J Glob Optim (2010) 46:163–189
DOI 10.1007/s10898-009-9415-1

Outer approximation algorithms for canonical DC problems

Giancarlo Bigi · Antonio Frangioni · Qinghua Zhang

Received: 26 March 2008 / Accepted: 10 March 2009 / Published online: 25 March 2009

Abstract The paper discusses a general framework for outer approximation type algorithms for the canonical DC optimization problem. The algorithms rely on a polar reformulation of the problem and exploit an approximated oracle in order to check global optimality. Consequently, approximate optimality conditions are introduced and bounds on the quality of the approximate global optimal solution are obtained. A thorough analysis of properties which guarantee convergence is carried out; two families of conditions are introduced which lead to design six implementable algorithms, whose convergence can be proved within a unified framework.

Keywords DC problems · Polar set · Approximate optimality conditions · Cutting plane algorithms

1 Introduction

Nonconvex optimization problems often arise from applications in engineering, economics and other fields (see, for instance, [10,13]). Often, these problems either have a natural formulation or can be reformulated as *DC optimization* problems, that is nonconvex problems where the objective function is the difference of two convex functions and the constraint can be expressed as the set difference of two convex sets. In turn, every DC optimization problem can be reduced to the so-called *canonical DC* (CDC) problem through standard

G. Bigi (✉) · Q. Zhang
Dipartimento di Informatica, Università di Pisa, Largo B.Pontecorvo 3, 56127 Pisa, Italy
e-mail: giancarlo.bigi@di.unipi.it; bigig@di.unipi.it

Q. Zhang
e-mail: zhang@di.unipi.it

A. Frangioni
Dipartimento di Informatica, Università di Pisa, Polo Universitario della Spezia, Via dei Colli 90, 19121 La Spezia, Italy
e-mail: frangio@di.unipi.it

transformations, which provide a linear objective function while preserving the structure of the constraint [22].

Several algorithms to solve the CDC problem have been proposed [7,11,12,18,22,23,27, 28,31]; many of them are modifications of the first cutting plane algorithm proposed by Tuy in [22]. The original algorithm builds outer approximations of the intersection between the feasible region and the level sets of the objective function, and seeks a point there inside. If the point is infeasible, then the approximation is improved by cutting it off; otherwise it gives a better feasible value, and the corresponding level set provides a cutting plane. Allowing slackened "objective cuts", i.e. requiring a fixed decrease for the current value, the algorithm may terminate with an infeasible (approximate optimal) solution. A further variant was developed in [23] for the case in which the linear objective function is replaced by a convex finite-valued function although this can also be recast as a CDC program. By deploying non-slackened "objective cuts", the algorithm terminates with a feasible approximate optimal solution. A general conceptual framework of outer approximation type algorithms was given in [28]. This algorithmic framework is more efficient and overcomes some defects of the previous algorithms; for instance, [28] allows to handle the infeasible case, while [22,23] require a feasible solution as starting point.

The *polyhedral annexation* algorithms, more akin to those presented in this paper, were proposed in [27,31] for the special case of a linear program with an additional reverse convex constraint. These algorithms exploit the "dual form" of the standard optimality condition and this leads to check optimality in a different way: while outer approximation methods need a procedure for a general convex maximization problem, polyhedral annexation methods need the computation of the maximum inner product between variables from two convex sets [(see (8)]. Afterwards, [29,33] interpreted polyhedral annexation method as *dual method* and showed that this algorithm can be extended to any CDC problem.

Several attempts at generalizing the results in the above papers were not entirely successful. A variant of [23] has been proposed in [11], where a binary search on the current value is proposed; however, this is unnecessary since it does not improve the convergence properties of the approach. The algorithm proposed in [19], a modified form of the one in [22], as well as its modified form in [9], were later shown not to guarantee convergence [28]. Similarly, a counter example disproving convergence was developed in [5] for the cutting plane algorithms of [3,4]. Finally, the analogous algorithm of [17], based on a slightly modified form of the standard optimality condition, was also shown not to be always convergent [20]; besides, the modified optimality condition is not easier to check than the standard one.

In this paper, we consider the CDC problem relying on an alternative equivalent formulation based on a polar characterization of the constraint. We define a unified algorithmic framework for outer approximation type algorithms, which are based on an "oracle" for checking the global optimality conditions, and we study different sets of conditions which guarantee its convergence to an (approximated) optimal solution. As the oracle is the most computationally demanding part of the approach, we allow working with an *approximated oracle* which solves the related (nonconvex) optimization problem only approximately. Because of this, we briefly investigate approximate optimality conditions in order to derive bounds on the quality of the obtained solution. Our analysis identifies two main classes of approaches, which give rise to six different implementable algorithms, four of which can't be reduced to the original cutting plane algorithm by Tuy and its modifications.

The paper is organized as follows. In Sect. 2 the *polar based* reformulation of the CDC problem is introduced, and the well-known optimality conditions are recalled. In Sect. 3 we propose a notion of approximate oracle and we define corresponding approximate optimality conditions, investigating the relationships between the exact optimal value and the approximate optimal values. In Sect. 4 a thorough convergence analysis is carried out for the "abstract" unified algorithmic framework, and then six *different implementable* algorithms are proposed which fit within the framework. Finally, some conclusions are drawn in the last section.

2 The canonical DC problem

Throughout all the paper we focus on the CDC minimization problem

$$\text{(CDC)} \quad \min\{\mathrm{d}x \mid x \in \Omega \backslash \operatorname{int} C\}$$

where $\Omega \subseteq \mathbb{R}^n$ and $C \subseteq \mathbb{R}^n$ are full-dimensional closed convex sets, $\mathrm{d} \in \mathbb{R}^n$ and $\mathrm{d}x$ denotes the scalar product between d and the vector of variables $x \in \mathbb{R}^n$.

The assumption on the dimension of the constraining sets is not restrictive. In fact, if Ω is not full-dimensional, the problem can be easily reformulated in the (affine) space generated by Ω. If C is not full-dimensional, then we have $\operatorname{int} C = \emptyset$ and the problem is actually a convex minimization problem.

In order to avoid that (CDC) could be reduced to a convex minimization problem, we also suppose that the set C provides an essential constraint, i.e.

$$\min\{\mathrm{d}x \mid x \in \Omega\} < \min\{\mathrm{d}x \mid x \in \Omega \backslash \operatorname{int} C\}.$$

Relying on an appropriate translation, this assumption can be equivalently stated through the following two conditions

$$0 \in \operatorname{int} \Omega \cap \operatorname{int} C, \tag{1}$$

$$\mathrm{d}x > 0 \ \forall x \in \Omega \backslash \operatorname{int} C. \tag{2}$$

Therefore, we assume that (1) and (2) hold. Notice that these assumptions guarantee that any feasible solution $x \in \Omega \backslash C$ provides a better feasible solution taking the unique intersection between the segment with 0 and x as end points and the boundary of C, i.e. $x' \in \operatorname{bd}(C) \cap (0, x)$ satisfies $\mathrm{d}x' < \mathrm{d}x$ where $\operatorname{bd}(C)$ denotes the boundary of C. As a consequence, all optimal solutions to (CDC) belong to the boundary of C.

In order to guarantee the existence of optimal solutions, we may assume the boundedness of the level sets

$$D(\gamma) := \{x \in \Omega \mid \mathrm{d}x \leq \gamma\}$$

for the feasible values γ, i.e. those values $\gamma = \mathrm{d}x \geq \gamma^*$ for some $x \in \Omega \backslash \operatorname{int} C$, where

$$\gamma^* := \min\{\mathrm{d}x \mid x \in \Omega \backslash \operatorname{int} C\}.$$

Actually, such an assumption on the level sets is strictly related to the compactness of the reverse constraining set C as the following result shows.

Lemma 1 *Let γ be a feasible value.*

(i) *If C is compact, then so is $D(\gamma)$.*
(ii) *If $D(\gamma)$ is compact, then*

$$\gamma^* = \min\{\mathrm{d}x \mid x \in \Omega \backslash \mathit{int}\ \hat{C}\}$$

where $\hat{C} = C \cap B$ for any given compact set B such that $D(\gamma) \subseteq \mathit{int}\ B$.

Proof

(*i*) Assume by contradiction, suppose there exists a sequence $\{x^k\} \subseteq D(\gamma)$ such that $\|x^k\| \to +\infty$. Possibly taking a suitable subsequence, let $u = \lim_{k\to\infty} x^k \|x^k\|^{-1}$: clearly $\mathrm{d}u \leq 0$ and u belongs to the recession cone of Ω [16, Theorem 8.2]. Since $0 \in \Omega$ and C is bounded, there exists $\lambda > 0$ such that $x^0 = 0 + \lambda u \in \Omega \backslash C$. As $\mathrm{d}x^0 \leq 0$, assumption (2) is contradicted.

(*ii*) Let $\bar{\gamma} := \min\{\mathrm{d}x \mid x \in \Omega \backslash \mathrm{int}\, \hat{C}\}$. Since $\hat{C} \subseteq C$, then $\gamma^* \geq \bar{\gamma}$. Furthermore, $\gamma \geq \gamma^*$ and the compactness of $D(\gamma)$ guarantee the existence of $\bar{x} \in \Omega \backslash \mathrm{int}\, \hat{C}$ such that $\bar{\gamma} = \mathrm{d}\bar{x}$. As $\mathrm{int}\, \hat{C} = \mathrm{int}\, C \cap \mathrm{int}\, B$ and $\bar{x} \in D(\gamma)$, then $\bar{x} \notin \mathrm{int}\, C$: $\bar{x}$ is feasible to (CDC) and therefore $\gamma^* \leq \bar{\gamma}$. □

Therefore, we assume that C is compact throughout all the paper. Moreover, this compactness assumption ensures existence of an optimal solution x^*, and therefore due to (2) we have $\gamma^* = \mathrm{d}x^* > 0$, a property that will turn out to be very useful.

The level sets introduced above are also helpful to check whether a feasible value is optimal or not. In fact, it is straightforward that $\gamma = \gamma^*$ implies the following inclusion:

$$D(\gamma) \subseteq C. \tag{3}$$

Furthermore, it has been shown (see [29, Proposition 10]) that the necessary optimality condition (3) is also sufficient when problem (CDC) is *regular*, i.e.

$$\min\{\mathrm{d}x \mid x \in \Omega \backslash \mathrm{int}\, C\} = \inf\{\mathrm{d}x \mid x \in \Omega \backslash C\}. \tag{4}$$

The above regularity condition is strongly related to the existence of optimal solutions to (CDC) with additional properties (see Lemma 2). Furthermore, regularity can be exploited to prove that stopping criteria with finite tolerance yield approximate optimal solutions.

Lemma 2 *The regularity condition* (4) *holds if and only if (CDC) has an optimal solution* $x^* \in \mathit{bd}\,(\Omega \backslash C)$.

Proof Given any optimal solution $x^* \in \mathrm{bd}\,(\Omega \backslash C)$, there exists a sequence $\{x^k\}$ such that $x^k \in \Omega \backslash C$ and $x^k \to x^*$; hence

$$\inf\{\mathrm{d}x \mid x \in \Omega \backslash C\} \leq \lim_{k\to\infty} \mathrm{d}x^k = \mathrm{d}x^* = \min\{\mathrm{d}x \mid x \in \Omega \backslash \mathrm{int}\, C\}.$$

As the reverse inequality always holds, the regularity condition (4) follows.

Vice versa, suppose the regularity condition (4) holds. Therefore, there exists a sequence $\{x^k\} \subseteq \Omega \backslash C$ such that $\mathrm{d}x^k \downarrow \gamma^*$. By Lemma 1 the compactness of C guarantees that $D(\gamma)$ is compact for $\gamma = \mathrm{d}x^1$. Therefore, the sequence $\{x^k\}$ admits at least one cluster point $x^* \in \mathrm{cl}\,(\Omega \backslash C)$. Since Ω is closed and $x^k \notin C$ for all k, we have $x^* \in \Omega$ and $x^* \notin \mathrm{int}\, C$. This implies that x^* is feasible and hence optimal as $\mathrm{d}x^* = \gamma^*$. Since all optimal solutions belong to the boundary of C, then $x^* \notin \Omega \backslash C$ and therefore $x^* \in \mathrm{bd}\,(\Omega \backslash C)$. □

Figure 1 shows the case of a non-regular problem: the unique optimal solution x^* does not belong to $\mathrm{bd}\,(\Omega \backslash C)$, in accordance with Lemma 2, and the feasible value γ is not the optimal one though it satisfies (3).

The constraint $x \notin \mathrm{int}\, C$ is the source of nonconvexity in problem (CDC) and it is given just as a set relation. However, relying on the polarity between convex sets, we can express this nonconvex constraint in a different fashion. Let us recall that

$$C^* = \{w \in \mathbb{R}^n \mid wx \leq 1, \quad \forall x \in C\}$$

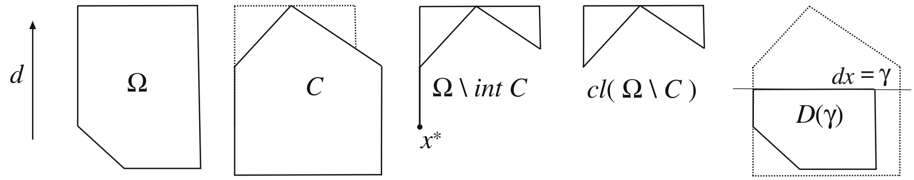

Fig. 1 Lack of regularity

is the polar set of C and it is a closed convex set. Exploiting bipolarity relations (see, for instance, [16]), it is easy to check that the assumption $0 \in \text{int } C$ ensures that $x \notin \text{int } C$ if and only if $wx \geq 1$ for some $w \in C^*$. Therefore, problem (CDC) can be equivalently formulated as

$$\min\{\mathrm{d}x \mid x \in \Omega, w \in C^*, wx \geq 1\} \tag{5}$$

where polar variables w have been introduced and the nonconvexity is given by the inequality constraint, which asks for some sort of reverse polar condition. Also, the assumption $0 \in \text{int} C$ ensures the compactness of C^*. The exploitation of polar variables will be an important tool to devise novel algorithms for (CDC) through its reformulation (5).

Relying on bipolarity relations, the optimality condition (3) can be equivalently stated in a polar fashion as

$$D(\gamma) \times C^* \subseteq \{(x, w) \in \mathbb{R}^n \times \mathbb{R}^n \mid wx \leq 1\} \tag{6}$$

while the regularity condition (4) reads

$$\min\{\mathrm{d}x \mid x \in \Omega, w \in C^*, wx \geq 1\} = \inf\{\mathrm{d}x \mid x \in \Omega, w \in C^*, wx > 1\}. \tag{7}$$

As an immediate consequence of (6), any optimal solution (x^*, w^*) to (5) satisfies both $x^* \in \text{bd}\,(C)$ and $w^*x^* = 1$. Since all the algorithms will be built to attain condition (6), in the following we always assume that condition (7) holds.

3 Approximate optimality conditions

Given a feasible value γ, the optimality condition (3) or (6) should be checked in order to recognize whether or not γ is the optimal value. Unfortunately, there is no known efficient way to check the inclusion between two sets. Yet, any exact algorithm for (CDC) or (5) must eventually cope with this problem.

3.1 Optimality conditions and (approximate) oracles

In order to make (3), or equivalently (6), more readily approachable, we consider the following "optimization version" of the optimality conditions:

$$\max\{vz - 1 \mid z \in D(\gamma), v \in C^*\}. \tag{8}$$

Obviously, (6) holds if and only if the optimal value $v(OC_\gamma)$ of (8) is nonpositive. Thus the above problem provides a way for checking optimality of a given value γ. Since the objective function of (8) is not concave, there are no known efficient approaches for this problem as well. However, checking (6) through the optimization problem (8) has the advantage of making it easy to define a proper notion of approximate optimality conditions.

A first way of approximating problem (8) is to replace $D(\gamma)$ and C^* with two convex sets S and Q, respectively, satisfying

$$C^* \subseteq Q, \tag{9}$$
$$D(\gamma) \subseteq S. \tag{10}$$

This is a standard step in cutting plane (outer approximation) approaches, where S and Q are chosen to be "easier" than the original sets (e.g. polyhedra with possibly few vertices or facets) and iteratively refined to become better and better approximations of $D(\gamma)$ and C^* as needed. Hence, one considers the following *relaxation* of (8):

$$\max\{vz - 1 \mid z \in S, v \in Q\}, \tag{11}$$

whose optimal value $v(\overline{OC}_\gamma)$ provides an upper bound on $v(OC_\gamma)$; thus, the inequality $v(\overline{OC}_\gamma) \leq 0$ provides a convenient *sufficient* optimality condition for (5). If it does not hold, then either γ is not the optimal value, or S and Q are not "good" approximations of $D(\gamma)$ and C^*, respectively.

All the cutting plane algorithms presented in this work follow the same basic scheme: (11) is solved and its solution is used to improve S, or Q, or γ, in such a way to guarantee convergence of γ to the optimal value. The focus of the research is on devising a number of different ways to achieve a convergent algorithm for (5) out of an "oracle" for (11). However, it is likely that in any such approach the solution of (11) is going to be the computational bottleneck. The problem could be easily solved if the vertices of S and Q were known, and "few", since an optimal solution $(\bar{z}, \bar{v})$ can surely be found such that $\bar{z}$ is a vertex of S and $\bar{v}$ is a vertex of Q. The problem would be easy even if only *one* of the two polyhedra had "few" vertices, since the computation of the best pairing for any given vertex only requires solving a linear program. This is, basically, the approach suggested in the literature for solving (11).

Unfortunately, as we will see, the number of inequalities in the description of S and Q may grow large during the solution process, and the number of vertices grows exponentially fast in the number of inequalities; therefore, it must be expected that the number of vertices rapidly becomes unmanageable, making exact solution to (11) by means of vertex enumeration techniques impractical. Therefore, it makes sense to consider solving (11) *only approximately*; this may actually mean two different things:

1. Computing a "large enough" lower bound on $v(\overline{OC}_\gamma)$, i.e. finding a feasible solution $(\bar{z}, \bar{v})$ of (11) "sufficiently close" to the optimal solution;
2. Computing a "small enough" upper bound $l \geq v(\overline{OC}_\gamma)$.

This is the rationale behind our definition of an *approximate oracle* for (11). In our development we will assume availability of a procedure Θ which, given S, Q, γ, and two positive tolerances ε and ε'

– *either* produces an upper bound

$$\varepsilon v(\overline{OC}_\gamma) \leq l \quad \text{such that} \quad l \leq \varepsilon' \tag{12}$$

– *or* produces a pair

$$(\bar{z}, \bar{v}) \in S \times Q \quad \text{such that} \quad \bar{v}\bar{z} - 1 \geq \varepsilon v(\overline{OC}_\gamma) > \varepsilon'. \tag{13}$$

We stress the fact that, for the sake of our approach, *only one* of the two conditions is actually needed at any given time; indeed, $v(\overline{OC}_\gamma)$ is either positive or non-positive, so we only need to prove whichever of the conditions happens to be true for the current value of γ.

Algorithmically, the two notions correspond to two entirely different classes of approaches. Lower bounds are produced by *heuristics* computing feasible solutions. In this context, effective heuristics exploiting the structure of (11) may be *alternating minimization* methods, whereby one of the two variables is kept fixed and a linear maximization problem is solved to optimize on the other, and then the role is reversed; iterating this procedure provably leads to a local optima of the problem [8], and this approach has been experimentally proven to be remarkably effective in several fields, such as machine learning [6] and image processing [34]. Upper bounds are produced instead by solving suitable *relaxations* of (11), i.e. by replacing the non-concave objective function vz with a suitable concave upper approximation. In particular, one may use well-known results that characterize the *concave envelope* (lower concave approximation) of the function, which happens to be polyhedral in this particular case [14]. If none of the two approaches in isolation is capable of satisfying the corresponding one among (12) and (13), *exact algorithms* combining the two can then be used to push the lower bound and the upper bound arbitrarily close together, until eventually one of the two conditions is realized. This can be obtained by iteratively partitioning the feasible space—typically on a variable-by-variable basis, which in this case has the advantage of keeping separability in the constraints set—and re-computing the bounds on each partition (e.g. [2,15] among the many).

Clearly, (13) corresponds to a pretty weak requirement about the way in which (11) is solved: a solution, which is optimal only with *fixed but arbitrary relative* tolerance $\varepsilon > 0$ and *absolute* tolerance ε', is required. Condition (12) allows the upper bound to be "small enough" but positive, rather than non-negative; this is taken as the stopping condition of the approach, and we will show that the positive tolerance allows for finite termination of the algorithms even when γ is optimal. The drawback is that a feasible value γ needn't be optimal when (12) holds: the next subsection is therefore devoted to the study of the relationships between the "quality" of γ and the tolerances ε and ε'.

3.2 Approximate optimality conditions

The stopping criterion (12) implies $v(OC_\gamma) \le \varepsilon'/\varepsilon$: the tolerances provide the upper bound $\delta = \varepsilon'/\varepsilon$ for the optimal value of (8). The values γ for which this upper bound holds are strictly related to the following approximated problem

$$\min\{dx \mid x \in \Omega, w \in C^*, wx \ge 1 + \delta\}, \tag{14}$$

which is obtained by perturbing the right-hand side of the nonconvex constraint in (5). Our analysis does not require any regularity assumption on (14) and it is based on the following value function

$$\phi(\delta) := \inf\{dx \mid x \in \Omega, w \in C^*, wx > 1 + \delta\},$$

or equivalently

$$\phi(\delta) = \sup\{\gamma \mid D(\gamma) \times C^* \subseteq \{(x, w) \in \mathbb{R}^n \times \mathbb{R}^n \mid wx \le 1 + \delta\}\}.$$

Proposition 1 *Let $\delta \ge 0$. Then, the following statements are equivalent:*

(i) $v(OC_\gamma) \le \delta$;
(ii) $D(\gamma) \times C^* \subseteq \{(x, w) \in \mathbb{R}^n \times \mathbb{R}^n \mid wx \le 1 + \delta\}$;
(iii) $\gamma \le \phi(\delta)$.

Proof The equivalence between (i) and (ii) follows immediately from the definition of $v(OC_\gamma)$. Analogously, (ii) implies (iii) by the definition of $\phi(\delta)$.

Suppose (ii) does not hold: there exist $x \in D(\gamma)$ and $w \in C^*$ such that $wx > 1+\delta$. Take any $t \in (0, 1)$ large enough to have $w(tx) > 1+\delta$. Since $0 \in \Omega$, the convexity of Ω implies $tx \in \Omega$; obviously $\mathrm{d}(tx) < \mathrm{d}x \leq \gamma$. Therefore, (tx, w) guarantees $\phi(\delta) < \gamma$ contradicting (iii). □

As (ii) with $\delta = 0$ reduces to the optimality condition for problem (5), inclusion (ii) can be considered as an approximate optimality condition for (5).

The stopping criterion (i) guarantees approximate optimality and condition (iii) provides the adequate tool to evaluate the quality of the approximation. In fact, the regularity assumption (7) reads $\gamma^* = \phi(0)$ and therefore

$$0 \leq \gamma - \gamma^* \leq \phi(\delta) - \phi(0)$$

holds for any feasible value γ which satisfies (i). If ϕ is Lipschitz, then for any feasible value γ satisfying the stopping criterion (12), i.e. (i) with $\delta = \varepsilon'/\varepsilon$, we have that $0 \leq \gamma - \gamma^* \leq M\varepsilon'/\varepsilon$, where M is the Lipschitz constant. The following results provide conditions which guarantee the Lipschitz behaviour of ϕ.

Theorem 1 [35, Theorem 2.2.1] *If there exists an optimal solution* (x^*, w^*) *to* (5) *such that* $T(\Omega, x^*) \nsubseteq T(C, x^*)$, *then* ϕ *is locally Lipschitz at 0* (*where* T *denotes the Bouligand tangent cone*).

Theorem 2 [35, Corollary 2.2.1] *If* C *is a polyhedron, then* ϕ *is locally Lipschitz at 0.*

4 Conditions and algorithms

In this section we present several algorithms which (approximately) solve (CDC) through its reformulation (5) if an approximated oracle Θ is available. We first establish a hierarchy of abstract conditions ensuring convergence; then, for each set of conditions we propose actual implementable procedures which realize it.

4.1 General convergence conditions

All the algorithms will follow the generic cutting plane scheme sketched in the previous section. More in details, a nonincreasing sequence of feasible values $\{\gamma^k\}$ is produced, and the oracle Θ is called for each γ^k, thereby producing either a value l^k such that condition (12) holds, or points z^k and v^k such that (13) are satisfied. By repeatedly calling the oracle, we can construct a procedure which either proves that γ^k satisfies condition (12) or produces a better feasible value $\gamma^{k+1} < \gamma^k$. In the latter case, γ^{k+1} is associated to (produced by) points x^k and w^k such that

$$x^k \in C, \quad w^k \in C^*, \quad w^k x^k = 1, \tag{15}$$

which implies also $(x^k, w^k) \in \mathrm{bd}(C) \times \mathrm{bd}(C^*)$. In fact, if $x^k \in \mathrm{int}\, C$ (analogously for $w^k \in \mathrm{int}\, C^*$), then $w^k x^k < \max\{w^k x \mid x \in C\} \leq 1$ (see [16, Theorem 13.1]). The rationale for (15) is that any optimal solution must satisfy these conditions.

It must be stressed that the above conditions do not require $x \in \Omega$ and therefore (x^k, w^k) may be infeasible for the polar reformulation (5). Anyway, (5) can be equivalently stated as

$$\min\{\zeta(w) \mid w \in C^*\} \tag{16}$$

where

$$\zeta(w) = \min\{\theta(x) \mid wx \geq 1\}$$

and

$$\theta(x) = \begin{cases} \mathrm{d}x & \text{if } x \in \Omega \\ +\infty & \text{otherwise.} \end{cases}$$

Therefore, the polar variable w^k is always feasible for (16), though it may be $\theta(x^k) = +\infty$. Since $\zeta(w) \leq \theta(x)$ for all pairs (x, w) satisfying (15), we can choose $\gamma^{k+1} = \zeta(w^k)$ whenever $x^k \notin \Omega$. As $\zeta(w^k)$ is the optimal value of a convex problem, it can be assumed to be efficiently computable. Moreover, if γ^{k+1} turns out to be optimal, then w^k is the "polar part" of an optimal solution: in fact any

$$\bar{x} \in \operatorname{argmin}\{\mathrm{d}x \mid x \in \Omega,\, w^k x \geq 1\}$$

provides the complementary x part of the optimal solution.

Thus, a given pair (x^k, w^k) can provide two (potentially) different feasible values: $\theta(x^k)$ which is essentially costless to compute but may be infinite, and $\zeta(w^k)$ whose computation requires the solution of a convex program. In general one may want to avoid the computation of $\zeta(w^k)$ unless it is strictly necessary; to allow a general treatment we will in the following indicate with $\gamma(x, w)$ a function taking a pair (x, w) satisfying (15) and producing a feasible value. Which of the two possible implementations is required will be discussed in the context of each implementable algorithm.

With the above notation, we can introduce the prototype of our algorithms.

Algorithm 1 Prototype algorithm

0. $\gamma^1 = +\infty$; $k = 1$;
1. If the optimality condition (3) holds, then γ^k is the optimal value: stop;
2. Select (x^k, w^k) satisfying (15) such that $\gamma^{k+1} = \gamma(x^k, w^k) < \gamma^k$;
   set $k = k + 1$; goto 1.

Clearly, if at Step 0 (initialization) some feasible pair (x^0, w^0) is known, one can alternatively set $\gamma^1 = \gamma(x^0, w^0)$. An important feature of Algorithm 1 is that $\{\gamma^k\}$ is a decreasing sequence bounded below:

$$0 \leq \lim_{k\to\infty} \gamma^k = \gamma^\infty < \cdots < \gamma^{k+1} < \gamma^k < \cdots < \gamma^1.$$

Therefore, $\{D(\gamma^k)\}$ is a "non-increasing" sequence of sets, i.e.

$$D(\gamma^\infty) \subseteq \cdots \subseteq D(\gamma^{k+1}) \subseteq D(\gamma^k) \subseteq \cdots \subseteq D(\gamma^1).$$

Obviously, Algorithm 1 is too general to deduce any meaningful property; something more has to be said:

1. How exactly the optimality condition (3) is checked.
2. How (x^k, w^k) such that $\gamma(x^k, w^k) < \gamma^k$ is selected once one knows that (3) is not fulfilled.

The two points are strictly interwoven: finding (x^k, w^k) such that $\gamma(x^k, w^k) < \gamma^k$ immediately proves that γ^k is not optimal; vice versa, assume that we have any constructive

procedure that produces a point $z^k \in D(\gamma^k)\backslash C$ when γ^k is not optimal: there exists $w^k \in C^*$ such that $w^k z^k > 1$ and $x^k = (w^k z^k)^{-1} z^k$ satisfies both $x^k \in D(\gamma^k)$ and $\gamma(x^k, w^k) \leq \mathrm{d}x^k < \mathrm{d}z^k \leq \gamma^k$.
Then, a first question is if such a method provides a convergent algorithm; not surprisingly, without further qualification the answer is negative.

Example 1 Consider (5) with $n = 2$, $d = (0, 1)$ and the sets

$$\Omega = \{x \in \mathbb{R}^2 \mid -1.8 \leq x_1 \leq 1.96, x_2 \geq -0.1\}, C = \{x \in \mathbb{R}^2 \mid x_1^2 + x_2^2 \leq 4\};$$

therefore, we have

$$C^* = \{w \in \mathbb{R}^2 \mid 4(w_1^2 + w_2^2) \leq 1\}.$$

Starting from any value $\gamma^1 > 0.87$ and applying the above procedure, we can find the sequences $z^k = (-1.8, \gamma^{k-1})$, $x^k = 2z^k/||z^k||$ and $w^k = z^k/2||z^k||$, which lead to a non-optimal solution $(x^\infty, w^\infty) \approx ((-1.8, 0.87), (-1.8, 0.87)/4)$, whereas the optimal solution is $(x^*, w^*) \approx ((1.96, 0.4), (1.96, 0.4)/4)$.

Thus, some care is needed in choosing the sequence w^k in Algorithm 1, as well as the accompanying sequences z^k and x^k if the mechanism illustrated above is to be used. Actually, our "more implementable" approximate optimality condition based on (8) indicates that a fourth sequence v^k, which "is to w^k what z^k is to x^k", should be taken into account as well. In fact, we propose the following general assumptions under which convergence can be proved:

$$v^k z^k - 1 \geq \varepsilon \max\{vz - 1 \mid (z, v) \in D(\gamma^k) \times C^*\} \tag{17}$$

$$\liminf_{k \to \infty} v^k z^k \leq 1 \tag{18}$$

where $\varepsilon \in (0, 1)$. Condition (17) basically says that v^k and z^k must be produced by some process attempting to solve the nonconvex problem (8) for $\gamma = \gamma^k$, although the process may be "terminated early" due to the optimality tolerance ε. Condition (18) rather requires the two sequences to be asymptotically jointly feasible, and, as we will see, there are several different implementable ways for ensuring that this holds. Anyway, as far as abstract conditions go, (17) and (18) are sufficient to guarantee convergence to the optimal value.

Proposition 2 *If conditions (17) and (18) hold, then the sequence of feasible values $\{\gamma^k\}$ in Algorithm 1 converges to the optimal value γ^*.*

Proof Since each γ^k is a feasible value, we have $\gamma^* \leq \gamma^\infty$, i.e. γ^∞ is a feasible value, too. Hence, (17) implies that

$$v^k z^k - 1 \geq \varepsilon \max\{vz - 1 \mid (z, v) \in D(\gamma^\infty) \times C^*\}$$

for all k. Taking the limit, (18) implies

$$\max\{vz - 1 \mid (z, v) \in D(\gamma^\infty) \times C^*\} \leq 0,$$

which guarantees (6) for $\gamma^* = \gamma^\infty$. Therefore, γ^∞ is the optimal value since the regularity condition (7) holds. □

When developing a "concrete" algorithm for (CDC), the abstract condition (18) shouldn't be directly imposed on the sequences $\{z^k\}$ and $\{v^k\}$. In fact, these are the results of a "complex" optimization process, i.e. approximately solving (8), upon which we want to impose as few conditions as possible, in order to leave as much freedom as possible to different implementations of this critical task. Therefore, we seek alternative ways for obtaining condition (18). One possibility is to rely on sequences of points x^k and w^k, which satisfy one of these pairs of conditions:

$$\begin{cases} \limsup_{k\to\infty} v^k(z^k - x^k) \le 0 & \text{(a)} \\ \limsup_{k\to\infty} v^k x^k \le 1 & \text{(b)}, \end{cases} \tag{19}$$

$$\begin{cases} \limsup_{k\to\infty} (v^k - w^k) z^k \le 0 & \text{(a)} \\ \limsup_{k\to\infty} w^k z^k \le 1 & \text{(b)}. \end{cases} \tag{20}$$

Both pairs of conditions clearly imply (18).

Lemma 3 *If either (19) or (20) hold, then (18) holds.*

Therefore, we can define the two sets of conditions which, separately, guarantee convergence of Algorithm 1:

$$B_1 \equiv (17) \wedge (19), \quad B_2 \equiv (17) \wedge (20).$$

Though they look highly symmetric to each other, we will show that B_1 and B_2 are by no means equivalent. In fact, we will propose several different sets of conditions (in particular, four for B_1 and two for B_2) which imply one of them, and develop implementable subprocedures that attain these conditions, thereby defining six different *implementable algorithms*.

4.2 The outer approximation machinery

As addressed in Sect. 3, one way to make (8) more tractable is to replace $D(\gamma)$ and C^* with two "simpler" convex sets Q and S such that $C^* \subseteq Q$ and $D(\gamma) \subseteq S$. Of course, this requires some appropriate machinery to update S and Q in order to make them "good enough" approximations of Ω and C^*. Convexity of both sets allows to rely on cutting procedures based on standard separation tools. In fact, the result below follows readily from the general Basic Outer Approximation Theorem [9, Theorem II.1].

Theorem 3 *Let r be a convex function such that $R = \{x \in \mathbb{R}^n \mid r(x) \le 0\}$ satisfies $0 \in int\, R$. Let $\{R^k\}$ be a sequence of convex sets and $\{x^k\}$ be a sequence of points which satisfy the following conditions:*

1. $x^k \in R^k \setminus R$
2. $R^{k+1} = R^k \cap \{x \in \mathbb{R}^n \mid p^k(x - y^k) + r(y^k) \le 0\}$ *where* $p^k \in \partial r(y^k)$ *for some* $y^k \in [0, x^k) \setminus int\, R$.

Then, any cluster point $\bar{x}$ of the sequence $\{x^k\}$ belongs to $\mathrm{bd}\,(R)$.

Theorem 3 suggests the standard cutting-plane procedure described in Subprocedure 1. It takes a "simple" representation S (typically a polyhedron) of the convex set R and a point x which proves the two are different, and it "improves" S to a representation of R which does not contain x (and still is a polyhedron if S is) by intersecting S with a separating hyperplane which cuts off x but no point in R. Due to Theorem 3, iterating this process leads, in the limit,

Subprocedure 1 Cutting-Plane subprocedure

Input: a closed convex set $R = \{x \in \mathbb{R}^n \mid r(x) \leq 0\}$ such that $0 \in \text{int}\, R$,
a closed convex set S such that $R \subseteq S$ and a point $x \in S \backslash R$.

1. Select a point $y \in (0, x) \cap \text{bd}\,(R)$ and a sub-gradient $p \in \partial r(y)$.
2. Set $S = S \cap \{x \mid p(x - y) + r(y) \leq 0\}$.

Output: S.

to a point in R; in other words, S becomes an "arbitrarily close" representation of R near a cluster point.

It is worth remarking that condition $0 \in \text{int}\, R$ is required to ensure that $y \neq x$, and therefore that the hyperplane actually separates R and x strictly. In our setting, the condition is satisfied for $D(\gamma)$: this is due to (1) and to the fact that $\gamma \geq \gamma^* > 0$, itself a consequence of the boundedness of C as discussed in Sect. 2. Boundedness of C is also equivalent to $0 \in \text{int}\, C^*$; therefore, the condition is *a fortiori* true for S and Q, the sets Subprocedure 1 will be called upon, due to (10) and (9), respectively.

4.3 A generic outer approximation subprocedure

We can now define a generic outer approximation procedure which, only provided with an approximate oracle Θ, allows implementations of Algorithm 1 which attain the convergence conditions introduced in Sect. 4.1. We call this a "generic" outer approximation procedure because it depends on two parameters: a selection rule Ψ for the x and w variables, and a stopping criterion Υ. In this subsection we will describe the properties of the subprocedure which are independent of the choices of Ψ and Υ; later on, we will show several different possible choices for these, leading to different implementable algorithms.

Subprocedure 2 Outer approximation subprocedure

Input: Q and S, closed convex sets satisfying (9) and (10), a feasible value γ.

0. $S^1 = S$; $Q^1 = Q$; $i = 1$;
1. Call the oracle Θ for S^i, Q^i, γ. If the oracle produces an upper bound l^i satisfying condition (12), then stop.
2. Otherwise, Θ produces (z^i, v^i) satisfying (13);
   Select (x^i, w^i) satisfying (15) and condition Ψ;
3. If $z^i \notin D(\gamma)$ then use Subprocedure 1 with $D(\gamma)$, S^i and z^i to get S^{i+1};
   else $S^{i+1} = S^i$;
4. If $v^i \notin C^*$ then use Subprocedure 1 with C^*, Q^i and v^i to get Q^{i+1};
   else $Q^{i+1} = Q^i$;
5. If stopping criterion Υ holds then stop.
   else $i = i + 1$; goto 1.

Output: Q^i and S^i; either l^i, or x^i, w^i, z^i, v^i.

Conditions (10) and (9) guarantee that $D(\gamma)$ and C^* are included in S^i and Q^i, respectively, for $i = 1$. The cutting-plane Subprocedure 1 ensures this is still true for any i and therefore we get the following "non-increasing" sequences of sets:

$$D(\gamma) \subseteq \cdots \subseteq S^{i+1} \subseteq S^i \subseteq \cdots \subseteq S^1,$$
$$C^* \subseteq \cdots \subseteq Q^{i+1} \subseteq Q^i \subseteq \cdots \subseteq Q^1.$$

We can now prove the basic properties of Subprocedure 2, which are independent of the choice of Ψ and Υ.

Lemma 4 *If Subprocedure 2 never ends, then all the cluster points of* $\{z^i\}$ *and* $\{v^i\}$ *belong to* $D(\gamma)$ *and* C^**, respectively.*

Proof Subprocedure 2 generates two sequences of points $\{z^i\}$ and $\{v^i\}$ such that $z^i \in S^i$, $v^i \in Q^i$, and the hypotheses of Theorem 3 are satisfied; hence, all the cluster points of $\{z^i\}$ and $\{v^i\}$ belong to $D(\gamma)$ and C^*, respectively. □

It will be crucial to ensure that the sequences $\{z^i\}$ and $\{v^i\}$ do indeed have cluster points. As both $D(\gamma)$ and C^* are assumed to be compact, it is very natural to suppose also that

$$\{z^i\} \text{ and } \{v^i\} \text{ are bounded.} \tag{21}$$

In fact, this condition holds, for instance, if S^1 and Q^1 are compact, which is not a restrictive assumption as $D(\gamma)$ and C^* are compact too. Therefore, from now onwards we suppose that (21) always holds. Note that the sequences $\{x^i\}$ and $\{w^i\}$ are always bounded as due to (15) they belong to bounded sets C and C^*, respectively.

Corollary 1 *If* $\varepsilon' > 0$*, and Subprocedure 2 never ends, then no cluster point of* $\{z^i\}$ *belongs to* C.

Proof By Lemma 4 all cluster points of $\{v^i\}$ belong to C^* and (21) guarantees that at least one exists. If there were a cluster point of $\{z^i\}$ in C, one would have that $\liminf_{i\to\infty} v^i z^i \leq 1$ in contradiction with $v^i z^i - 1 > \varepsilon'$, which is guaranteed by the oracle for any i [c.f. (13)]. □

Proposition 3 *If* $\varepsilon' > 0$ *and* $D(\gamma) \subseteq C$*, then Subprocedure 2 stops after a finite number of iterations.*

Proof Suppose Subprocedure 2 never ends; due to (21), the sequence $\{(z^i, v^i)\}$ has at least a cluster point which belongs to $D(\gamma) \times C^*$ by Lemma 4. Since $D(\gamma) \subseteq C$, then all the cluster points actually belong to $C \times C^*$: therefore, we have $\liminf_{i\to\infty} v^i z^i \leq 1$ which yields a contradiction as in Corollary 1. □

The above proofs show the need of requiring $\varepsilon' > 0$, since for $\varepsilon' = 0$ the subprocedure may never stop. In other words, Subprocedure 2 can not *finitely* prove that the optimal value is optimal; this is why it is relevant to clarify the relationship between approximated optimal values and the optimal value.

Finally, it is useful to remark that while condition (18) is characteristic of optimizing sequences, it holds for every fixed γ by substituting x^i to z^i, even if $\varepsilon' = 0$.

Lemma 5 *If Subprocedure 2 never ends, then* $\limsup_{i\to\infty} v^i x^i \leq 1$.

Proof Lemma 4 guarantees that all the cluster points of $\{v^i\}$ belong to C^*. Since $x^i \in C$ for all i, the thesis follows immediately. □

The subprocedure can then be used to define implementable versions of the prototype Algorithm 1.

Algorithm 2 Implementable outer approximation algorithm

0. $\gamma^1 = +\infty$; Select $S^1 \supseteq D(\gamma^1)$, $Q^1 \supseteq C^*$; $k = 1$;
1. Call Subprocedure 2 with S^k, Q^k, and γ^k;
2. If Subprocedure 2 stops at Step 1, then stop.
3. Set x^k, w^k, z^k and v^k as the output of Subprocedure 2;
4. Set Q^{k+1} and S^{k+1}, possibly using the output of Subprocedure 2;
5. Set $\gamma^{k+1} = \gamma(x^k, w^k)$; set $k = k + 1$; goto 1.

Some remarks on Algorithm 2 are in order:

- Since $D(\gamma^k) \subseteq S^k$ and $C^* \subseteq Q^k$, (13) guarantees that condition (17) is always satisfied by all possible variants of the algorithm, i.e. irrespective of the concrete choices for Ψ and Υ;
- At Step 4, the obvious possibility for Q^{k+1} and S^{k+1} is to set them as the sets produced by Subprocedure 2; this leads to accumulation in Q^k and S^k of *all* cutting planes generated along the iterates, and therefore possibly to “large” descriptions of Q^k and S^k;
- Which implementation of $\gamma(x^k, w^k)$ has to be chosen depends on the properties of the points x^k and w^k (see Table 1 in Sect. 4.6) and therefore ultimately on Ψ.

The following subsections are devoted to the study of which conditions Ψ and Υ result in a convergent Algorithm 2.

4.4 Algorithms exploiting the set of conditions B_1

While the oracle in Subprocedure 2 guarantees (17), condition (19) has to be achieved through additional properties. The algorithms of this subsection will require (19b) more or less directly and will obtain (19a) by imposing (20b) and one extra condition, which simply requires x^k and z^k to be collinear:

$$z^k = \mu_1^k x^k \qquad \text{for some } \mu_1^k > 0. \tag{22}$$

Lemma 6 *If (22) holds for all k, then (20b) implies (19a).*

Proof Due to (22) and $w^k x^k = 1$, (20b) reads $\limsup_{k\to\infty} \mu_1^k \leq 1$, thus we have

$$\limsup_{k\to\infty} v^k(z^k - x^k) = \limsup_{k\to\infty} (\mu_1^k - 1)v^k x^k.$$

Due to (17) and (22), we have $v^k x^k > 0$; therefore, the boundedness of $\{v^k\}$ and $\{x^k\}$ guarantees that the above lim sup is less or equal to 0. □

All algorithms in this subsection will exploit condition (22). Together with (15), this forces to choose $x^k \in \{\alpha z^k \mid \alpha \geq 0\} \cap \mathrm{bd}\,(C)$, thereby basically making the choice of x^k automatic once z^k is known. Note that the intersection is nonempty due to boundedness of C, and therefore x^k is always well defined.

The easiest way to guarantee that the sequences generated by Algorithm 2 satisfy (22) is to impose that z^i and x^i are always collinear in Subprocedure 2. Furthermore, this allows to prove that Subprocedure 2 either attains a decrease of the objective function or detects approximate optimality in a finite number of steps, provided that $dz^i \leq \gamma^k$.

Lemma 7 *Suppose*

$$S^k \subseteq \{z \in \mathbb{R}^n \mid dz \leq \gamma^k\} \tag{23}$$

and set

$$\Psi \equiv [z^i = \mu_1^i x^i \quad with \quad \mu_1^i > 0].$$

If $\varepsilon' > 0$ *and Subprocedure 2 never ends, then it produces iterates satisfying* $x^i \in (0, z^i) \cap \Omega$, $z^i \notin C$ *and* $\gamma(x^i, w^i) < \gamma$ *for sufficiently large* i.

Proof Lemma 4 guarantees that all the cluster points of $\{z^i\}$ and $\{v^i\}$ belong to $D(\gamma^k)$ and C^*, and Corollary 1 guarantees that each cluster point $\bar{z}$ of $\{z^i\}$ does not belong to C, therefore $\bar{z} \in \Omega \backslash C$. Thus, there exists $\bar{x} \in (0, \bar{z})$ such that $\bar{x}$ is a cluster point of $\{x^i\}$. By eventually taking the appropriate subsequences, suppose $z^i \to \bar{z}$ and $x^i \to \bar{x}$. All the above implies that $x^i \in (0, z^i)$ and $z^i \notin C$ for all sufficiently large i. Since $0 \in \text{int}\,\Omega$ and $\bar{z} \in \Omega$, we have also $\bar{x} \in \text{int}\,\Omega$ and therefore, $x^i \in \Omega$ for all sufficiently large i. Hence, we have $\gamma(x^i, w^i) \leq \mathrm{d}x^i < \mathrm{d}z^i \leq \gamma^k$ as $z^i \in S^i \subseteq S^1$. □

Condition (23) is actually a mild assumption on how S^k is updated in Algorithm 2: it is enough to keep the "objective cut" $dz \leq \gamma^k$ among the inequalities which define S^k and update it at each iteration to the current value of γ^k. Furthermore, this assumption implies that the membership test in Step 3 of Subprocedure 2 can be reduced to $z^i \notin \Omega$.

Some of the properties guaranteed by the above Lemma can be exploited in the stopping criterion Υ. Anyway, in order to guarantee that the decrease guaranteed by Subprocedure 2 under (22) is "sufficient", one has to prove also that the set of conditions B_1 is satisfied: this requires (19), which will be achieved through (19b) and (20b). In the next subsections we develop four different ways in which this can be done.

4.4.1 Algorithm C_1

The first possibility, directly inspired by the algorithms already proposed in the literature (see, for instance, [29]), is to resort to the following conditions:

$$\mathrm{d}z^k \leq \gamma^k, \tag{24}$$
$$x^k \in (0, z^k) \cap \Omega \cap \text{bd}\,(C). \tag{25}$$

Condition (25) implies (22) with $\mu_1^k > 1$. Actually, the two conditions are equivalent if $z^k \notin C$ and $x^k \in \Omega$ (since we always have $x^k \in \text{bd}\,(C)$); anyway we don't ask for these two conditions. As (25) guarantees that the sequence of points $\{x^k\}$ is feasible, we can set $\gamma(x^k, w^k) = \mathrm{d}x^k$.

Lemma 8 *If* $\gamma^* > 0$ *and (24), (25) hold for all k, then (20b) holds.*

Proof Since x^h is feasible, we have

$$\mathrm{d}x^0 - \sum_{k=1}^{h} (\mathrm{d}x^{k-1} - \mathrm{d}x^k) = \mathrm{d}x^h \geq \gamma^*$$

and therefore

$$\mathrm{d}x^0 - \gamma^* \geq \sum_{k=1}^{h} (\mathrm{d}x^{k-1} - \mathrm{d}x^k) \geq \sum_{k=1}^{h} (\mathrm{d}z^k - \mathrm{d}x^k)$$

where the last inequality holds since (24) reads $\mathrm{d}z^k \le \gamma^k = \mathrm{d}x^{k-1}$. Taking the limit, we get

$$\lim_{h\to+\infty} \sum_{k=1}^{h} (\mathrm{d}z^k - \mathrm{d}x^k) \le \mathrm{d}x^0 - \gamma^* < +\infty.$$

Since $\mu_1^k > 1$, (25) implies $\mathrm{d}z^k - \mathrm{d}x^k > 0$ and therefore we get $\mathrm{d}z^k - \mathrm{d}x^k = (\mu_1^k - 1)\mathrm{d}x^k \to 0$, which implies that $\lim_{k\to\infty} \mu_1^k = 1$ since the feasibility of x^k gives $\mathrm{d}x^k \ge \gamma^* > 0$. Therefore, we have

$$\limsup_{k\to\infty} w^k z^k = \limsup_{k\to\infty} \mu_1^k w^k x^k = \limsup_{k\to\infty} \mu_1^k \le 1$$

since (15) guarantees $w^k x^k = 1$. □

Therefore, we can define the following set of conditions

$$C_1 \equiv (17) \wedge (19b) \wedge (24) \wedge (25)$$

which implies B_1 and thus guarantees convergence for Algorithm 2. The proper choice of Ψ and Υ ensures that these conditions are finitely attained within Subprocedure 2 except (19b), which requires the knowledge of the entire sequences generated by Algorithm 2. Therefore, we consider a positive sequence $\sigma^k \to 0$ and ask for the subprocedure to provide points v^i and x^i such that

$$v^i x^i \le 1 + \sigma^k.$$

This condition can be considered an appropriate formulation of (19b) within Subprocedure 2 as in this way Algorithm 2 will surely satisfy (19b).

Proposition 4 *Suppose that (23) holds and set*

$$\Psi \equiv [z^i = \mu_1^i x^i \quad with \quad \mu_1^i > 0], \qquad \Upsilon \equiv [x^i \in \Omega] \wedge [v^i x^i \le 1 + \sigma^k].$$

If $\varepsilon' > \sigma^k > 0$, then Subprocedure 2 ends in a finite number of steps; if it stops at Step 5, it reports points x^i, w^i, z^i and v^i satisfying the set of conditions C_1.

Proof Lemma 5 and 7 guarantee that the stopping criterion Υ will be satisfied for i large enough, independently from the choice of σ^k. Therefore, Subprocedure 2 ends in a finite number of steps. Suppose it ends at Step 5. The stopping criterion Υ directly guarantees (17), (19b) holds as all iterates satisfy (13), (24) follows immediately from the assumption on S^1 as $S^i \subseteq S^1$. Finally, the stopping criterion Υ and (13) allow to get

$$0 < v^i x^i \le 1 + \sigma^k < 1 + \varepsilon' \le v^i z^i = \mu_1^i v^i x^i$$

where the first inequality follows from Ψ and (13). This implies $\mu_1^i > 1$ and thus we have $z^i \notin C$. Therefore, $x^i \in (0, z^i) \cap \mathrm{bd}\,(C)$ and hence (25) holds since the stopping criterion Υ provides $x^i \in \Omega$. □

An illustration of Algorithm 2 under conditions C_1 is provided in Fig. 2. The oracle is called onto S^1 and Q^1 (dashed squares), which outer-approximate Ω and C^* (solid circles) respectively, yielding z^1 and v^1. Note that working with Q^1, an outer approximation of C^*, corresponds to considering the set difference between S^1 and the polar of Q^1 (oblique dashed square), an inner approximation of C. In the x (primal) space, the intersection between the line joining z^1 and 0 with $\mathrm{bd}\,(\Omega)$ provides a linear inequality (s^1) cutting off z^1, which can then be used to improve S^1; the same happens in the w (polar) space. Assuming that

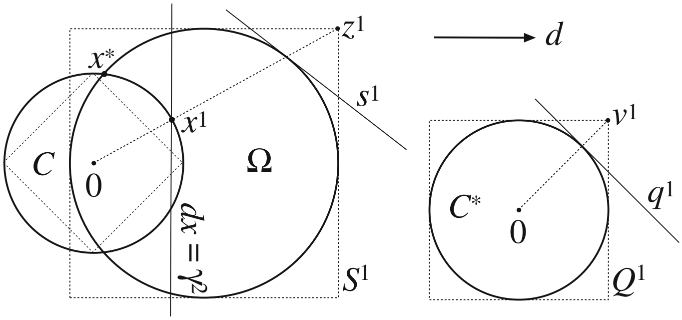

Fig. 2 Illustration of Algorithm 2 under conditions C_1

$v^1 x^1 \leq 1 + \sigma^1$, and therefore the stopping criterion Υ holds, the subprocedure ends providing the new feasible value γ^2, and the corresponding objective cut drives convergence of the algorithm by cutting off a substantial part of Ω.

For this algorithm to work, the sequence $\{\sigma^k\}$ has to be defined explicitly, either a-priori or dynamically as it is used to stop Subprocedure 2. Unlike most algorithms in the literature, it is not needed to require $\mu_1^i > 1$ at every iteration *within the subprocedure*, thus leaving a wider freedom of choice.

4.4.2 Algorithm C_2

An alternative way to obtain (19b) is to require

$$v^k x^h \leq 1 \quad \text{for all } h < k. \tag{26}$$

Lemma 9 *If* (26) *holds for all k, then* (19b) *holds.*

Proof Assume by contradiction that $v^k x^k > 1 + \delta$ for infinitely many k and a given $\delta > 0$. Since $\{v^k\}$ and $\{x^k\}$ are bounded, we can suppose $v^k \to \bar{v}$ and $x^k \to \bar{x}$ (eventually taking the appropriate subsequences). Condition (26) implies that $\bar{v}x^h \leq 1$ for all h and therefore $\bar{v}\bar{x} \leq 1$, a contradiction. □

Therefore, we can define the set of conditions

$$C_2 \equiv (17) \wedge (24) \wedge (25) \wedge (26)$$

which implies C_1 and therefore B_1, thus ensuring convergence for Algorithm 2.

Clearly, condition (26) is guaranteed if

$$Q^k \subseteq \bigcap_{h<k} \{v \in \mathbb{R}^n \mid vx^h \leq 1\}. \tag{27}$$

This can be easily achieved updating Q^{k+1} in Step 4 of Algorithm 2 as follows:

$$Q^{k+1} = Q^i \cap \{v \in \mathbb{R}^n \mid vx^i \leq 1\}, \tag{28}$$

where Q^i and x^i are those produced at the end Subprocedure 2.

Lemma 10 *If (28) holds, then $C^* \subseteq Q^{k+1}$.*

Proof Subprocedure 2 guarantees $C^* \subseteq Q^i$. If we consider the support function of C, namely

$$\sigma_C(v) := \max\{vx \mid x \in C\},$$

then we have

$$C^* = \{v \in \mathbb{R}^n \mid \sigma_C(v) - 1 \leq 0\}.$$

Since (15) guarantees $x^i \in C$, any $v \in C^*$ satisfies $vx^i \leq \sigma_C(v) \leq 1$. □

In this way all the inequalities produced by the Subprocedure 2 are kept: the "quality" of Q^{k+1} may improve, reducing the number of iterations required to stop the subprocedure, but it is likely to increase the cost of each iteration; the practical impact of this trade-off could be gauged only experimentally. In any case, in (28) it is always possible to replace Q^i with Q^k or any intermediate Q^j produced by the subprocedure since they both contain C^*.

Again, an implementable version of the Algorithm 2 can be obtained by choosing Ψ and Υ properly.

Proposition 5 *Set*

$$\Psi \equiv [z^i = \mu_1^i x^i \quad with \quad \mu_1^i > 0], \qquad \Upsilon \equiv [x^i \in \Omega] \wedge [z^i \notin C].$$

If $\varepsilon' > 0$ and (27) holds, then Subprocedure 2 ends in a finite number of steps; if it stops at Step 5, it reports points x^i, w^i, z^i and v^i satisfying the set of conditions C_2.

Proof Analogous to that of Proposition 4, considering that (26) follows from (27) and that $x^i \in \Omega$ and $z^i \notin C$ imply (25). □

The effect of (28) is depicted in (the polar part of) Fig. 3, relative to the same instance of Fig. 2. Exploiting polarity relationships, x^1 provides another inequality in the polar space cutting off v^1, that can be used to improve Q^1. This somewhat strengthens the convergence properties of the algorithm, doing away with the need for the sequence $\{\sigma^k\}$, but at the potential cost of requiring a large number of inequalities to be kept.

4.4.3 Algorithm C_3

Lemma 9 states that condition (19b) is implied by condition (26) under our boundedness assumptions. Symmetrically, we can prove the following result in the same way.

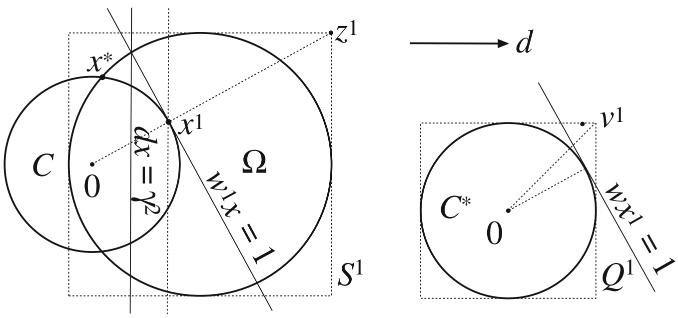

Fig. 3 Illustration of Algorithm 2 under conditions C_2 and C_3

Lemma 11 *If*

$$z^k w^h \leq 1 \quad \textit{for all } h < k \tag{29}$$

hold for all k, then (20*b*) *holds.*

Therefore, we can define the set of conditions

$$C_3 \equiv (17) \wedge (19b) \wedge (22) \wedge (29)$$

which implies B_1 (and thus guarantees convergence for Algorithm 2) as (22) and (29) imply (19a) by combining Lemmas 6 and 11.

Clearly, (29) is guaranteed if

$$S^k \subseteq \bigcap_{h<k} \{z \in \mathbb{R}^n \mid w^h z \leq 1\}. \tag{30}$$

This is easily obtained, for instance, by implementing Step 4 of Algorithm 2 as

$$S^{k+1} = S^i \cap \{z \in \mathbb{R}^n \mid w^i z \leq 1\} \tag{31}$$

where S^i and w^i are those produced at the end Subprocedure 2, but it is always possible to replace S^i with S^k or any intermediate S^j produced by the subprocedure. Anyway, the current value has to be updated through ζ in order to guarantee that S^{k+1} outer approximates $D(\gamma^{k+1})$.

Lemma 12 *Suppose* $\gamma(x, w) = \zeta(w)$. *If* (31) *is used in Algorithm 2, then* $D(\gamma^k) \subseteq S^k$ *for all k.*

Proof The proof is by induction on the iterate index k. If $k = 1$, the thesis is guaranteed by the choice of the input data. Suppose the thesis holds for a given k and there exists $\bar{x} \in D(\gamma^{k+1})$ such that $\bar{x} \notin S^{k+1}$: we have

$$\bar{x} \in D(\gamma^{k+1}) \subseteq D(\gamma^k) \subseteq S^i$$

where the last inclusion is guaranteed by the way Subprocedure 2 updates S^k. Therefore, (31) implies $w^i\bar{x} > 1$. Since $\bar{x} \in \Omega$, then $\hat{x} := (w^i\bar{x})^{-1}\bar{x} \in \Omega$ (as $w^i\bar{x} > 1$ and $0 \in \Omega$). Moreover, $w^i\hat{x} = 1$ and therefore $\gamma^{k+1} \leq \mathrm{d}\hat{x} < \mathrm{d}\bar{x}$ providing the contradiction $\bar{x} \notin D(\gamma^{k+1})$. □

Again, an implementable version of Algorithm 2 can be obtained by choosing Ψ and Υ properly. Note that the correctness of this version requires $\gamma(x, w) = \zeta(w)$; besides, there is no guarantee that x^k is feasible.

Proposition 6 *Set*

$$\Psi \equiv [z^i = \mu_1^i x^i \quad with \quad \mu_1^i > 0], \qquad \Upsilon \equiv [\zeta(w^i) < \gamma^k] \wedge [v^i x^i \leq 1 + \sigma^k],$$

If $\varepsilon', \sigma^k > 0$ *and* (30) *holds, then Subprocedure 2 ends in a finite number of steps; if it stops at Step 5, it reports points* x^i, w^i, z^i *and* v^i *satisfying the set of conditions* C_3.

Proof Analogous to that of Proposition 4, considering that (22) comes by Ψ and that (29) is implied by (30). □

Like Algorithm C_1, one has to use a sequence σ^k converging to zero explicitly; in this case, however, it is *not* required $\sigma^k < \varepsilon'$, at least initially.

The impact of (31) is illustrated in (the primal part of) Fig. 3; symmetrically to C_2, w^1 provides another inequality in the primal space cutting off z^1, that can be "significantly deeper" than s^1 (cf. Fig. 2), as it does not need to be valid for the whole Ω, but only for $D(\gamma^2)$. Besides, $\zeta(w^1) < \theta(x^1)$ in this case, therefore improving the convergence speed of the algorithm, albeit at the cost of the solution of a further convex program.

4.4.4 Algorithm C_4

The sets of conditions C_2 and C_3 are two independent modifications of C_1; the specific update (28) for Q^{k+1} is exploited for the former, while the "symmetric" update (31) for S^{k+1} is exploited for the latter. The two modifications can be combined: the set of conditions

$$C_4 \equiv (17) \wedge (26) \wedge (22) \wedge (29)$$

implies B_1 thanks to Lemmas 6, 9, and 11 and thus ensuring convergence for Algorithm 2. The following result provides an implementable version of the algorithm.

Proposition 7 *Set*

$$\Psi \equiv [z^i = \mu_1^i x^i \quad with \quad \mu_1^i > 0], \qquad \Upsilon \equiv [\zeta(w^i) < \gamma^k].$$

If $\varepsilon' > 0$, (27), and (30) hold, then Subprocedure 2 ends in a finite number of steps; if it stops at Step 5, it reports points x^i, w^i, z^i and v^i satisfying the set of conditions C_4.

4.5 Algorithms exploiting the set of conditions B_2

The algorithms of this subsection need (20) instead of (19). As (20b) has been exploited to achieve (19a), symmetrically (20a) can be obtained through (19b), relying on the "polar counterpart" of (22), namely

$$v^k = \mu_2^k w^k \qquad \text{for some} \quad \mu_2^k > 0. \tag{32}$$

Together with (15), this forces to choose $w^k \in \{\alpha v^k \mid \alpha \geq 0\} \cap \mathrm{bd}\,(C^*)$, thereby basically fixing w^k once v^k is known. Note that this intersection is always nonempty since C^* is compact.

Lemma 13 *If (32) holds for all k, then (19b) implies (20a).*

Proof Due to (32) and $w^k x^k = 1$, (19b) reads $\limsup_{k\to\infty} \mu_2^k \leq 1$, thus we have

$$\limsup_{k\to\infty}(v^k - w^k)z^k = \limsup_{k\to\infty}(\mu_2^k - 1)w^k z^k \leq 0$$

Due to (17) and (32), we have $w^k z^k > 0$; therefore, the boundedness of $\{z^k\}$ and $\{w^k\}$ guarantees that the above lim sup is less or equal to 0. □

The algorithms of this subsection will exploit (32). The easiest way to guarantee that the sequences generated by Algorithm 2 satisfy it is to impose that w^i and v^i are always collinear in Subprocedure 2.

Lemma 14 *Suppose (23) holds and set*

$$\Psi \equiv [v^i = \mu_2^i w^i \quad with \quad \mu_2^i > 0].$$

If $\varepsilon' > 0$ and Subprocedure 2 never ends, then it produces iterates satisfying $\zeta(w^i) < \gamma$ for sufficiently large i.

Proof Taking the appropriate subsequences, we can suppose $w^i \to \bar{w}$, $v^i \to \bar{v}$ and $z^i \to \bar{z}$. The collinearly assumption Ψ implies that $\bar{v} = \bar{\mu}\bar{w}$ for some $\bar{\mu} \geq 0$ and condition (13) guarantees $\bar{\mu} \neq 0$. Lemma 4 guarantees $\bar{v} \in C^*$; since $w^i \in \mathrm{bd}\,(C^*)$, we have $\bar{w} \in \mathrm{bd}\,(C^*)$ and thus $\bar{\mu} \in (0, 1]$. Therefore, we have

$$\lim_{i\to\infty} w^i z^i = \bar{w}\bar{z} = \bar{\mu}^{-1}\bar{v}\bar{z} \geq \lim_{i\to\infty} v^i z^i \geq 1 + \varepsilon'.$$

where the last inequality is due to (13). Therefore, $w^i z^i \geq 1 + \varepsilon'/2$ holds for all sufficiently large i. By Lemma 4 we have $\bar{z} \in \Omega$; since $0 \in \mathrm{int}\,\Omega$, we get $\bar{z}^i := (1+\varepsilon'/2)^{-1} z^i \in \Omega$ for all sufficiently large i. Hence, we have $\zeta(w^i) \leq \mathrm{d}\bar{z}^i < \mathrm{d}z^i \leq \gamma^k$ as $w^i \bar{z}^i \geq 1$ and $z^i \in S^i \subseteq S^1$. □

Using the above results, we can develop versions of Algorithm 2, which are "symmetric" to those that rely on the set of conditions B_1. However, the polar reformulation (5) is asymmetric in the sense that only the "original" variables x appear in the objective function. Therefore, only two of those four algorithms can be mirrored in this case. Specifically, we will develop sets of conditions D_1 and D_2 corresponding to C_3 and C_4, respectively. No algorithms corresponding to C_1 and C_2 can be devised since they should exploit the condition

$$w^k \in (0, v^k) \cap C^* \cap \mathrm{bd}\,(\Omega^*),$$

which is "symmetric" to (25). However, it would imply the existence of an optimal solution (x^*, w^*) such that $w^* \in C^* \cap \mathrm{bd}\,(\Omega^*)$, which is not necessarily true: if you consider (5) with $n = 1$, $d = 1$ and $\Omega = C^* = [-1/2, 4]$, the unique optimal point is $(x^*, w^*) = (1/4, 4)$ while $C^* \cap \mathrm{bd}\,(\Omega^*) = \{1/4\}$.

4.5.1 Algorithm D_1

We can define the set of conditions

$$D_1 \equiv (17) \wedge (19b) \wedge (29) \wedge (32)$$

in a "symmetric" way with respect to C_3. Due to Lemmas 11 and 13, D_1 implies B_2 and therefore it ensures convergence for Algorithm 2. An implementable version can be obtained by choosing Ψ and Υ as follows.

Proposition 8 *Set*

$$\Psi \equiv [v^i = \mu_2^i w^i \quad with \quad \mu_2^i > 0], \qquad \Upsilon \equiv [\zeta(w^i) < \gamma^k] \wedge [v^i x^i \leq 1 + \sigma^k],$$

If $\varepsilon', \sigma^k > 0$ and (30) holds, then Subprocedure 2 ends in a finite number of steps; if it stops at Step 5, it reports points x^i, w^i, z^i and v^i satisfying the set of conditions D_1.

4.5.2 Algorithm D_2

We can define the set of conditions

$$D_2 \equiv (17) \wedge (26) \wedge (29) \wedge (32)$$

in a "symmetric" way with respect to C_4. Due to Lemmas 9, 11 and 13, D_2 implies B_2 and therefore it ensures convergence for Algorithm 2. An implementable version can be obtained by choosing Ψ and Υ as follows.

Proposition 9 *Set*

$$\Psi \equiv [v^i = \mu_2^i w^i \quad with \quad \mu_2^i > 0], \quad \Upsilon \equiv [\zeta(w^i) < \gamma^k].$$

If $\varepsilon' > 0$, *(27) and (30) hold, then Subprocedure 2 ends in a finite number of steps; if it stops at Step 5, it reports points* x^i, w^i, z^i *and* v^i *satisfying the set of conditions* D_2.

4.6 Summary

We have developed six different implementable versions of Algorithm 2: while they are all based on Subprocedure 2, they differ for the stopping criterion Ψ, the condition Υ on the iterations, how the evaluation function γ is implemented and how S^k and Q^k are updated. All the considered variants are summarized in Table 1.

Now, we want to show that all these algorithms are indeed different, in the sense that they can produce different optimizing sequences even if the same instance and the same starting conditions are given. To this aim, we consider problem (CDC) with $n = 2$, $d = (0, 1)$ and

$$\Omega = \{x \in \mathbb{R}^2 \mid -1 \leq x_1 \leq 2, -1 \leq x_2 \leq 5, 3x_1 - x_2 \leq 4\},$$
$$C = \{x \in \mathbb{R}^2 \mid x_1^2 + x_2^2 \leq 4\}.$$

Notice that Ω is a bounded polyhedron, whose vertices provide the alternative description

$$\Omega = \text{conv}\,(\{(1, -1), (-1, -1), (-1, 5), (2, 5), (2, 2)\}).$$

It is easy to check that the unique optimal solution is the intersection between the segment $[(1, -1), (2, 2)]$ (the boundary of the constraint $3x_1 - x_2 \leq 4$) and the boundary of C, namely the point $x^* = (6 + \sqrt{6}, 3\sqrt{6} - 2)/5 \in \Omega \backslash \text{int}\, C$. Therefore, the optimal value is $\gamma^* = (3\sqrt{6} - 2)/5 \approx 1.0697$. Note that all standard assumptions are satisfied: (1) and (2) hold, C is compact while regularity follows from Lemma 2. Furthermore, the value function

Table 1 Summary of implementable versions of Algorithm 2

	Ψ	Υ	γ	Q^k	S^k
C_1	$z^i = \mu_1^i x^i, \mu_1^i > 0$	$x^i \in \Omega \wedge v^i x^i \leq 1 + \sigma^k$	θ		
C_2	$z^i = \mu_1^i x^i, \mu_1^i > 0$	$x^i \in \Omega \wedge z^i \notin C$	θ	(28)	
C_3	$z^i = \mu_1^i x^i, \mu_1^i > 0$	$\zeta(w^i) < \gamma^k \wedge v^i x^i \leq 1 + \sigma^k$	ζ		(31)
C_4	$z^i = \mu_1^i x^i, \mu_1^i > 0$	$\zeta(w^i) < \gamma^k$	ζ	(28)	(31)
D_1	$v^i = \mu_2^i w^i, \mu_2^i > 0$	$\zeta(w^i) < \gamma^k \wedge v^i x^i \leq 1 + \sigma^k$	ζ		(31)
D_2	$v^i = \mu_2^i w^i, \mu_2^i > 0$	$\zeta(w^i) < \gamma^k$	ζ	(28)	(31)

ϕ is locally Lipschitz at 0, as $(0, \delta) \in T(\Omega, x^*)$ and $(0, \delta) \notin T(C, x^*)$ for any $\delta > 0$ (see Theorem 1).

Considering the polar reformulation (5), we have

$$C^* = \{w \in \mathbb{R}^2 \mid 4(w_1^2 + w_2^2) \leq 1\}.$$

Since any optimal solution of (5) must satisfy $w^* x^* = 1$ and $w^* \in \mathrm{bd}\,(C^*)$, we have that $w^* = (6 + \sqrt{6}, 3\sqrt{6} - 2)/20$ provides the only possibility for the polar part of the optimal solution.

In the following, we assume the oracle Θ to always choose the same (z, v) when S, Q and γ are the same; furthermore, we set $\varepsilon = 1$ so that the pairs (z, v) satisfying (13) must actually be optimal for (11). In this way, we eliminate the nondeterminism due to the fact that the oracle may return different $\varepsilon-$optimal solutions of (11), which may be "many" especially if $\varepsilon \ll 1$; nonetheless, the six algorithms all construct different optimizing sequences for this instance.

Consider the following starting situation:

$$\begin{aligned} \sigma^1 &= 0.1, \quad \gamma^1 = +\infty, \quad Q^1 = [-1/2, 1/2] \times [-1/2, 1/2], \\ S^1 &= \{x \in \mathbb{R}^2 \mid -1 \leq x_1 \leq 2, -1 \leq x_2 \leq 10, 3x_1 - x_2 \leq 4\} \\ &= \mathrm{conv}\;(\{(1, -1), (-1, -1), (-1, 10), (2, 10), (2, 2)\})\,. \end{aligned}$$

All algorithms start call Subprocedure 2 with S^1, Q^1 and γ^1 as input data. The oracle provides an optimal solution of the certificate problem

$$\max\{vz - 1 \mid (z, v) \in S^1 \times Q^1\},$$

which can be easily obtained comparing the value $\bar{v}\bar{z}$ for all pairs where $\bar{z}$ is an extreme point of S^1 and $\bar{v}$ is an extreme point of Q^1. In this case, the unique optimal solution turns out to be $(z^1, v^1) = ((2, 10), (1/2, 1/2))$ with optimal value $v(\overline{OC}_{\gamma^1}) = 5$; thus, according to our assumptions, this is the pair the oracle Θ returns for all algorithms.

Algorithms implementing the set of conditions B_1. The four algorithms C_1, C_2, C_3, and C_4 ask for x^i and z^i to be collinear. Due to (15) the only possible choice is $x^1 = (2, 10)/\sqrt{26}$; since we have both $z^1 \notin C$ and $x^1 \in \Omega$, then the point satisfies also the more restrictive condition (25). Due to (15) the only choice for the corresponding polar point is $w^1 = (1, 5)/\sqrt{104}$.

The subprocedure stops at this first iteration for algorithms C_2 and C_4, since we have $x^1 \in \Omega$, $z^1 \notin C$ and $\zeta(w^1) \leq \mathrm{d}x^1 < \gamma^1$. On the contrary, it does not stop for algorithms C_1 and C_3 since

$$v^1 x^1 = 6/\sqrt{26} \approx 1.1767 > 1 + \sigma^1.$$

In algorithm C_2 the subprocedure provides the new current value $\gamma^2 = \theta(x^1) = \mathrm{d}x^1 = 10/\sqrt{26} \approx 1.9612$ while in algorithm C_4 it provides γ^2 as

$$\zeta(w^1) = \min\{\mathrm{d}x \mid x \in \Omega, x_1 + 5x_2 \geq \sqrt{104}\}.$$

The optimal solution of the above linear program is $\bar{x}^1 = (10 + \sqrt{26}, 3\sqrt{26} - 2)/8$ and therefore the current value will be updated to

$$\gamma^2 = \zeta(w^1) = \mathrm{d}\bar{x}^1 = (3\sqrt{26} - 2)/8 \approx 1.6621 < 10/\sqrt{26}.$$

As for algorithms C_1 and C_3, the subprocedure performs one more iteration after the sets S^1 and Q^1 have been updated through Subprocedure 1 (since $z^1 \notin \Omega$ and $v^1 \notin C^*$):

$$S^2 = S^1 \cap \{(x \in \mathbb{R}^2 \mid x_2 \leq 5\} = \Omega,$$
$$Q^2 = Q^1 \cap \{w \in \mathbb{R}^2 \mid \sqrt{2}(w_1 + w_2) \leq 1\}.$$

At the second iteration of the subprocedure the oracle returns the (unique) optimal solution of the certificate problem

$$\max\{vz - 1 \mid (z, v) \in S^2 \times Q^2\},$$

which is $(z^2, v^2) = ((2, 5), (\sqrt{2}-1, 1)/2)$. Therefore, the collinearity condition Ψ and (15) imply $x^2 = (4, 10)/\sqrt{29}$ and $w^2 = (2, 5)/2\sqrt{29}$. Since $x^2 \in \Omega$, $\zeta(w^2) \leq \mathrm{d}x^2 < \gamma^1$ and

$$v^2 x^2 = (3 + 2\sqrt{2})/\sqrt{29} \approx 1.0823 \leq 1 + \sigma^1,$$

the subprocedure stops: algorithm C_1 selects $\gamma^2 = \theta(x^2) = \mathrm{d}x^2 = 10/\sqrt{29} \approx 1.6569$ while algorithm C_3 solves the linear program

$$\zeta(w^2) = \min\{\mathrm{d}x \mid x \in \Omega, 2x_1 + 5x_2 \geq 2\sqrt{29}\}$$

in order to get the point $\bar{x}^2 = (20 + 2\sqrt{29}, 6\sqrt{29} - 8)/17)$ and set $\gamma^2 = \zeta(w^2) = \mathrm{d}\bar{x}^2 = (6\sqrt{29} - 8)/17 \approx 1.4301$.

The four algorithms have all provided different values for γ^2 and therefore they are different from each other.

Algorithms implementing the set of conditions B_2. The algorithms D_1 and D_2 require w^i and v^i to be collinear. Due to (15) the only possible choice is $w^1 = (1, 1)/2\sqrt{2}$ and the corresponding point in the original space can be only $x^1 = (\sqrt{2}, \sqrt{2})$. The subprocedure stops at this first iteration for algorithm D_2, since we have $x^1 \in \Omega$ and therefore $\zeta(w^1) \leq dx^1 < \gamma^1$. On the contrary, it does not stop for algorithm D_1 since

$$v^1 x^1 = \sqrt{2} \approx 1.4142 > 1 + \sigma^1.$$

In algorithm D_2 the subprocedure provides the new current value γ^2 as

$$\zeta(w^1) = \min\{\mathrm{d}x \mid x \in \Omega, x_1 + x_2 \geq 2\sqrt{2}\} = (3 - \sqrt{2})/\sqrt{2} \approx 1.1213.$$

and the corresponding optimal solution $\bar{x}^1 = (1 + \sqrt{2}, 3 - \sqrt{2})/\sqrt{2}$ is the best achieved point. Since this value for γ^2 is different from all those seen so far, D_2 is yet another different algorithm.

In algorithm D_1 the subprocedure performs a second iteration after the sets S^1 and Q^1 have been updated exactly in the same way as in algorithms C_1 and C_3 (since z^1 and v^1 are indeed the same). Therefore, the oracle provides the same $z^2 = (2, 5)$ and $v^2 = (\sqrt{2} - 1, 1)/2$. Due to the collinearity condition Ψ and (15), we get $w^2 = (\sqrt{2} - 1, 1)/2\sqrt{4 - 2\sqrt{2}}$ and $x^2 = \sqrt{2 - \sqrt{2}}(1, 1 + \sqrt{2})$. Since

$$v^2 x^2 = \sqrt{4 - 2\sqrt{2}} \approx 1.0824 \leq 1 + \sigma^1$$

the subprocedure ends. The value it returns as γ^2 is

$$\zeta(w^2) = \min\{\mathrm{d}x \mid x \in \Omega, (\sqrt{2} - 1)x_1 + x_2 \geq 2\sqrt{4 - 2\sqrt{2}}\} \approx 1.4169.$$

and the corresponding optimal solution

$$\bar{x}^2 = \left(\frac{4 + 2\sqrt{4 - 2\sqrt{2}}}{2 + \sqrt{2}}, \frac{4 + 6\sqrt{4 - 2\sqrt{2}} - 4\sqrt{2}}{2 + \sqrt{2}} \right)$$

is the best achieved point. Once again, this value for γ^2 is different from all previous ones: all the six algorithms are different.

5 Comparisons and conclusions

All the convergent outer approximation algorithms recalled in Sect. 1 satisfy the set of conditions C_1 or C_2, and are special cases of those presented in this paper. Furthermore, it is basically given for granted that the "oracle" for checking the optimality conditions is realized through enumeration of vertices. The contributions of the present paper are therefore the following:

– The introduction of "approximate oracle" conditions (12)–(13), which are designed to allow for more sophisticated and efficient solution procedures, with respect to pure vertex enumeration, to tackle the problem of checking the optimality condition, arguably the computational bottleneck in this type of approaches.
– A study of the impact of approximations in the optimality conditions onto the quality of the approximately optimal solutions satisfying them.
– Full exploitation of the "primal-polar" formulation of the optimality conditions based on (8) in order to derive a very general hierarchy of conditions ensuring convergence.
– A general algorithmic scheme based on the developed hierarchy which gives rise to six different implementable algorithms, four of which (C_3, C_4, D_1 and D_2) do not seem to have previously been considered in the literature; each of these algorithms can generate an approximate optimal value in a finite number of steps, where the error can be managed and controlled.

It may be worth remarking that the "new" algorithms C_3, C_4, D_1 and D_2 all use $\gamma(x, w) = \zeta(w)$. This has been inspired by the reformulation of (CDC) as the *quasi-concave minimization* problem (16) already proposed in [32]. However, in that paper a "cut and split" method was used, that is entirely different from the outer approximation algorithms proposed in this paper. Indeed, that method belongs to the main other family of algorithms for canonical DC problems, that of *branch and bound* methods (see, for instance, [24–26]). So, this research has shown how concepts developed for one family of approaches can be useful even for an entirely unrelated one.

While this paper seems to offer a quite comprehensive convergence theory for "oracle-based" outer approximation algorithms for canonical DC programs, much still needs to be done before these algorithms become widely used and accepted as those based on the branch and bound paradigm. In particular, more work is needed to identify practically efficient ways to implement the oracle, at least on special types of canonical DC programs in which the sets Ω and C have some form of exploitable structure; this will be the focus of further research.

References

1. Al-Khayyal, F., Sherali, H.: On finitely terminating branch-and-bound algorithms for some global optimization problems. SIAM J. Optim. **10**, 1049–1057 (2000)

2. Androulakis, I., Maranas, C., Floudas, C.: α–BB: a global optimization method for general constrained nonconvex problems. J. Global Optim. **7**, 337–363 (1995)
3. Ben Saad, S.: A new cutting plane algorithm for a class of reverse convex 0–1 integer programs. In: Floudas, C.A., Pardalos, P.M. (eds.) Recent advances in global optimization, pp. 152–164. Princeton University Press, Princeton (1992)
4. Ben Saad, S., Jacobsen, S.E.: A level set algorithm for a class of reverse convex programs. Ann. Oper. Res. **25**, 19–42 (1990)
5. Ben Saad, S., Jacobsen, S.E.: Comments on a reverse convex programming algorithm. J. Global Optim. **5**, 95–96 (1994)
6. Chapelle, O., Sindhwani, V., Keerthi, S.S.: Optimization techniques for semi-supervised support vector machines. J. Mach. Learn. Res. **9**, 203–233 (2008)
7. Fulop, J.: A finite cutting plane method for solving linear programs with an additional reverse constraint. European J. Oper. Res. **44**, 395–409 (1990)
8. Grippo, L., Sciandrone, M.: On the convergence of the block nonlinear Gauss–Seidel method under convex constraints. Oper. Res. Lett. **26**, 127–136 (2000)
9. Horst, R., Tuy, H.: Global optimization. Springer, Berlin (1990)
10. Horst, R., Pardalos, P.M. (eds.): Handbook of global optimization. Kluwer, Dordrecht (1995)
11. Nghia, M.D., Hieu, N.D.: A method for solving reverse convex programming problems. Acta Math. Vietnam. **11**, 241–252 (1986)
12. Pham, D.T., El Bernoussi, S.: Numerical methods for solving a class of global nonconvex optimization problems. Int. Ser. Numer. Math. **87**, 97–132 (1989)
13. Pintér, J.D. (ed.): Global optimization: scientific and engineering case studies. Springer, Berlin (2006)
14. Rikun, A.D.: A convex envelope formula for multilinear functions. J. Global Optim. **10**, 425–437 (1997)
15. Ryoo, H., Sahinidis, N.: Global optimization of multiplicative programs. J. Global Optim. **26**, 387–418 (2003)
16. Rockafellar, R.T.: Convex analysis. Princeton University Press, Princeton (1970)
17. Strekalovsky, A.S., Tsevendorj, I.: Testing the $\mathbb{R}$-strategy for a reverse convex problem. J. Global Optim. **13**, 61–74 (1998)
18. Thach, P.T.: Convex programs with several additional reverse convex constraints. Acta Math. Vietnam. **10**, 35–57 (1985)
19. Thoai, N.V.: A modified version of Tuy's method for solving DC programming problems. Optimization **19**, 665–674 (1988)
20. Tuan, H.D.: Remarks on an algorithm for reverse convex programs. J. Global Optim. **16**, 295–297 (2000)
21. Tuy, H.: Global minimization of a difference of two convex functions. Math. Program. Stud. **30**, 150–182 (1987)
22. Tuy, H.: A general deterministic approach to global optimization via DC programming. In: Hiriart-Urruty, J.B. (ed.) FERMAT Days 85: mathematics for optimization, pp. 273–303. North-Holland, Amsterdam (1986)
23. Tuy, H.: Convex programs with an additional reverse convex constraint. J. Optim. Theory Appl. **52**, 463–486 (1987)
24. Tuy, H., Horst, R.: Convergence and restart in branch-and-bound algorithms for global optimization. Application to concave minimization and DC optimization problems. Math. Program. **41**, 161–183 (1988)
25. Tuy, H.: Normal conical algorithm for concave minimization over polytopes. Math. Program. **51**, 229–245 (1991)
26. Tuy, H.: Effect of the subdivision strategy on convergence and efficiency of some global optimization algorithms. J. Global Optim. **1**, 23–36 (1991)
27. Tuy, H.: On nonconvex optimization problems with separated nonconvex variables. J. Global Optim. **2**, 133–144 (1992)
28. Tuy, H.: Canonical DC programming problem: outer approximation methods revisited. Oper. Res. Lett. **18**, 99–106 (1995)
29. Tuy, H.: DC optimization: theory, methods and algorithms. In: Horst, R., Pardalos, P.M. (eds.) Handbook of global optimization, pp. 149–216. Kluwer, Dordrecht (1995)
30. Tuy, H.: Convex analysis and global optimization. Kluwer, Dordrecht (1998)
31. Tuy, H., Al-Khayyal, F.A.: Global optimization of a nonconvex single facility location problem by sequential unconstrained convex minimization. J. Global Optim. **2**, 61–71 (1992)
32. Tuy, H., Migdalas, A., Varbrand, P.: A quasiconcave minimization method for solving linear two-level programs. J. Global Optim. **4**, 243–263 (1994)
33. Tuy, H., Tam, B.T.: Polyhedral annexation vs outer approximation for the decomposition of monotonic quasiconcave minimization problems. Acta Math. Vietnam. **20**, 99–114 (1995)

34. Wen, Y.-W., Ng, M.K., Huang, Y.-M.: Efficient total variation minimization methods for color image restoration. IEEE Trans. Image Process. **17**, 2081–2088 (2008)
35. Zhang, Q.H.: Outer approximation algorithms for DC programs and beyond. PhD Thesis, Università di Pisa, Pisa. http://etd.adm.unipi.it/theses/available/etd-07022008-181656/unrestricted/Thesis.pdf (2008). Accessed 02 July 2008