



Extension of the LP-Newton method to conic programming problems via semi-infinite representation

Mirai Tanaka¹ · Takayuki Okuno²

Received: 2 February 2019 / Accepted: 7 April 2020 / Published online: 27 April 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

The LP-Newton method solves linear programming (LP) problems by repeatedly projecting a current point onto a certain relevant polytope. In this paper, we extend the algorithmic framework of the LP-Newton method to conic programming (CP) problems via a linear semi-infinite programming (LSIP) reformulation. In this extension, we produce a sequence by projection onto polyhedral cones constructed from LP problems obtained by finitely relaxing the LSIP problem equivalent to the CP problem. We show global convergence of the proposed algorithm under mild assumptions. To investigate its efficiency, we apply our proposed algorithm and a primal-dual interior-point method to second-order cone programming problems and compare the obtained results.

Keywords Conic programming · Semi-infinite programming · Adaptive polyhedral approximation · LP-Newton method

1 Introduction

In this paper, we consider the following conic programming problem (CP):

$$\begin{aligned} & \text{minimize} && c^\top x \\ & \text{subject to} && Ax = b, \\ & && x \in \mathcal{H}, \end{aligned} \tag{1}$$

✉ Mirai Tanaka
mirai@ism.ac.jp

Takayuki Okuno
takayuki.okuno.ks@riken.jp

¹ Department of Statistical Inference and Mathematics, The Institute of Statistical Mathematics, 10-3, Midori-cho, Tachikawa, Tokyo 190-8562, Japan

² RIKEN Center for Advanced Intelligence Project, 1-4-1 Nihonbashi, Chuo-ku, Tokyo 103-0027, Japan

where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$ are a given matrix and vectors, respectively, and $\mathcal{K} := \prod_{i=1}^p \mathcal{K}_i \subseteq \mathbb{R}^n$ is the Cartesian product of nonempty closed convex cones \mathcal{K}_i for $i = 1, \dots, p$. In what follows, we denote n_i by the dimension of \mathcal{K}_i , i.e., $\mathcal{K}_i \subseteq \mathbb{R}^{n_i}$. If $\mathcal{K} = \mathbb{R}_+^n$ (nonnegative orthant), then the CP problem (1) reduces to the linear programming (LP) problem of the standard form:

$$\begin{aligned} &\text{minimize} && c^\top x \\ &\text{subject to} && Ax = b, \\ &&& x \geq 0. \end{aligned} \tag{2}$$

More generally, if \mathcal{K} is a symmetric cone, the CP problem (1) is called a symmetric cone programming (SCP) problem. The SCP problem includes the second-order cone programming (SOCP) problems and the semidefinite programming (SDP) problems. The CP problem (1) is a very important optimization model because it has many practical applications in fields such as synthesis of filter and antennae arrays, structural design, stability analysis of mechanics, robust optimization, and relaxation of combinatorial optimization [1, 2, 13, 20, 21].

To solve the CP problem (1), many researchers have developed algorithms exploiting the geometrical or algebraic structure of the cone \mathcal{K} . For instance, we can find Newton-type methods such as primal-dual interior-point methods [16, 18, 20] and non-interior continuous methods along with complementarity functions [3, 7, 9], Chubanov-type algorithms [12, 15], and simplex-type algorithms [6, 17, 25]. These algorithms were originally carried over from LP. One popular extension from LP to SCP is based on the Jordan algebra [4], by which the LP and SCP problems can be handled in the same algebraic framework. Another approach is based on a semi-infinite reformulation of the conic constraint $x \in \mathcal{K}$. This approach can be applied to the CP problem (1) with \mathcal{K} being possibly nonsymmetric cone. By representing the conic constraint as the intersection of an infinite number of half-spaces, the CP problem (1) can be reformulated as the following linear semi-infinite programming (LSIP) problem with infinitely many linear inequality constraints as we mention below. To derive the semi-infinite reformulation of the CP problem (1), we define the dual cone \mathcal{K}_i^* of \mathcal{K}_i by $\mathcal{K}_i^* := \{s \in \mathbb{R}^{n_i} : x^\top s \geq 0 \ (x \in \mathcal{K}_i)\}$. In addition, we call a compact subset S of \mathbb{R}^{n_i} a *base* of \mathcal{K}_i^* when it satisfies

$$\{\alpha v : \alpha \in \mathbb{R}_+, v \in S\} = \mathcal{K}_i^*.$$

Note that for each cone, a base is not unique. In this paper, we denote a fixed base of \mathcal{K}_i^* by $\text{base}(\mathcal{K}_i^*)$. Using these notations, we can derive the following LSIP problem equivalent to the CP problem (1):

$$\begin{aligned} &\text{minimize} && c^\top x \\ &\text{subject to} && Ax = b, \\ &&& (v^i)^\top x^i \geq 0 \ (v^i \in \text{base}(\mathcal{K}_i^*); i = 1, \dots, p), \end{aligned} \tag{3}$$

where $x^i \in \mathbb{R}^{n_i}$ denotes the i -th block of x partitioned so that it matches the Cartesian structure of \mathcal{K} , i.e., $x = ((x^1)^\top, (x^2)^\top, \dots, (x^p)^\top)^\top \in \prod_{i=1}^p \mathbb{R}^{n_i}$. For example, if

we take a base of a second-order cone appropriately, this semi-infinite constraint can be simplified to $(1, (\bar{v}^i)^\top)x^i \geq 0$ ($\bar{v}^i \in \mathbb{R}^{n_i-1} : \|\bar{v}^i\| \leq 1; i = 1, \dots, p$) as we mention later. Hayashi et al. [6] tailored the dual-simplex method for LP problems to dual SOCP problems via their LSIP representation. For an overview of semi-infinite programming problems, we refer readers to the survey articles [8, 14].

Algorithms for solving LP problems include the simplex method, ellipsoid method, and interior-point method. Although the ellipsoid and interior-point methods are polynomial-time algorithms, the existence of a strongly polynomial-time algorithm for solving LPs remains an open problem. In an attempt to devise a strongly polynomial-time algorithm for LPs, Fujishige et al. [5] proposed the LP-Newton method for box-constrained LP problems, which have a box constraint $l \leq x \leq u$ instead of the nonnegativity constraint $x \geq 0$ in the LP problem (2). Kitahara et al. [10] extended it to the LP problem (2). This algorithm repeats projecting a current point onto a polytope arising from the feasible region and the computation of a supporting hyperplane. Numerical results in [5] suggest that the algorithm is promising. In addition, Kitahara and Sukegawa [11] proposed a simple projection algorithm for the box-constrained LP problems based on the LP-Newton method.

Recently, Silvestri and Reinelt [19] developed an LP-Newton method for SCP problems. To the best of our knowledge, this is the first extension of the LP-Newton method to SCP problems. In [19], the authors considered conic-box-constrained SCP problems, i.e., the CP problem (1) with $x \in \mathcal{K}$ replaced by a conic-box constraint $l \preceq x \preceq u$, which denotes $x - l, u - x \in \mathcal{K}$ with a symmetric cone \mathcal{K} . Their algorithm computes a projection onto a conic zonotope at each iteration, and they proposed a Frank–Wolfe-based inner algorithm for this computation. Nevertheless, the computation of the projection still appears to be difficult. In fact, their numerical results showed that the inner algorithm for obtaining the projection required a number of iterations, although the outer loop was repeated relatively few times.

In this paper, we propose a different type of LP-Newton method for the CP problem (1) with \mathcal{K} being possibly a nonsymmetric cone, which is based on the semi-infinite reformulation (3). In our approach, we construct a sequence of LPs by adaptively selecting finitely many constraints from the infinitely many constraints of the LSIP problem (3). To produce an iteration point, we compute a projection onto a polytope arising from a polyhedral approximation of \mathcal{K} , which can be realized by solving a convex quadratic programming (QP) problem.

The remainder of this paper is organized as follows. In Section 2, we describe our proposed LP-Newton method for the CP problem (1). In Section 3, we establish global convergence of the proposed algorithm under the boundedness assumption of the optimal set of the CP problem (1). In Section 4, we propose a primal-dual algorithm that generates a sequence in the dual space of the CP problem (1). We also show its global convergence to a pair of optimums of the CP problem (1) and its dual problem under Slater’s constraint qualification. In Section 5, we report numerical results of applying the proposed method to SOCPs to investigate its validity and effectiveness.

2 Primal algorithm

In this section, we extend the LP-Newton method for the LP problem (2) proposed by Kitahara et al. [10] to the CP problem (1). For simplicity, we use the following notation:

$$\bar{A} := \begin{pmatrix} c^\top \\ A \end{pmatrix} \in \mathbb{R}^{(1+m) \times n}, \quad L := \left\{ \begin{pmatrix} \gamma \\ b \end{pmatrix} : \gamma \in \mathbb{R} \right\} \subseteq \mathbb{R}^{1+m}, \quad (4)$$

and for some $V_i \subseteq \text{base}(\mathcal{K}_i^*)$ ($i = 1, \dots, p$),

$$V := \prod_{i=1}^p V_i, \quad \text{cone}(V)^* := \{x \in \mathbb{R}^n : (v^i)^\top x^i \geq 0 \ (v^i \in V_i; \ i = 1, \dots, p)\}.$$

Note that $\text{cone}(V)^*$ defined above is the dual cone of the conic hull of V , denoted by $\text{cone}(V)$, and hence especially $\text{cone}(\text{base}(\mathcal{K}_i^*))^* = (\mathcal{K}_i^*)^* = \mathcal{K}_i$ for each i . Moreover, we often use a MATLAB notation for a partitioned vector, i.e., we denote $(u^\top, v^\top)^\top$ by $(u; v)$ for any vectors u and v . In addition, we denote m -dimensional zero vector $(0; \dots; 0) \in \mathbb{R}^m$ by 0_m .

In the proposed algorithm, we construct sequences $\{x^{(k)}\} \subseteq \mathbb{R}^n$ and $\{V^{(k)}\} \subseteq \prod_{i=1}^p \text{base}(\mathcal{K}_i^*)$ such that $V^{(k)} := \prod_{i=1}^p V_i^{(k)}$ and $|V_i^{(k)}| < \infty$ ($i = 1, \dots, p$) for each $k \geq 1$. As more specifically described shortly, the sequence $\{x^{(k)}\}$ is produced by performing one iteration of the LP-Newton method for the following LP problem, which is obtained by replacing $\text{base}(\mathcal{K}^*)$ in the LSIP problem (3) by the finite set $V_i^{(k)}$ for $i = 1, \dots, p$:

$$\begin{aligned} &\text{minimize} && c^\top x \\ &\text{subject to} && Ax = b, \\ &&& (v^i)^\top x^i \geq 0 \ (v^i \in V_i^{(k)}; \ i = 1, \dots, p). \end{aligned} \quad (5)$$

Note that the finite number of linear inequality constraints of the LP problem (5) form an outer polyhedral approximation of \mathcal{K} . Thus, the LP problem (5) is a relaxation problem of the LSIP problem (3) or the CP problem (1). As will be proved later, if $x^{(k)}$ is feasible to the CP problem (1), it is nothing but an optimum of the CP problem (1).

Otherwise, for each i such that $x^{i,(k)} \notin \mathcal{K}_i$, we update $V_i^{(k)}$ by adding $v^{i,(k)} \in \text{argmin}_{v^i \in \text{base}(\mathcal{K}_i^*)} (v^i)^\top x^{i,(k)}$, which is one of the indices most violating the i -th conic constraint $x^i \in \mathcal{K}^{(i)}$ at $x^{(k)}$.

We name the proposed algorithm the *adaptive LP-Newton (ALPN) method* for the CP problem (1) and formally describe it in Algorithm 1. This algorithm generates sequences $\{w^{(k-1/2)}\}_{k \geq 1}$ on the line L and $\{w^{(k)}\}$ on the outside of $\bar{A}\mathcal{K}$ (recall the definition by (4)). For illustration, we give a schematic diagram in Fig. 1. At the k -th iteration of Algorithm 1, we first compute $w^{(k)}$ and $w^{(k+1/2)}$ from $w^{(k-1/2)}$ by applying one iteration of the existing LP-Newton method [10] developed for LPs to the LP problem (5). Specifically, we set the projection of $w^{(k-1/2)}$ onto $\bar{A}\mathcal{K}$ as $w^{(k)}$. Subsequently, we update an outer approximation $\bar{A}\text{cone}(V_i^{(k)})^*$ of $\bar{A}\mathcal{K}_i$ for each i by updating $V^{(k)}$ in the way explained in the previous paragraph. After that, we construct the support hyperplane $H^{(k)}$ of $\bar{A}\mathcal{K}$ at $w^{(k)}$ and set the intersection

of the plane $H^{(k)}$ and the line L as $w^{(k+1/2)}$. A sequence $\{\gamma^{(k-1/2)}\} \cup \{\gamma^{(k)}\} = \{\gamma^{(1/2)}, \gamma^{(1)}, \gamma^{(3/2)}, \gamma^{(2)}, \dots\}$ is nondecreasing and bounded above by the optimal value θ^* of the CP problem (1) as we show in Lemma 2. Moreover, it converges to θ^* under mild assumptions. Another sequence $\{\beta^{(k)}\}$ converges to b as we show in Lemma 3. In conjunction with these two sequences, a sequence $\{x^{(k)}\}$ converges to an optimal solution (Theorem 1).

Algorithm 1 Adaptive LP-Newton method for the CP problem (1).

- 1: Choose initial finite sets $V_i^{(1)} \subseteq \text{base}(\mathcal{K}_i^*)$ for $i = 1, \dots, p$ such that $\text{argmin}\{c^\top x : Ax = b, x \in \text{cone}(V^{(1)})^*\} \neq \emptyset$. \triangleright If no such $V_i^{(1)}$ exists for some i , the CP problem (1) is infeasible or unbounded.
 - 2: Choose an initial point $w^{(1/2)} := (\gamma^{(1/2)}, b) \in L$ with sufficiently small $\gamma^{(1/2)} \in \mathbb{R}$.
 - 3: **for** $k := 1, 2, \dots$ **do**
 - 4: Find the nearest point $w^{(k)} := (\gamma^{(k)}, \beta^{(k)}) \in \mathbb{R} \times \mathbb{R}^m$ of $w^{(k-1/2)}$ in $\bar{A}\text{cone}(V^{(k)})^* = \{\bar{A}x : x \in \text{cone}(V^{(k)})^*\}$ and obtain $x^{(k)}$ such that $\bar{A}x^{(k)} = w^{(k)}$ and $x^{(k)} \in \text{cone}(V^{(k)})^*$.
 - 5: **for** $i = 1, \dots, p$ **do**
 - 6: Find $v^{i,(k)} \in \text{argmin}_{v^i \in \text{base}(\mathcal{K}_i^*)} (v^i)^\top x^{i,(k)}$.
 - 7: **if** $(v^{i,(k)})^\top x^{i,(k)} < 0$ **then** \triangleright Otherwise, $x^{i,(k)}$ satisfies the i -th conic constraint $x^i \in \mathcal{K}_i$.
 - 8: Set $V_i^{(k+1)} := V_i^{(k)} \cup \{v^{i,(k)}\}$.
 - 9: **if** $\beta^{(k)} = b$ and $\gamma^{(k-1/2)} \leq \gamma^{(k)}$ **then** $\triangleright x^{(k)}$ is an optimal solution of the LP problem (5).
 - 10: **if** $V^{(k+1)} = V^{(k)}$ **then**
 - 11: **return** $x^{(k)}$. $\triangleright x^{(k)}$ is an optimal solution of the CP problem (1).
 - 12: **else**
 - 13: Set $w^{(k+1/2)} := (\gamma^{(k)}, b)$.
 - 14: **else if** $(\beta^{(k)} \neq b$ and $\gamma^{(k-1/2)} = \gamma^{(k)})$ or $\gamma^{(k-1/2)} > \gamma^{(k)}$ **then**
 - 15: **break** \triangleright The LP problem (5) is infeasible and hence so is the CP problem (1).
 - 16: **else**
 - 17: Let $w^{(k+1/2)} := (\gamma^{(k+1/2)}, b)$ be the intersecting point of L and $H^{(k)}$, where $H^{(k)}$ is the supporting hyperplane of $\bar{A}\text{cone}(V^{(k)})^*$ on $w^{(k)}$ orthogonal to $w^{(k-1/2)} - w^{(k)}$.
-

In Algorithm 1, it is easy to compute $\gamma^{(k+1/2)}$ as follows:

$$\gamma^{(k+1/2)} = \gamma^{(k)} - \frac{\|\beta^{(k)} - b\|^2}{\gamma^{(k-1/2)} - \gamma^{(k)}}.$$

In addition, it is also easy to compute

$$v^{i,(k)} \in \text{argmin}_{v^i \in \text{base}(\mathcal{K}_i^*)} (v^i)^\top x^{i,(k)} \tag{6}$$

for some particular cones as illustrated below.

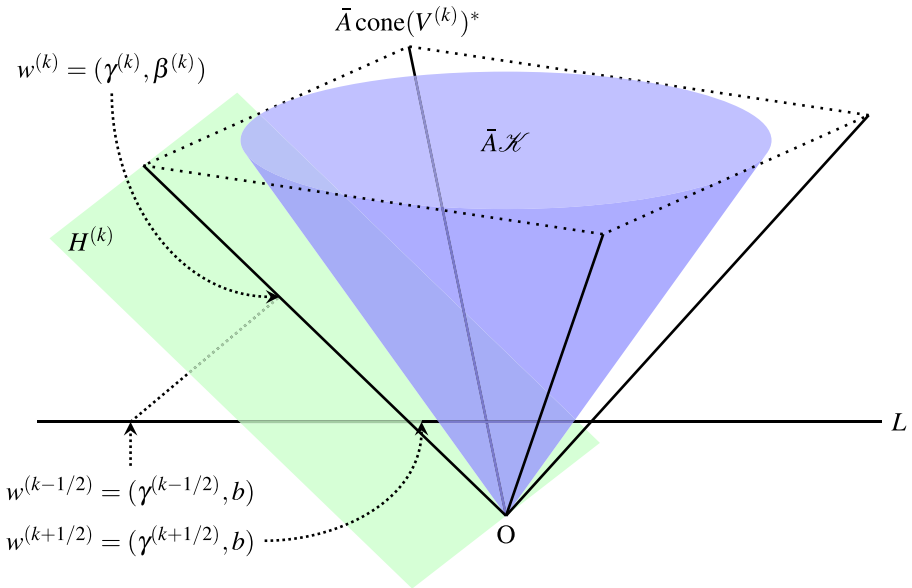


Fig. 1 A schematic diagram of the k -th iteration of Algorithm 1. The horizontal line depicts L , which is parallel to the γ -axis associated with the objective function. The circular cone (shown in blue) and the square pyramid (whose four ridges are shown in solid lines) depict $\bar{A}\mathcal{K}$ and $\bar{A}\text{cone}(V^{(k)})^*$, respectively. The line segment from $w^{(k-1/2)}$ to $w^{(k)}$ (shown in dotted line) is orthogonal to a supporting hyperplane $H^{(k)}$ (shown in green) of $\bar{A}\text{cone}(V^{(k)})^*$ at $w^{(k)}$. Note that $H^{(k)}$ touches $\bar{A}\text{cone}(V^{(k)})^*$ on a ridge. The intersecting point of L and $H^{(k)}$ is $w^{(k+1/2)}$

Example 1 (Second-order cone) For the n -dimensional second-order cone $\mathcal{K}_{\text{SOC}}^n \subset \mathbb{R}^n$ defined by $\mathcal{K}_{\text{SOC}}^n := \{x \in \mathbb{R}_+ \times \mathbb{R}^{n-1} : x_1 \geq \|\bar{x}\|\}$, where $\bar{x} \in \mathbb{R}^{n-1}$ denotes the subvector of x without the first element, that is, $x = (x_1; \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1}$, we can take $\{(s_1; \bar{s}) \in \mathcal{K}_{\text{SOC}}^n : s_1 = 1\}$ as $\text{base}((\mathcal{K}_{\text{SOC}}^n)^*) = \text{base}(\mathcal{K}_{\text{SOC}}^n)$. In addition, $(1; -\bar{x}^{i,(k)} / \|\bar{x}^{i,(k)}\|)$ is an optimal solution to the subproblem (6) with $\mathcal{K}_i^* = \mathcal{K}_{\text{SOC}}^n$ if $\bar{x}^{i,(k)} \neq 0_{n-1}$; otherwise, $(1; 0_{n-1})$ is optimal.

Example 2 (Positive semidefinite cone) The set of all n -dimensional positive semidefinite matrices forms a cone. The cone is called the n -dimensional positive semidefinite cone and denoted by $\mathcal{K}_{\text{PSD}}^n$. For $\mathcal{K}_{\text{PSD}}^n$, we can take $\{S \in \mathcal{K}_{\text{PSD}}^n : \text{trace}(S) = 1\}$ as $\text{base}((\mathcal{K}_{\text{PSD}}^n)^*) = \text{base}(\mathcal{K}_{\text{PSD}}^n)$. In addition, it holds that $\xi^*(\xi^*)^\top \in \text{argmin}_{V^i \in \text{base}(\mathcal{K}_i^*)} \text{trace}(V^i X^{i,(k)})$, which is the matrix version of the subproblem (6), where ξ^* is a normalized eigenvector of $X^{i,(k)}$ corresponding to the minimum eigenvalue.

Example 3 (ℓ_p -norm cone) For $p \geq 1$, the n -dimensional ℓ_p -norm cone $\mathcal{K}_{\ell_p}^n \subset \mathbb{R}^n$ is defined by $\mathcal{K}_{\ell_p}^n := \{x \in \mathbb{R}_+ \times \mathbb{R}^{n-1} : x_1 \geq \|\bar{x}\|_p\}$. The ℓ_p -norm cone is a generalization of the second-order cone. In fact, $\mathcal{K}_{\ell_2}^n = \mathcal{K}_{\text{SOC}}^n$. In addition, it holds

that $\mathcal{K}_{\ell_p}^* = \mathcal{K}_{\ell_q}$, where $q := (1 - p^{-1})^{-1}$. For this cone, we can take $\{(s_1; \bar{s}) \in \mathcal{K}_{\ell_q}^n : s_1 = 1\}$ as $\text{base}(\mathcal{K}_{\ell_q}^n)$ and an optimal solution to the subproblem (6) with $\mathcal{K}_i^* = \mathcal{K}_{\ell_q}^n$ can be written as $(1; -\text{sign}(\bar{x}^{i,(k)}) \circ (|\bar{x}^{i,(k)}|/\|\bar{x}^{i,(k)}\|_p)^{p/q})$ if $\bar{x}^{i,(k)} \neq 0_{n-1}$, where \circ denotes the Hadamard product and $(\text{sign}(\bar{x}^{i,(k)}))_j := 1$ ($\bar{x}_j^{i,(k)} > 0$), 0 ($\bar{x}_j^{i,(k)} = 0$), -1 ($\bar{x}_j^{i,(k)} < 0$) and $(|\bar{x}^{i,(k)}|)_j = |\bar{x}_j^{i,(k)}|$ for each j ; otherwise, $(1; 0_{n-1})$ is optimal.

Remark 1 Cones $\mathcal{K}_{\text{SOC}}^n, \mathcal{K}_{\text{PSD}}^n, \mathcal{K}_{\ell_p}^n$ in Examples 1–3 have nonempty interior. Such cones are called *solid*. For a solid cone \mathcal{K} , we can take $\{s \in \mathcal{K}^* : \tilde{x}^\top s = 1\}$ as $\text{base}(\mathcal{K}^*)$ for any $\tilde{x} \in \text{int } \mathcal{K}$, where $\text{int } \mathcal{K}$ denotes the interior of \mathcal{K} . Bases given in Examples 1–3 are constructed by this procedure.

Drawbacks of Algorithm 1 are the construction of $V^{(1)}$, which corresponds to initial cutting hyperplanes, and the computation of $w^{(k)}$ and $x^{(k)}$, that is, the projection onto $\bar{A}\text{cone}(V^{(k)})^*$. In general, it is not easy to construct an initial finite set $V^{(1)}$ such that the following LP problem is lower-bounded:

$$\begin{aligned} &\text{minimize } c^\top x \\ &\text{subject to } Ax = b, \\ &\quad x \in \text{cone}(V^{(1)})^*. \end{aligned} \tag{7}$$

However, if a feasible solution \hat{y} to the dual of the CP problem (1)

$$\begin{aligned} &\text{maximize } b^\top y \\ &\text{subject to } c - A^\top y \in \mathcal{K}^* \end{aligned} \tag{8}$$

is available, we can take $\{c - A^\top \hat{y}\}$ as $V^{(1)}$. In fact, if we take $V^{(1)} := \{c - A^\top \hat{y}\}$, $x \in \text{cone}(V^{(1)})^*$ is equivalent to $(c - A^\top \hat{y})^\top x \geq 0$, which implies $c^\top x \geq b^\top \hat{y}$ for any feasible solution x of the LP problem (7). If there are no dual feasible solutions available, we may construct $V^{(1)}$ heuristically. In the computation of $w^{(k)}$ and $x^{(k)}$ in Algorithm 1, we may solve the following convex quadratic programming (QP) problem for $i = 1, \dots, p$:

$$\begin{aligned} &\text{minimize } \|\bar{A}x - w^{(k-1/2)}\|^2 \\ &\text{subject to } (v^i)^\top x^i \geq 0 \quad (v^i \in V_i^{(k)}; i = 1, \dots, p), \end{aligned} \tag{9}$$

and use an optimal solution of the QP problem (9) as $x^{(k)}$, and set $w^{(k)} = \bar{A}x^{(k)}$. The number of constraints in the QP problem (9) increases as iterations progress. This drawback can be mitigated by using warm-start techniques or exchange methods [8, 14]. For example, if we solve the QP problem (9) using the active set method, we can set $(w^{(k)}, x^{(k)})$ as an initial point of the active method in the $(k + 1)$ -th iteration. The use of initial points often accelerates the active set method. Moreover, like exchange methods, by dropping constraints, we may refrain the number of constraints of the QP problem (9) at every iteration. But, it is nontrivial how to select constraints to be dropped, while preserving the global convergence. Even though warm-start techniques or exchange methods are utilized, solving QP problems is still computationally expensive. Hence, a more sophisticated subroutine is required. The

LP-Newton method [5] for a box-constrained LP problem employs Wolfe’s algorithm [24] to find the nearest point in a zonotope to a given point, and the LP-Newton method [10] for the standard form LP problem (2) uses Wilhelmsen’s algorithm [23] to find the nearest point in a polyhedral cone to a given point. The subroutines are conjectured to be polynomial-time algorithms, and thus the LP-Newton methods for LPs have the potential to be strongly polynomial-time algorithms. Although these subroutines are powerful, it may be difficult to use them in Algorithm 1. In these subroutines, extreme directions or points of the zonotope or polyhedral cone are explicitly required. In our case, unfortunately, we do not have such explicit formulas.

Remark 2 Algorithm 1 is able to detect infeasibility of the LP problem (5), and hence of the CP problem (1), if $\beta^{(k)} \neq b$ and $\gamma^{(k-1/2)} = \gamma^{(k)}$ hold. It can be proved in a manner similar to [10, Lemma 3.3].

3 Convergence analysis

In this section, we prove that a sequence generated by Algorithm 1 converges globally to an optimum of the CP problem (1). We first state the following technical lemmas.

Lemma 1 *Let $\{\alpha^{(k)}\}$ be a sequence of nonnegative scalars and $\{z^{(k)}\}$ be defined by $z^{(k)} := \alpha^{(k)}x^{(k)}$ for each k . If Algorithm 1 generates an infinite sequence $\{x^{(k)}\}$ and there exists an accumulation point of $\{z^{(k)}\}$, then it belongs to \mathcal{H} .*

Proof Let us express $z^{(k)}$ as $z^{(k)} := (z^{1,(k)}, \dots, z^{p,(k)}) = \alpha^{(k)}(x^{1,(k)}, \dots, x^{p,(k)})$ and let $z^* := (z^{1,*}, \dots, z^{p,*})$ be an arbitrarily chosen accumulation point of $\{z^{(k)}\}$. Because of the compactness of $\text{base}(\mathcal{H}_i^*)$ for every $i = 1, \dots, p$, $\{v^{(k)}\} \subseteq \text{base}(\mathcal{H}^*) = \prod_{i=1}^p \text{base}(\mathcal{H}_i^*)$ has at least one accumulation point in $\text{base}(\mathcal{H}^*)$. Denote this point by v^* and express it as $(v^{1,*}, v^{2,*}, \dots, v^{p,*}) \in \prod_{i=1}^p \text{base}(\mathcal{H}_i^*)$. Taking an appropriate subsequence $\{v^{(k)}\}_{k \in K}$, we can assume that $(v^{(k)}, z^{(k)})$ converges to (v^*, z^*) as $k \in K \rightarrow \infty$.

Note that $v^{i,(k)} \in \text{argmin}_{v^i \in \text{base}(\mathcal{H}_i^*)} (v^i)^\top z^{i,(k)}$ holds because $v^{i,(k)} \in \text{argmin}_{v^i \in \text{base}(\mathcal{H}_i^*)} (v^i)^\top x^{i,(k)}$ and $\alpha^{(k)} \geq 0$. Here, by letting $k \in K \rightarrow \infty$, we obtain $v^{i,*} \in \text{argmin}_{v^i \in \text{base}(\mathcal{H}_i^*)} (v^i)^\top z^{i,*}$. Thus, to show $z^* \in \mathcal{H}$, it suffices to prove that $(v^{i,*})^\top z^{i,*} \geq 0$ for each i . To this end, let us fix i and prove $(v^{i,(k)})^\top z^{i,(k)} \geq 0$ for any $k \in K$. Choosing some arbitrary $\hat{k} \in K$, it follows that $(v^{i,(\hat{k})})^\top z^{i,(k)} \geq 0$ for any $k > \hat{k}$ in K , because $z^{i,(k)} = \alpha^{(k)}x^{i,(k)} \in \text{cone}(V^{(k)})^*$ and $v^{i,(\hat{k})} \in V_i^{(k)}$ for any $k > \hat{k}$. Then, by letting $k \in K \rightarrow \infty$, we obtain $(v^{i,(\hat{k})})^\top z^{i,*} \geq 0$. As \hat{k} was arbitrarily chosen from K , we conclude that $(v^{i,(k)})^\top z^{i,*} \geq 0$ ($k \in K$) holds. Finally, by forcing $k \in K \rightarrow \infty$, we conclude that $(v^{i,*})^\top z^{i,*} \geq 0$ for any i . Therefore, $z^* \in \mathcal{H}$. □

Lemma 2 Assume that the optimal value of the CP problem (1) is finite and θ^* be the optimal value. If Algorithm 1 does not stop at the k -th iteration, we have

$$\gamma^{(k-1/2)} \leq \gamma^{(k)} \leq \gamma^{(k+1/2)} \leq \theta^*.$$

Proof Let $\theta^{(k)} \in \mathbb{R}$ be the optimal value of the LSIP problem (3) with $\text{base}(\mathcal{K}_i^*)$ replaced by $V_i^{(k)}$ for $i = 1, \dots, p$, which is a relaxation problem for the CP problem (1). Therefore, $\theta^{(k)} \leq \theta^*$ holds. In a similar manner to [10, Lemma 3.1], it can be verified that

$$\gamma^{(k-1/2)} \leq \gamma^{(k)} \leq \gamma^{(k+1/2)} \leq \theta^{(k)}.$$

Hence, we have the desired result. □

Lemma 3 Assume that the optimal value of the CP problem (1) is finite. If Algorithm 1 generates an infinite sequence $\{\beta^{(k)}\}$, then $\lim_{k \rightarrow \infty} \beta^{(k)} = b$ holds.

Proof To show the desired result, we prove $\lim_{k \rightarrow \infty} \|\beta^{(k)} - b\| = 0$. Note that $\{\gamma^{(k-1/2)}\}$ and $\{\gamma^{(k)}\}$ converge to the same point by Lemma 2. In fact, the joint sequence $\{\gamma^{(k-1/2)}\} \cup \{\gamma^{(k)}\} = \{\gamma^{(1/2)}, \gamma^{(1)}, \gamma^{(1+1/2)}, \gamma^{(2)}, \dots\}$ is a nondecreasing sequence bounded above and thus convergent. Since $\{\gamma^{(k-1/2)}\}$ and $\{\gamma^{(k)}\}$ are both convergent subsequences of the joint sequence, they must share the same limit point. Thus, $|\gamma^{(k+1/2)} - \gamma^{(k-1/2)}| \rightarrow 0$ and $|\gamma^{(k)} - \gamma^{(k-1/2)}| \rightarrow 0$ as $k \rightarrow \infty$. Moreover, as $w^{(k)}$ is the projection of $w^{(k-1/2)}$ onto the supporting hyperplane $H^{(k)}$ and $w^{(k+1/2)} \in H^{(k)}$, we have

$$|\gamma^{(k+1/2)} - \gamma^{(k-1/2)}| = \|w^{(k+1/2)} - w^{(k-1/2)}\| \geq \|w^{(k)} - w^{(k-1/2)}\| = \left\| \begin{pmatrix} \gamma^{(k)} - \gamma^{(k-1/2)} \\ \beta^{(k)} - b \end{pmatrix} \right\|.$$

From these facts, it follows that $\lim_{k \rightarrow \infty} \|\beta^{(k)} - b\| = 0$. □

In what follows, we prove the boundedness of $\{x^{(k)}\}$ and its global convergence to an optimal solution. To this end, let us make the following assumption.

Assumption 1 The optimal solution set \mathcal{S}_P of the CP problem (1) is nonempty and compact.

Remark 3 Assumption 1 holds if the dual CP problem (8) has an optimum and strictly feasible solution, i.e., there exists some $\hat{y} \in \mathbb{R}^m$ such that $c - A^T \hat{y} \in \text{int } \mathcal{K}^*$.

Proposition 1 If Algorithm 1 generates an infinite sequence $\{x^{(k)}\}$ and Assumption 1 holds, then the generated sequence $\{x^{(k)}\}$ is bounded.

Proof Denote the feasible domain of the CP problem (1) by \mathcal{F}_P . To show the boundedness of $\{x^{(k)}\}$, we assume to the contrary for a contradiction. Thus, there

exists some subsequence $\{x^{(k)}\}_{k \in K} \subseteq \{x^{(k)}\}$ such that $\lim_{k \in K \rightarrow \infty} \|x^{(k)}\| = \infty$ and $\|x^{(k)}\| \neq 0$ for any $k \in K$. Then, we have

$$\frac{Ax^{(k)}}{\|x^{(k)}\|} = \frac{\beta^{(k)}}{\|x^{(k)}\|}, \quad \frac{x^{(k)}}{\|x^{(k)}\|} \in \text{cone}(V^{(k)})^*, \quad \frac{c^\top x^{(k)}}{\|x^{(k)}\|} \leq \frac{\theta^*}{\|x^{(k)}\|},$$

where the second relation is derived from the fact that $x^{(k)} \in \text{cone}(V^{(k)})^*$ and $\text{cone}(V^{(k)})^*$ is a cone. Letting $k \in K \rightarrow \infty$ in the above and choosing an arbitrary accumulation point of $\{x^{(k)}/\|x^{(k)}\|\}$, denoted by d^* , implies that

$$Ad^* = 0_m, \quad d^* \in \mathcal{H}, \quad c^\top d^* \leq 0, \quad \|d^*\| = 1, \tag{10}$$

where the first relation follows from the boundedness of $\{\beta^{(k)}\}$ implied by Lemma 3 and the second one follows from Lemma 1 with $z^{(k)} = x^{(k)}/\|x^{(k)}\|$ and $\alpha^{(k)} = 1/\|x^{(k)}\|$. Choose $x^* \in \mathcal{S}_P$ arbitrarily and define $\Omega := \{x^* + \alpha d^* : \alpha \geq 0\}$. We then deduce that $\Omega \subseteq \mathcal{S}_P$ from (10), because

$$A(x^* + \alpha d^*) = b, \quad x^* + \alpha d^* \in \mathcal{H}, \quad c^\top (x^* + \alpha d^*) \leq c^\top x^*$$

for any $\alpha \geq 0$, where the second statement follows from the facts that $x^* \in \mathcal{H}$, $\alpha d^* \in \mathcal{H}$, and \mathcal{H} is a convex cone. Note that Ω is unbounded because $\|d^*\| = 1$, which implies the unboundedness of \mathcal{S}_P . However, this contradicts Assumption 1. As a consequence, $\{x^{(k)}\}$ is bounded. □

Theorem 1 *Under Assumption 1, Algorithm 1 stops in a finite number of iterations and returns an optimal solution, or it generates an infinite sequence $\{x^{(k)}\}$ such that any accumulation point of $\{x^{(k)}\}$ is an optimal solution of the CP problem (1).*

Proof First, we assume Algorithm 1 terminates at the k -th iteration. Since we are assuming the feasibility of the CP problem (1) here, it terminates because conditions $\beta^{(k)} = b$ and $V^{(k+1)} = V^{(k)}$ are simultaneously satisfied. Because $\beta^{(k)} = b$ and $x^{(k)} \in \bar{A}\text{cone}(V^{(k)})$, we have $Ax^{(k)} = \beta^{(k)} = b$ and $(v^i)^\top x^{i,(k)} \geq 0$ ($v^i \in V_i^{(k)}$). This means the feasibility of $x^{(k)}$ for the LP problem (5). In addition, from $V^{(k+1)} = V^{(k)}$, we can deduce $(v^i)^\top x^{i,(k)} \geq 0$ ($v^i \in \text{base}(\mathcal{X}_i^*)$) for each i . Therefore, $x^{(k)}$ is feasible for the LSIP problem (3) and the CP problem (1), and hence $c^\top x^{(k)} \geq \theta^*$ follows, where θ^* denotes the optimal value of the CP problem (1). On the other hand, from Lemma 2, we have $c^\top x^{(k)} = \gamma^{(k)} \leq \theta^*$. Therefore, we obtain $c^\top x^{(k)} = \theta^*$ and conclude that $x^{(k)}$ is optimal for the CP problem (1).

Hereinafter, we assume $\{x^{(k)}\}$ is an infinite sequence. From Proposition 1, $\{x^{(k)}\}$ is bounded and has an accumulation point. Choose an arbitrary accumulation point and denote it by x^* . Without loss of generality, we may assume that $\lim_{k \rightarrow \infty} x^{(k)} = x^*$ by taking a subsequence if necessary. Now, let us recall that $Ax^{(k)} = \beta^{(k)}$ and $x^{(k)} \in \text{cone}(V^{(k)})^*$ hold for any k . Together with Lemmas 1 and 3, this implies that $Ax^* = b$ and $x^* \in \mathcal{H}$, that is, x^* is feasible for the CP problem (1). Hence, $c^\top x^* \geq \theta^*$ follows. On the other hand, by Lemma 2, it holds that $c^\top x^{(k)} = \gamma^{(k)} \leq \theta^*$ for each k , and by taking the limit therein, we obtain $c^\top x^* \leq \theta^*$. Therefore, we have $c^\top x^* = \theta^*$. Thus, we conclude that x^* is optimal for the CP problem (1). □

Remark 4 According to [10, Theorem 3.1], for the case with $\mathcal{X} = \mathbb{R}_+^n$, the number of iterations of the algorithm is, at most, the number of faces of the cone $\bar{A}\mathcal{X}$.

4 Primal-dual algorithm

4.1 Description of the algorithm

In Section 3, we proposed the ALPN method for solving the CP problem (1). In this section, we consider a primal-dual algorithm for the ALPN method, which solves the CP problem (1) and the dual problem (8) simultaneously.

Our primal-dual algorithm is described in Algorithm 2, where $\{x^{(k)}\}$, $\{\gamma^{(k-1/2)}\}$, $\{\gamma^{(k)}\}$, and $\{\beta^{(k)}\}$ represent sequences generated by Algorithm 1. Since Algorithm 2 uses sequences generated by Algorithm 1, they are executed jointly. Note that $(1; -y^{(k)})$ is the normal vector of the supporting hyperplane $H^{(k)}$ of $\bar{A}\mathcal{X}$ at $\bar{A}x$. Supporting hyperplanes of $\bar{A}\mathcal{X}$ play a crucial role as the following property.

Algorithm 2 Primal-dual adaptive LP-Newton method for the CP problem (1).

```

1: for  $k := 1, 2, \dots$  do
2:   if  $\beta^{(k)} \neq b$  then
3:     if  $\gamma^{(k-1/2)} \neq \gamma^{(k)}$  then
4:       Set  $y^{(k)} = -(\beta^{(k)} - b)/(\gamma^{(k)} - \gamma^{(k-1/2)})$ .
5:     else
6:       break ▷ The dual problem (8) is unbounded or infeasible.
7:   else
8:     Set  $y^{(k)} = 0_m$ .
9:   Set  $s^{(k)} = c - A^\top y^{(k)}$ .
10:  if  $(x^{(k)}, y^{(k)}, s^{(k)})$  satisfies the KKT conditions (11) then
11:    return  $y^{(k)}$  ▷  $y^{(k)}$  is an optimal solution of the dual problem (8).

```

Proposition 2 Let $x^* \in \mathcal{X}$ be an optimum of the CP problem (1) and H^* be a supporting hyperplane of $\bar{A}\mathcal{X}$ at $\bar{A}x^* \in \bar{A}\mathcal{X}$. Suppose that $(1; -y^*)$ is a normal vector to H^* . Then, together with x^* , y^* , and $s^* := c - A^\top y^*$ satisfy the Karush–Kuhn–Tucker (KKT) conditions of the CP problem (1):

$$Ax^* = b, A^\top y^* + s = c, x^* \in \mathcal{X}, s^* \in \mathcal{X}^*, (s^*)^\top x^* = 0. \tag{11}$$

In particular, y^* is an optimum of the dual CP problem (8).

Proof As H^* is a supporting hyperplane of $\bar{A}\mathcal{X}$ at $\bar{A}x^* = (c^\top x^*; Ax^*)$ and x^* solves the CP problem (1), we have

$$\bar{A}x^* \in \operatorname{argmin}_{w \in \bar{A}\mathcal{X}} (1, -(y^*)^\top)w.$$

Hence, it holds that

$$x^* \in \operatorname{argmin}_{x \in \mathcal{X}} (c - A^\top y^*)^\top x. \tag{12}$$

By the KKT conditions of the problem (12), there exists some $s^* \in \mathbb{R}^n$ such that

$$s^* = c - A^\top y^*, \quad x^* \in \mathcal{X}, \quad s^* \in \mathcal{X}^*, \quad (s^*)^\top x^* = 0,$$

which, together with $Ax^* = b$, implies the system (11). The optimality of y^* for the dual CP problem (8) is obvious. □

4.2 Convergence analysis

In our analysis of the primal-dual algorithm, we make the following assumption:

Assumption 2 Slater’s constraint qualification holds for the CP problem (1), i.e., there exists some $\hat{x} \in \mathbb{R}^n$ such that $\hat{x} \in \text{int } \mathcal{X}$ and $A\hat{x} = b$, and the matrix A is of full row rank.

Under Assumptions 1 and 2, it is guaranteed that the optimal set of the dual CP problem (8) is nonempty and compact.

Remark 5 In Proposition 2, by assuming the existence of a supporting hyperplane H^* of $\bar{A}\mathcal{X}$ at $\bar{A}x^* \in \bar{A}\mathcal{X}$ that is not parallel to L , we proved the existence of a KKT point. We can derive the assumption of such a supporting hyperplane from Assumption 2 (details are omitted here) but the converse is not true. Hence, we can regard the assumption of the existence of such a supporting hyperplane as a weaker constraint qualification than Slater’s one and the full rank assumption.

Here, we prove the global convergence of Algorithm 2 under Assumptions 1 and 2.

Theorem 2 Under Assumptions 1 and 2, Algorithm 2 stops in a finite number of iterations and returns an optimal solution, or it generates an infinite sequence $\{y^{(k)}\}$ such that $\{y^{(k)}\}$ is bounded and any accumulation point of $\{y^{(k)}\}$ solves the dual CP problem (8).

Proof Since the case of the finite termination is obvious, we assume $\{y^{(k)}\}$ is an infinite sequence in what follows. We first show the boundedness of $\{y^{(k)}\}$. To derive a contradiction, suppose that $\{y^{(k)}\}$ is unbounded, and hence there exists some subsequence $\{y^{(k)}\}_{k \in K}$ such that $\|y^{(k)}\| \rightarrow \infty$ and $y^{(k)} \neq 0_m$ for any $k \in K$. Note that $H^{(k)}$ is a supporting hyperplane of $\bar{A}\text{cone}(V^{(k)})^*$ at $\bar{A}x^{(k)}$ that has the normal vector $(1; -y^{(k)})$. Then, by the definition of $x^{(k)}$, we find that

$$\bar{A}x^{(k)} \in \underset{w \in \bar{A}\text{cone}(V^{(k)})^*}{\text{argmin}} \quad (1, -(y^{(k)})^\top)w,$$

and thus

$$x^{(k)} \in \underset{x \in \text{cone}(V^{(k)})^*}{\text{argmin}} \quad (c - A^\top y^{(k)})^\top x.$$

Table 1 Performance of the adaptive LP-Newton method

Dimensions			# hyperplanes			
<i>m</i>	<i>n</i>	(n_1, n_2, \dots, n_p)	Time [s]	# iter	Initial	Final
10	200	(1, 1, ..., 1)	0.1	3.5	200.0	200.0
10	200	(2, 2, ..., 2)	0.3	3.8	200.0	385.5
10	200	(5, 5, ..., 5)	7.3	19.1	320.0	1019.4
10	200	(10, 10, ..., 10)	58.4	62.5	360.0	1580.9
10	200	(20, 20, ..., 20)	167.6	141.1	380.0	1779.4
10	200	(100, 100)	136.8	352.0	396.0	1098.0
10	200	200	48.1	274.4	398.0	671.4
10	350	(1, 1, ..., 1)	0.1	3.6	350.0	350.0
10	350	(2, 2, ..., 2)	0.9	3.5	350.0	592.8
10	350	(5, 5, ..., 5)	34.0	19.2	560.0	1786.6
10	350	(10, 10, ..., 10)	303.0	64.7	630.0	2849.2
10	350	(35, 35, ..., 35)	1911.0	281.9	680.0	3489.0
10	350	(175, 175)	462.4	414.7	696.0	1523.4
10	350	350	104.4	217.3	698.0	914.3
10	500	(1, 1, ..., 1)	0.1	3.8	500.0	500.0
10	500	(2, 2, ..., 2)	2.0	3.5	500.0	805.7
10	500	(5, 5, ..., 5)	98.0	19.9	800.0	2638.5
10	500	(10, 10, ..., 10)	856.6	66.0	900.0	4135.5
10	500	(50, 50, ..., 50)	5618.0	372.5	980.0	4694.8
10	500	(250, 250)	866.0	398.1	996.0	1790.1
10	500	500	183.6	172.7	998.0	1169.7
50	200	(1, 1, ..., 1)	0.0	4.0	200.0	200.0
50	200	(2, 2, ..., 2)	0.3	4.0	200.0	461.8
50	200	(5, 5, ..., 5)	5.2	16.8	320.0	947.6
50	200	(10, 10, ..., 10)	26.3	47.9	360.0	1297.9
50	200	(20, 20, ..., 20)	67.3	105.3	380.0	1423.0
50	200	(100, 100)	62.7	257.1	396.0	908.2
50	200	200	30.5	238.1	398.0	635.1
50	350	(1, 1, ..., 1)	0.1	4.0	350.0	350.0
50	350	(2, 2, ..., 2)	1.3	3.9	350.0	752.3
50	350	(5, 5, ..., 5)	27.3	17.2	560.0	1690.5
50	350	(10, 10, ..., 10)	184.6	54.4	630.0	2499.0
50	350	(35, 35, ..., 35)	675.8	200.5	680.0	2674.9
50	350	(175, 175)	235.7	308.3	696.0	1310.6
50	350	350	64.0	164.6	698.0	861.6
50	500	(1, 1, ..., 1)	0.1	3.9	500.0	500.0
50	500	(2, 2, ..., 2)	3.2	3.8	500.0	1026.4
50	500	(5, 5, ..., 5)	80.1	17.9	800.0	2487.1
50	500	(10, 10, ..., 10)	548.5	56.4	900.0	3669.7

Table 1 (continued)

Dimensions			# hyperplanes			
<i>m</i>	<i>n</i>	(n_1, n_2, \dots, n_p)	Time [s]	# iter	Initial	Final
50	500	(50, 50, ..., 50)	2287.4	274.9	980.0	3719.0
50	500	(250, 250)	392.6	281.0	996.0	1556.0
50	500	500	125.5	140.3	998.0	1137.3
100	200	(1, 1, ..., 1)	0.0	4.0	200.0	200.0
100	200	(2, 2, ..., 2)	0.3	4.0	200.0	481.9
100	200	(5, 5, ..., 5)	3.8	14.9	320.0	875.7
100	200	(10, 10, ..., 10)	13.6	38.1	360.0	1102.0
100	200	(20, 20, ..., 20)	30.3	78.4	380.0	1154.0
100	200	(100, 100)	35.3	195.4	396.0	784.8
100	200	200	24.5	211.5	398.0	608.5
100	350	(1, 1, ..., 1)	0.1	4.0	350.0	350.0
100	350	(2, 2, ..., 2)	1.5	4.0	350.0	828.4
100	350	(5, 5, ..., 5)	24.1	16.6	560.0	1645.8
100	350	(10, 10, ..., 10)	114.6	46.9	630.0	2236.5
100	350	(35, 35, ..., 35)	397.4	166.1	680.0	2331.0
100	350	(175, 175)	159.0	257.8	696.0	1209.6
100	350	350	87.9	212.3	698.0	909.3
100	500	(1, 1, ..., 1)	0.1	4.0	500.0	500.0
100	500	(2, 2, ..., 2)	5.9	4.0	500.0	1195.9
100	500	(5, 5, ..., 5)	72.2	17.0	800.0	2399.6
100	500	(10, 10, ..., 10)	388.0	50.2	900.0	3360.0
100	500	(50, 50, ..., 50)	1286.0	223.6	980.0	3206.0
100	500	(250, 250)	399.2	290.1	996.0	1574.2
100	500	500	120.4	139.4	998.0	1136.4

Using the KKT conditions and the definition of $x^{(k)}$ again, we have

$$x^{(k)} \in \text{cone}(V^{(k)})^*, c - A^\top y^{(k)} \in \text{cone}(V^{(k)}), (c - A^\top y^{(k)})^\top x^{(k)} = 0, Ax^{(k)} = \beta^{(k)}. \tag{13}$$

As $\text{cone}(V^{(k)}) \subseteq \mathcal{H}^*$ follows from $\mathcal{H} \subseteq \text{cone}(V^{(k)})^*$, we find that $c - A^\top y^{(k)} \in \mathcal{H}^*$. Divide $c - A^\top y^{(k)} \in \mathcal{H}^*$ and $(c - A^\top y^{(k)})^\top x^{(k)} = 0$ by $\|y^{(k)}\|$ and let $k \in K \rightarrow \infty$ in (13). Choose an accumulation point of $\{y^{(k)}/\|y^{(k)}\|\}$ and denote it by d^* . Let x^* be an accumulation point of $\{x^{(k)}\}$ (recall Theorem 1). Without loss of generality, we may assume that $\lim_{k \in K \rightarrow \infty} (x^{(k)}, y^{(k)}/\|y^{(k)}\|) = (x^*, d^*)$. Then, noting that $(c - A^\top y^{(k)})/\|y^{(k)}\| \in \mathcal{H}^*$ for any $k \in K$ and $\lim_{k \rightarrow \infty} \beta^{(k)} = b$ by Lemma 3, it holds that

$$x^* \in \mathcal{H}, -A^\top d^* \in \mathcal{H}^*, (Ax^*)^\top d^* = 0, Ax^* = b. \tag{14}$$

Here, let y^* be an arbitrary optimum of the dual CP problem (8). Then, the set $\Omega := \{y^* + \alpha d^* : \alpha \geq 0\}$ is contained in the optimal solution set of the dual CP problem (8),

denoted by \mathcal{S}_D . This is shown as follows. Fix an arbitrary value of $\alpha \geq 0$. Using the second relation in (14) and $c - A^\top y^* \in \mathcal{K}^*$ together with the fact that \mathcal{K}^* is a closed convex cone, we have $c - A^\top (y^* + \alpha d^*) \in \mathcal{K}^*$, and so $y^* + \alpha d^*$ is feasible for the dual CP problem (8). Moreover, we can deduce from (14) that

$$b^\top (y^* + \alpha d^*) = b^\top y^* + \alpha (Ax^*)^\top d^* = b^\top y^*,$$

which indicates that the optimal value of the dual CP problem (8) is also attained at $y^* + \alpha d^*$. Therefore, $\Omega \subseteq \mathcal{S}_D$. Note that Ω is unbounded because $\|d^*\| \neq 0$, which implies the unboundedness of \mathcal{S}_D . However, this contradicts the boundedness of \mathcal{S}_D derived from Assumptions 1 and 2. Hence, $\{y^{(k)}\}$ is bounded.

The remaining part of the claim is easy to prove by taking the limit in (13) with the second relation replaced by $c - A^\top y^{(k)} \in \mathcal{K}$ using $\text{cone}(V^{(k)}) \subseteq \mathcal{K}^*$. \square

5 Numerical results

We conducted numerical experiments to verify the performance of our proposed algorithm for SOCP problems. We implemented the ALPN method with MATLAB R2018a (9.4.0.813654) on a workstation running CentOS release 6.10 with eight Intel Xeon CPUs (E3-1276 v3 3.60 GHz) and 32 GB RAM.

In our experiments, we solved the CP problem (1) with $\mathcal{K}_i := \mathcal{K}_{\text{SOC}}^{n_i}$ for each i , which is SOCP. We used an initial polyhedral approximation of the i -th block $\mathcal{K}_{\text{SOC}}^{n_i}$ of \mathcal{K} with $n_i \geq 2$ given by $V_i^{(1)} := \bigcup_{j=1}^{n_i-1} \{e_j, -e_j\}$, where e_j denotes the j -th column of the identity matrix of appropriate size. Note that $V_i^{(1)}$ defined above exactly represents $\mathcal{K}_{\text{SOC}}^{n_i}$ if $n_i = 1, 2$. In the projection step, we solved Problem (9) using the MATLAB function `lsq1in`. We stopped the algorithm when an approximate primal optimal solution was found, namely, a primal solution $x^{(k)}$ at the k -th iteration satisfies

$$\max\{\|Ax^{(k)} - b\|, \max_{i=1}^p \|\tilde{x}^{i,(k)} - x_1^{i,(k)}\|\} \leq 10^{-4}.$$

We randomly generated the following instances of the SOCP problem. First, we set m, p , and (n_1, n_2, \dots, n_p) and randomly generated each element of A from the standard Gaussian distribution. Next, we set $b = A\tilde{x}$ and $c = A^\top e - \tilde{s}$, where e is the vector whose elements are all ones and $\tilde{x}^i = \tilde{s}^i := e_1$ for each i . Note that the two points \tilde{x} and \tilde{s} are interior feasible solutions of the primal and dual problems, respectively.

5.1 Performance of the adaptive LP-Newton method

For the ALPN method, Table 1 presents the average runtime, number of iterations, and number of hyperplanes in the initial and final approximations of the second-order cone over ten executions. From this table, we can make the following observations:

- When $\mathcal{K} = \prod_{i=1}^p \mathcal{K}_{\text{SOC}}^{n_i}$ is polyhedral-like, i.e., $p \approx n$ and $n_i \approx 1$ for all $i = 1, \dots, p$, the ALPN method works well. The algorithm gives a good polyhedral

Table 2 Comparison with SDPT3

Dimensions			Time [s]	
m	n	(n_1, n_2, \dots, n_p)	ALPN	SDPT3
1400	1500	(3, 3, ..., 3)	177.3	366.6
1700	1800	(3, 3, ..., 3)	260.4	638.4
2000	2100	(3, 3, ..., 3)	363.4	970.0

approximation of \mathcal{K} with a small number of hyperplanes. As a result, there are few iterations and the computation time is short.

- When $\mathcal{K} = \prod_{i=1}^p \mathcal{K}_{\text{SOC}}^{n_i}$ is *medium-dimensional*, i.e., $1 \ll p \ll n$ and $1 \ll n_i \ll n$ for all $i = 1, \dots, p$, the ALPN method becomes slow, although it gets better when $\mathcal{K} = \prod_{i=1}^p \mathcal{K}_{\text{SOC}}^{n_i}$ is *high-dimensional*, i.e., $p \approx 1$ and $n_i \approx n$ for all $i = 1, \dots, p$. The medium-dimensional \mathcal{K} requires many hyperplanes to obtain a good polyhedral approximation.
- The total dimension n of the variables seems to be positively correlated with the runtime, although the runtime of the original LP-Newton method for LP is almost independent of n [5]. This difference arises from the solution methods of the minimum norm point step. In our implementation, we solve the QP problem (9) using the MATLAB function `lsq1in`, for which the computation time depends on n .
- Surprisingly, the number m of linear constraints is *negatively* correlated with the runtime. This might be because the dimension of the feasible region is low for large values of m , and this region can then be approximated by a small number of hyperplanes.

5.2 Comparison with a primal-dual interior-point method

We also compared our proposed ALPN method with a primal-dual interior-point method. In this experiment, we solved the randomly generated instances using our implementation of ALPN and SDPT3 [22], which is a MATLAB implementation of the primal-dual interior-point method. Basically, SDPT3 was found to be faster than the ALPN method. However, the computation time of SDPT3 increases with m , whereas that of ALPN decreases as m and p increase. For instances with large values of m and p , ALPN outperformed SDPT3. The results are presented in Table 2, which shows the average runtime of the ALPN and SDPT3 methods over ten runs.

6 Conclusions

In this paper, we have developed an LP-Newton method for the CP problem (1) through a transformation into an LSIP problem with an infinite number of linear inequality constraints. The proposed ALPN algorithm produces a sequence by sequentially projecting a current point onto a polyhedral cone arising from finitely

many linear inequality constraints chosen from the constraints of the LSIP problem. We have also proposed a dual algorithm for the ALPN method for solving the dual of the CP problem. Under some mild assumptions, we have proved that an arbitrary accumulation point of a couple of sequences generated by the two proposed algorithms are optima of the CP problem and its dual. Finally, we have conducted some numerical experiments for SOCP problems and compared the performance of our algorithms with that of the primal-dual interior-point method. Our future work includes analysis under weaker assumptions, for example, in the absence of Slater's constraint qualification, or infeasible cases.

Acknowledgments We would like to thank the editor Claude Brezinski, the anonymous reviewers, and Dr. Bruno Figueira Lourenço for useful comments.

Funding information This research is supported by JST CREST JPMJCR14D2 and JSPS Grants-in-Aid for Young Scientists 15K15943, 16K16357, and 19K15247.

References

1. Ben-Tal, A., Nemirovski, A.: Lectures on Modern Convex Optimization. SIAM (2001)
2. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
3. Chen, X., Tseng, P.: Non-interior continuation methods for solving semidefinite complementarity problems. *Math. Program.* **95**, 431–474 (2003)
4. Faraut, J., Koranyi, A.: Analysis on Symmetric Cones. Oxford University Press, Oxford (1994)
5. Fujishige, S., Hayashi, T., Yamashita, K., Zimmermann, U.: Zonotopes and the LP-Newton method. *Optim. Eng.* **10**, 193–205 (2009)
6. Hayashi, S., Okuno, T., Ito, Y.: Simplex-type algorithm for second-order cone programmes via semi-infinite programming reformulation. *Optim. Methods Softw.* **31**, 1272–1297 (2016)
7. Hayashi, S., Yamashita, N., Fukushima, M.: A combined smoothing and regularization method for monotone second-order cone complementarity problems. *SIAM J. Optim.* **15**, 593–615 (2005)
8. Hettich, R., Kortanek, K.O.: Semi-infinite programming: theory, methods, and applications. *SIAM Rev.* **35**, 380–429 (1993)
9. Huang, Z.H., Ni, T.: Smoothing algorithms for complementarity problems over symmetric cones. *Comput. Optim. Appl.* **45**, 557–579 (2010)
10. Kitahara, T., Mizuno, S., Shi, J.: The LP-Newton method for standard form linear programming problems. *Oper. Res. Lett.* **41**, 426–429 (2013)
11. Kitahara, T., Sukegawa, N.: A simple projection algorithm for linear programming problems. *Algorithmica* **81**, 167–178 (2019)
12. Kitahara, T., Tsuchiya, T.: An extension of Chubanov's polynomial-time linear programming algorithm to second-order cone programming. *Optim. Methods Softw.* **33**, 1–25 (2018)
13. Lobo, M.S., Vandenberghe, L., Boyd, S., Lebret, H.: Applications of second-order cone programming. *Linear Algebra Appl.* **284**, 193–228 (1998)
14. López, M., Still, G.: Semi-infinite programming. *Eur. J. Oper. Res.* **180**, 491–518 (2007)
15. Lourenço, B.F., Kitahara, T., Muramatsu, M., Tsuchiya, T.: An extension of Chubanov's algorithm to symmetric cones. *Math. Program.* **173**, 117–149 (2019)
16. Monteiro, R.D.C., Tsuchiya, T.: Polynomial convergence of primal-dual algorithms for the second-order cone program based on the MZ-family of directions. *Math. Programm.* **88**, 61–83 (2000)
17. Muramatsu, M.: A pivoting procedure for a class of second-order cone programming. *Optim. Methods Softw.* **21**, 295–315 (2006)
18. Nesterov, Y., Nemirovski, A.: Interior-Point Polynomial Methods in Convex Programming. SIAM (1994)
19. Silvestri, F., Reinelt, G.: The LP-Newton method and conic optimization. arXiv:1611.09260v2 (2017)

20. Skajaa, A., Ye, Y.: A homogeneous interior-point algorithm for nonsymmetric convex conic optimization. *Math. Program.* **150**, 391–422 (2015)
21. Todd, M.J.: Semidefinite optimization. *Acta Numer.* **10**, 515–560 (2001)
22. Tütüncü, R.H., Toh, K.C., Todd, M.J.: Solving semidefinite-quadratic-linear programs using SDPT3. *Math. Program.* **95**, 189–217 (2003)
23. Wilhelmsen, D.R.: A nearest point algorithm for convex polyhedral cones and applications to positive linear approximation. *Math. Comput.* **30**, 48–57 (1976)
24. Wolfe, P.: Finding the nearest point in a polytope. *Math. Program.* **11**, 128–149 (1976)
25. Zhadan, V.: Two-phase simplex method for linear semidefinite optimization. *Optim. Lett.* **13**, 1969–1984 (2019)

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