i \leq 0$, or $p_3(\hat{v}_i) - s_i \leq 0$ must be active at the global minimum for each $i = 1, ..., n_y$.
2. Of the three pieces, only $p_2(\hat{v}_i)$ is nonlinear in $\hat{v}_i$ (see Fig. 11). Suppose that $p_2(\hat{v}_i) - s_i \leq 0$ is active at the global minimum for two data points. Assign them, for simplicity, as $i = 1$ and $i = 2$.
3. Since $p_2(\hat{v}_i)$ is nonlinear in $\hat{v}_i$, it follows that the elements in δx^* corresponding to both $\hat{v}_1$ and $\hat{v}_2$ must be 0, since otherwise $p_2(\hat{v}_1) - s_1 \leq 0$ and $p_2(\hat{v}_2) - s_2 \leq 0$ cannot remain active for a local perturbation in δx^*.
4. Since the constraints $\hat{v}_1 = y_1 - L_1 u_1 - L_2$, $\hat{v}_2 = y_2 - L_1 u_2 - L_2$ must hold in δx^*, the first and second δx^* elements that would allow this would be $[1 \quad -u_1]$ and $[1 \quad -u_2]$ for the two constraints, respectively, but these cannot be equal by the definition of u. As such, no such δx^* exists.

This justification is not entirely rigorous as it depends on the additional, though not very unrealistic, supposition that the $p_2(v)$ constraint will prove limiting for at least two data points. For a fully rigorous justification one could use nonlinear pieces for $p_1(v)$ and $p_3(v)$, but this was not done here, and the justification above was considered sufficient.

The results for different data set sizes n_y are presented in Table 6, with Figure 12 giving the visual performance for $n_y = 10$ and $n_y = 50$. There is a clear trend of the problem becoming less and less nonconvex (and therefore more susceptible to domain reduction) as n_y increases – compare, for example, the contour plots in Figure 12. This is reflected in the computational results, as domain reduction virtually takes over the computational effort for problems with greater n_y and almost single-handedly solves the problem.

Table 6 Computational effort for Example 10.

n_y	Convex	LP	Local	Termination
5	159	293 + 4	5	III
10	95	548 + 2	4	III
20	51	983 + 2	3	II
50	0	2039 + 1	1	II
100	1	7640 + 2	2	II
200	0	8840 + 1	1	II

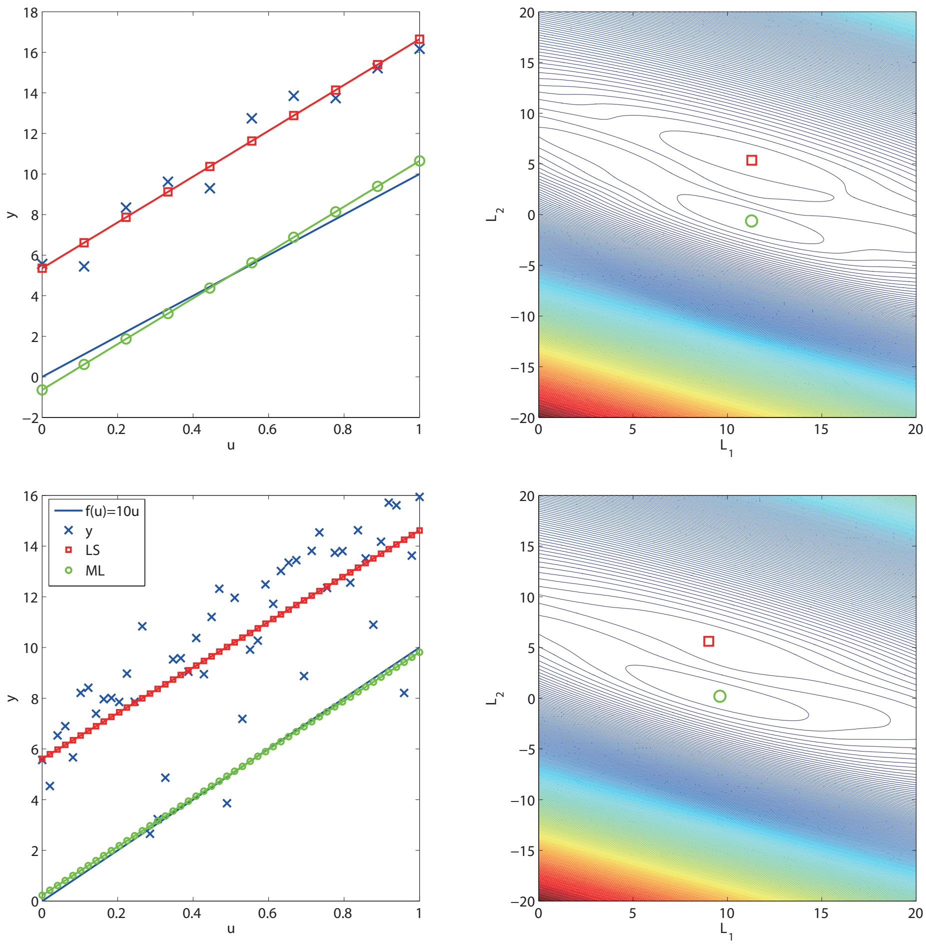

Fig. 12 Results for the maximum-likelihood regression of Example 10 for $n_y = 10$ (top) and $n_y = 50$ (bottom), with the plots on the left showing the linear regression of the noisy data and the plots on the right showing the contours of the original maximum-likelihood problem. Circles denote the linear fit (left) and the optimum (right) found by solving the problem with the RCP method, while squares denote the same results for the standard least-squares approach. In this case, the latter can be seen to correspond to an inferior local minimum.

General Observations

It is first stated that the proposed method found the global minimum for all the problems where this was readily verifiable, thereby confirming its theoretical properties and guarantees.

Perhaps the most striking aspect of the results is the role of domain reduction (Subroutine C) in the algorithm, as this allows for many problems to be solved without completing the active-set search. Even for many of the cases where the algorithm finished the search (e.g. Example 7), it was observed that the algorithm terminated much quicker following significant domain reduction. This is not too surprising, however, given the documented success of domain reduction techniques [25] and given the success of branch-and-reduce solvers like BARON [33,26]. It is also interesting to note, however, that even after nearly complete reduction to a single point (as observed for Example 7), the algorithm terminated by Criterion III and not by Criteria I or II – the tolerances ε and ε_g not being sufficiently large in these cases, but fast termination still made possible by the effectiveness of the fathoming methods in the active-set search. This may be seen as an advantage of the RCP scheme, in that it is able to avoid the clustering issue that may arise in standard branch-and-bound schemes [25].

The noted dependance on domain reduction also acts, not surprisingly, as a weakness. This is particularly obvious in Example 9, where the inability to reduce the domain due to the presence of two well-dispersed global minima leads to significantly slower solution times than in the other examples, although the dimensionality is not that large. A natural option would be to split the search space as is done in branch-and-bound, although this introduces a heuristic difficulty (how and when to split).

While not necessarily practical, Example 6 is nevertheless inspirational as it opens a new avenue for solving large problems with very acceptable computational effort. The key, of course, is the discrepancy between the number of possible active constraints and the degrees of freedom $n - n_C$. When effective fathoming techniques are able to reduce the number of constraints to be considered significantly, it is not inconceivable that good scaling properties may be noted for problems other than the one of Example 6. At the same time, Example 10 offers a case of a problem that scales well practically, though not theoretically, due to the effectiveness of domain reduction techniques.

6 Concluding Remarks: Towards Extended RCP Homotopy

The work in this paper has proposed a new approach to solving a broad class of NLP problems to global optimality by approximating them as RCP problems and then applying an active-set search scheme to find the active set defining the global minimum. While the preliminary results are promising, there is naturally a lot to be done before this method can be turned into a competitive NLP solver. Certainly, the fathoming "theory" presented in Section 4.2 is incomplete, and numerous advances, either already available or waiting to be discovered, can improve the effectiveness of these techniques further. The same may be said for domain reduction methods.

A seemingly more promising path for extended RCP improvement, however, may be the one of homotopy, where the fundamental idea would be to solve simpler, more brute RCP problems and recycle the obtained information to warm start more difficult ones. Examples 8 and 9 are perhaps the most direct illustration of this, as one sees that using coarser approximations improves computation times at the cost of accuracy. A homotopy approach for these kinds of problems could incorporate the following:

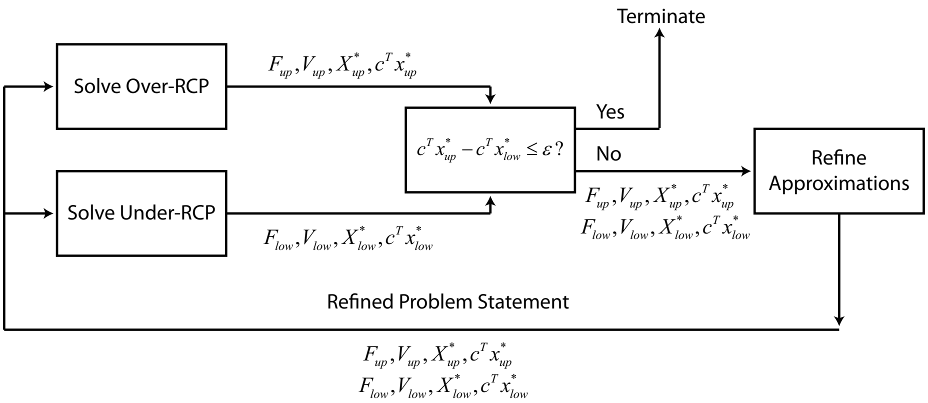

Fig. 13 A homotopy approach to solving extended RCP problems.

– starting with a small n_p, and using the obtained $c^T x^*$ as an *a priori* upper bound on the cost for problems with larger n_p,
– refining approximations only for those pieces that are shown to be relevant (i.e. not refining approximations of constraints that will be fathomed anyway),
– recycling feasible points obtained from problems with small n_p values to be used for problems with large n_p values,
– recycling fathomed constraint information wherever possible.

A very qualitative illustration of such a homotopy scheme, together with the idea of lower and upper bounds, is given in Fig. 13. Note, however, that posing an over-RCP may not always be so straightforward, as shown by Example 9 for the case of nonlinear inequality constraints.

An additional interesting trait of the homotopy approach is that it may be generalized to a standard RCP problem without approximations. The key technique at work here would be that of joint constraints. Consider, for example:

$$\begin{array}{l} g_1(x) \le 0 \\ g_2(x) \le 0 \end{array} \Rightarrow g_1(x) + g_2(x) \le 0\,, \tag{102}$$

where the original RCP set (left) is approximated by a relaxed RCP constraint (right). The latter would lead to an easier RCP problem with a larger feasible set, the solution of which would yield a lower bound on the original. Such techniques could potentially be used to apply homotopy methods to RCP problems with unfavorable complexity due to a large $\binom{n_g}{n-n_C}$ value.

The ideal version of such a homotopy method would likely be one that solves very difficult NLP problems via a sequence of simple RCPs that are intelligently refined so as to close the suboptimality gap, $c^T x^*_{up} - c^T x^*_{low}$, as quickly and efficiently as possible. Incorporating such intelligence into the refinement may not be trivial, however, and thus represents a worthy topic for future research.

References

1. Al-Khayyal, F. A., Falk, J. E.: Jointly constrained biconvex programming. Math. Oper. Res. 8(2), 273-286 (1983)

2. Avriel, M.: Methods for solving signomial and reverse nonconvex programming problems. In: Avriel, M. et al. (eds.) Optimization and Design, pp. 307-320. Prentice-Hall Inc., Englewood Cliffs (1973)
3. Bhattacharjee, S., Ramesh, R.: A multi-period profit maximizing model for retail supply chain management: An integration of demand and supply-side mechanism. Eur. J. Oper. Res. 122, 584-601 (2000)
4. Böhringer, M. C., Jacobsen, S. E.: Convergent cutting planes for linear programs with additional reverse convex constraints. Syst. Model. Optim.: Lect. Notes Control Inf. Sci. 59, 263-272 (1984)
5. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2008)
6. Caprara, A., Locatelli, M.: Global optimization problems and domain reduction strategies. Math. Program. 125(1), 123-137 (2010)
7. Chubanov, S.: Sensitivity Analysis and Efficient Algorithms for Some Economic Lot-Sizing and Scheduling Problems, PhD Thesis, University of Siegen (2006)
8. CVX Research, Inc.: CVX: MATLAB software for disciplined convex programming, version 2.0 beta. http://cvxr.com/cvx (2012). Accessed September 2012
9. Falk, J. E., Soland, R. M.: An algorithm for separable nonconvex programming problems. Manag. Sci. 15(9), 550-569 (1969)
10. Floudas, C. A., Pardalos, P. M., Adjiman, C. S., Esposito, W. R., Gümüs, Z. H., Harding, S. T., Klepeis, J. L., Meyer, C. A., Schweiger, C. A.: Handbook of Test Problems in Local and Global Optimization. Kluwer Academic Publishers, Dordrecht (1999)
11. Giannessi, F., Tardella, F.: Connections between nonlinear programming and discrete optimization. In: Du, D.-Z., Pardalos, P. M. (eds.) Handbook of Combinatorial Optimization (Vol. 1), pp. 149-188. Kluwer Academic Publishers, Boston (1998)
12. Grant, M., Boyd, S.: Graph implementations for nonsmooth convex programs, recent advances in learning and control (a tribute to M. Vidyasagar). In: Blondel, V., Boyd, S., Kimura, H. (eds.) Lecture Notes in Control and Information Sciences, pp. 95-110, Springer (2008)
13. Gurlitz, T. R., Jacobsen, S. E.: On the use of cuts in reverse convex programs. J. Optim. Theory Appl. 68(2), 257-274 (1991)
14. Hillestad, R. J., Jacobsen, S. E.: Reverse convex programming. Appl. Math. Optim. 6, 63-78 (1980)
15. Horst, R., Pardalos, P. M., Thoai, N.: Introduction to Global Optimization. Kluwer Academic Publishers, Dordrecht (1995)
16. Jacobsen, S. E., Moshirvaziri, K.: Computational experience using an edge search algorithm for linear reverse convex programs. J. Glob. Optim. 9(2), 153-167 (1996)
17. Kearfott, R. B.: Decomposition of arithmetic expressions to improve the behavior of interval iteration for nonlinear systems. Comput. 47, 169-191 (1991)
18. Locatelli, M., Raber, U.: Packing equal circles in a square: a deterministic global optimization approach. Discret. Appl. Math. 122(1-3), 139-166 (2002)
19. Mangasarian, O. L.: Nonlinear Programming. Society for Industrial and Applied Mathematics, Philadelphia (1994)
20. Marchetti, A., Chachuat, B., Bonvin, D.: A dual modifier-adaptation approach for real-time optimization. J. Process Control, 20(9), 1027-1037 (2010)
21. McCormick, G. P.: Computability of global solutions to factorable nonconvex programs: Part I - convex underestimating problems. Math. Program. 10, 147-175 (1976)
22. McKeown, P.: Extreme point ranking algorithms: A computational survey. In: White, W. W. (ed.) Computers and Mathematical Programming: Proceedings of the Bicentennial Conference on Mathematical Programming, pp. 216-222. U.S. Government Printing Office, Washington (1976)
23. Melachrinoudis, E., Smith, J. M.: An O(mn^2) algorithm for the maximin problem in E^2. Oper. Res. Lett. 18(1), 25-30 (1995)
24. Meyer, R.: The validity of a family of optimization methods. SIAM J. Control. 8, 41-54 (1970)
25. Neumaier, A.: Complete search in continuous global optimization and constraint satisfaction. Acta Numer. 13(1), 271-369 (2004)
26. Neumaier, A., Shcherbina, O., Huyer, W., Vinkó, T.: A comparison of complete global optimization solvers. Math. Program. 103(2), 335-356 (2005)
27. Pardalos, P. M., Rosen, J.: Methods for global concave minimization: A bibliographic survey. SIAM Rev. 28(3), 367-379 (1986)
28. Rodger, E.: Dual Modifier Adaptation Methodology for the On-line Optimization of Uncertain Processes. Master's thesis, McMaster University (2010)
29. Rosen, J. B.: Iterative solution of nonlinear optimal control problems. SIAM J. Control. 4(1), 223-244 (1966)
30. Rozvany, G. I. N.: A new calculus for optimum design. Int. J. Mech. Sci. 9, 885-886 (1967)
31. Rozvany, G. I. N.: Concave programming in structural optimization. Int. J. Mech. Sci. 12, 131-142 (1970)
32. Rozvany, G. I. N.: Concave programming and piece-wise linear programming. Int. J. Numer. Methods Eng. 3, 131-144 (1971)

33. Ryoo, H. S., Sahinidis, N. V.: A branch-and-reduce approach to global optimization. J. Glob. Optim. 8(2), 107-138 (1996)
34. Sen, S., Sherali, H. D.: Nondifferentiable reverse convex programs and facetial convexity cuts via a disjunctive characterization. Math. Program. 37, 169-183 (1987)
35. Sturm, J. F.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optim. Methods Softw. 11(1-4), 625-653 (1999)
36. Tuy, H.: Convex programs with an additional reverse convex constraint. J. Optim. Theory Appl. 52(3), 463-486 (1987)
37. Tuy, H.: Canonical DC programming problem: Outer approximation methods revisited. Oper. Res. Lett. 18(2), 99-106 (1995)
38. Tuy, H.: Convex Analysis and Global Optimization. Kluwer Academic Publishers, Dordrecht (1998)
39. Ueing, U.: A combinatorial method to compute a global solution of certain non-convex optimization problems. In: Lootsma, F. A. (ed.), Numerical Methods for Non-Linear Optimization, pp. 223-230. Academic Press (1972)
40. Zamora, J. M., Grossmann, I. E.: A branch and contract algorithm for problems with concave univariate, bilinear and linear fractional terms. J. Glob. Optim. 14(3), 217-249 (1999)
41. Zangwill, W. I.: The piecewise concave function. Manag. Sci. 13(11), 900-912 (1967)