An Early Termination Technique for ADMM in Mixed Integer Conic Programming

Yuwen Chen*, Paul Goulart

Abstract

The branch-and-bound (B&B) method is a commonly used technique in mixed-integer programming, whose performance is known to improve substantially if early termination methods can be applied to accelerate the pruning of branches. We propose an early termination technique that estimates a lower bound of the objective of current node problem and can stop the computation early instead of solving it to optimality. We show that our proposed technique can be generalized to ADMM-based mixed integer conic programming and speed up convergence in practice.

I. Introduction

Mixed integer conic programming (MICP) encompasses a broad range of problems such as mixed integer quadratic programming (MIQP), mixed integer second-order cone programming (MISOCP) and mixed integer semidefinite programming (MISDP) and has applications in portfolio optimization [1], hybrid model predictive control [2], power electronics [3], robust truss topology [4], power system unit commitment [5] and planning for unmanned aerial vehicles [6]. Many numerical solvers have been developed for MICP, including Gurobi [7], Mosek [8], SCIP [9], [10] and Pajarito [11], [12]. Throughout, we will consider MICPs in the general form:

$$\begin{aligned} \min \quad & \frac{1}{2} x^\top P x + q^\top x \\ \text{s.t.} \quad & Ax + s = b, \\ & s \in \mathcal{C}, \\ & x_{\mathbb{I}} \in \mathcal{Z}, \end{aligned} \tag{1}$$

where $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$ and $\mathcal{C} \subseteq \mathbb{R}^m$ is a convex cone. The vector $x \in \mathbb{R}^n$ is the decision variable and $\mathbb{I}$ denotes the entries of x constrained to a finite integer set $\mathcal{Z}$. The objective function is convex quadratic with symmetric positive semidefinite $P \in \mathbb{S}_+^n$ and vector $q \in \mathbb{R}^n$. If we set $b = \mathbf{0}$ and $\mathcal{C}$ as a box constraint, $\mathcal{C} = [-u, -l] \subseteq \mathbb{R}^m$, (1) reduces to the MIQP

$$\begin{aligned} \min \quad & \frac{1}{2} x^\top P x + q^\top x \\ \text{s.t.} \quad & l \leq Ax \leq u, \\ & x_{\mathbb{I}} \in \mathcal{Z}. \end{aligned} \tag{2}$$

*Corresponding author: yuwen.chen@eng.ox.ac.uk. The authors are with the Department of Engineering Science, University of Oxford, UK.

The inequality condition in (2) generalizes to both linear constraints, if $l = u$, and to one sided inequality constraints, if elements of (l, u) take values in $\{-\infty, \infty\}$.

A. *Techniques for Mixed Integer Programming*

Mixed integer programming (MIP) is $\mathcal{NP}$-hard in general and can not be solved in polynomial time. The branch-and-bound (B&B) method is the most commonly used technique for the search of a optimal solution in MIP solvers. The branch-and-cut algorithm is a variant of B&B that adds cutting planes iteratively to refine the constraint set. Since B&B algorithms are time-consuming, many techniques have been proposed to speed up MIP computation, including presolve, heuristics and early termination. Presolve [13] can be regarded as a collection of preprocessing methods before solving a MIP, including bound strengthening, coefficient strengthening, constraint reduction and conflict analysis. Heuristics are generally divided into start and improvement heuristics [14], both of which are crucial for pruning nodes in B&B algorithms. Start heuristics, like the feasibility pump [15], aim to find a feasible solution as early as possible when the B&B algorithm starts, while improvement heuristics, like RINS [16] and the crossover method [17], search for a feasible point of better objective value based on information from previously obtained feasible points.

If a dual feasible point of a relaxed problem within a B&B search can be generated intermediately with corresponding dual objective larger than the current upper bound (value of a primal feasible solution), one can abort the computation of the corresponding node before solving it to the optimality. Previous work [18]–[22] in QP early termination generally requires that the Hessian $P \succ 0$ in the problem (2) where a dual feasible solution can be obtained easily. Recently, [23] present an early termination technique for the relaxed condition $P \succeq 0$ but the feasibility of the generated dual point is not guaranteed.

ADMM [24] is a first-order optimization method that is known for its application in distributed and large-scale optimization and has been well-studied for many years. However, contrary to its popularity in convex optimization, much less has been done for it in the mixed integer case. The authors of [25] propose an MIQP solver where each relaxed convex problem in B&B is solved by an ADMM solver. In contrast [26] and [27] deal with the mixed integer problem directly via ADMM by introducing auxiliary variables for the projection onto the integer constraint set, and can be regarded as nonconvex ADMM algorithms without convergence guarantees. In addition, the authors of [28] utilize ADMM decomposing a MIP into multiple small MIPs where each small problem is solved by an external MIP solver in every outer loop of ADMM.

B. *Contributions*

This paper proposes an early termination strategy for an ADMM algorithm to solve convex QP relaxations using a B&B method. It is the first early termination technique for mixed integer ADMM that doesn't require strict positive definiteness of the Hessian P in (2). We develop a correction method to compute a dual feasible solution for early termination before an optimum of the QP is found. We require the mild assumption that the constraint set is bounded, which is reasonable for many real-world applications. The complexity of the correction step is no worse than solving an inner loop step, i.e. $O(n^2)$, and hence computationally efficient.

C. Organization

The rest of the paper is organized as follows: Section 2 gives background on the B&B method and the ADMM algorithm in OSQP. Section 3 presents our early termination strategy for mixed integer conic programming. Numerical results are then provided in Section 4 to show its effectiveness. The conclusions are summarized in Section 5.

D. Notation

Denote the n-dimensional real space $\mathbb{R}^n$, the $n \times n$ symmetric matrices $\mathbb{S}^n$, and the set of positive semidefinite matrices $\mathbb{S}^n_+$. We denote $|\mathbb{I}|$ as the number of elements in the discrete set $\mathbb{I}$. For a closed and convex set $\mathcal{C}$, denote the normal cone of $\mathcal{C}$ by

$$N_{\mathcal{C}}(x) := \left\{ y \in \mathbb{R}^n \mid \sup_{\bar{x} \in \mathcal{C}} \langle \bar{x} - x, y \rangle \leq 0 \right\},$$

the tangent cone of $\mathcal{C}$ by

$$T_{\mathcal{C}}(x) := \text{cl} \left\{ w \mid \exists \lambda > 0 \text{ with } x + \lambda w \in \mathcal{C} \right\},$$

and the recession cone of $\mathcal{C}$ by

$$\mathcal{C}^{\infty} := \left\{ y \in \mathbb{R}^n \mid x + ay \in \mathcal{C}, \forall x \in \mathcal{C}, \forall a \geq 0 \right\}.$$

The proximal operator of a convex, closed and proper function $f : \mathbb{R}^n \to \mathbb{R}$ is given by

$$\text{prox}_f(x) := \underset{y}{\text{argmin}} \left\{ f(y) + \frac{1}{2} ||y - x||^2 \right\}.$$

The norm $|| \cdot ||$ is the Euclidean norm unless otherwise specified. We denote the indicator function of a nonempty, closed convex set $\mathcal{C} \subseteq \mathbb{R}^n$ by

$$I_{\mathcal{C}}(x) := \begin{cases} 0 & x \in \mathcal{C} \\ +\infty & \text{otherwise,} \end{cases}$$

and the projection of $x \in \mathbb{R}^n$ onto $\mathcal{C}$ by

$$\Pi_{\mathcal{C}}(x) := \underset{y \in \mathcal{C}}{\text{argmin}} ||x - y||^2.$$

We further denote the support function of $\mathcal{C}$ by

$$\sigma_{\mathcal{C}}(x) = \sup_{y \in \mathcal{C}} \langle x, y \rangle.$$

Finally, we denote the polar cone of a convex cone $\mathcal{K}$ by

$$\mathcal{K}^{\circ} := \left\{ y \in \mathbb{R}^n \mid \sup_{x \in \mathcal{K}} \langle x, y \rangle \leq 0 \right\}.$$

II. Background

A. Mixed Integer Conic Programming

The MICP (1) is a mixed integer extension of the conic formulation in [29]. To make our early termination technique possible, we require the following assumption.

Assumption II.1. *The feasible region in* (1) *is a subset of* $\mathcal{D} := \{x \mid l \leq x \leq u,\ l, u \in \mathbb{R}^n\}$ *where* l, u *are both finite.*

The boundedness assumption above is reasonable for many real world applications when x is an 0-1 switching signal or subjected to some physical limitations, like in some QP problems where $||x|| \leq R$. Under Assumption II.1, (1) can be reformulated as

$$\begin{aligned} \min \quad & \frac{1}{2}x^\top P x + q^\top x \\ \text{s.t} \quad & Ax + s = b, \\ & s \in \mathcal{C}, \\ & x \in \mathcal{D}, \quad x_{\mathbb{I}} \in \mathcal{Z}. \end{aligned} \tag{3}$$

If we combine the box constraint $\mathcal{D}$ with the linear constraint $Ax + s = b$, the continuous relaxation of (3) is

$$\begin{aligned} \min \quad & f(x, s_c, s_b) := \frac{1}{2}x^\top P x + q^\top x \\ \text{s.t} \quad & \begin{bmatrix} A \\ -I \end{bmatrix} x + \begin{bmatrix} s_c \\ s_b \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}, \qquad (\text{CP}(l, u)) \\ & s_c \in \mathcal{C}, \\ & s_b \in \mathcal{B}, \end{aligned} \tag{4}$$

where $\mathcal{B} = \mathcal{D} \cap \mathcal{RZ}$, $\mathcal{RZ}$ is the continuous relaxation of the integer constraint $\mathcal{Z}$ and is denoted as $[\underline{l}, \bar{u}] \subset \mathbb{R}^{|\mathbb{I}|}$.

B. ADMM step

This section discusses an ADMM to solve problem (4), which will be used for solving individual nodes within a branch-and-bound algorithm. Using the same splitting as in COSMO [29], we solve a problem equivalent to (4) as follows,

$$\begin{aligned} \min \quad & \frac{1}{2}\tilde{x}^\top P \tilde{x} + q^\top \tilde{x} + I_{\mathcal{L}}(\tilde{x}, \tilde{s}_c, \tilde{s}_b) + I_{\mathcal{C}}(s_c) + I_{\mathcal{B}}(s_b) \\ \text{s.t} \quad & (\tilde{x}, \tilde{s}_c, \tilde{s}_b) = (x, s_c, s_b), \end{aligned} \tag{5}$$

where $\mathcal{L}$ is the affine set defined by

$$\mathcal{L} := \left\{ (x, s) \middle| \begin{bmatrix} A \\ -I \end{bmatrix} x + \begin{bmatrix} s_c \\ s_b \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \right\}.$$

Compared to the original problem (4), we introduce the duplicate $(\tilde{x}, \tilde{s}_c, \tilde{s}_b)$ for (x, s_c, s_b) and rewrite the set constraints $\mathcal{L}, \mathcal{C}$ and $\mathcal{B}$ as indicator functions in the objective function for the problem (5). The augmented Lagrangian of (5) is then given by

$$
\begin{aligned}
& L\left(\tilde{x}, \tilde{s}_c, \tilde{s}_b, x, s_c, s_b, \lambda, y_c, y_b\right) \\
=& \frac{1}{2}\tilde{x}^\top P \tilde{x} + q^\top \tilde{x} + I_{\mathcal{L}}(\tilde{x}, \tilde{s}_c, \tilde{s}_b) + I_{\mathcal{C}}(s_c) + I_{\mathcal{B}}(s_b) + \frac{\sigma}{2}||\tilde{x} - x + \frac{1}{\sigma}\lambda||^2 \\
& + \frac{\rho_c}{2}||\tilde{s}_c - s_c + \frac{1}{\rho}y_c||^2 + \frac{\rho_b}{2}||\tilde{s}_b - s_b + \frac{1}{\rho_b}y_b||^2,
\end{aligned}
$$

with parameters $\sigma > 0, \rho_c > 0, \rho_b > 0$ and dual variables $\lambda \in \mathbb{R}^n, y_c \in \mathbb{R}^m, y_b \in \mathbb{R}^n$ for the matching constraints $\tilde{x} = x$, $\tilde{s}_c = s_c$, $\tilde{s}_b = s_b$, respectively. Following the approach in [29]–[31], our ADMM algorithm becomes,

$$
(\tilde{x}^{k+1}, \tilde{s}_c^{k+1}, \tilde{s}_b^{k+1}) \leftarrow \underset{\tilde{x}, \tilde{s}_c, \tilde{s}_b}{\operatorname{argmin}} \quad L\left(\tilde{x}, \tilde{s}_c, \tilde{s}_b, x^k, s_c^k, s_b^k, \lambda^k, y_c^k, y_b^k\right), \tag{6}
$$

$$
x^{k+1} \leftarrow \alpha\tilde{x}^{k+1} + (1-\alpha)x^k + \frac{1}{\sigma}\lambda^k, \tag{7}
$$

$$
s_c^{k+1} \leftarrow \underset{s_c}{\operatorname{argmin}} \ \frac{\rho_c}{2}\|\alpha\tilde{s}_c^{k+1} + (1-\alpha)s_c^k - s_c + \tfrac{1}{\rho_c}y_c^k\|^2 + I_{\mathcal{C}}(s_c), \tag{8}
$$

$$
s_b^{k+1} \leftarrow \underset{s_b}{\operatorname{argmin}} \ \frac{\rho_b}{2}\|\alpha\tilde{s}_b^{k+1} + (1-\alpha)s_b^k - s_b + \tfrac{1}{\rho_b}y_b^k\|^2 + I_{\mathcal{B}}(s_b), \tag{9}
$$

$$
\lambda^{k+1} \leftarrow \lambda^k + \sigma\left(\alpha\tilde{x}^{k+1} + (1-\alpha)x^k - x^{k+1}\right), \tag{10}
$$

$$
y_c^{k+1} \leftarrow y_c^k + \rho_c\left(\alpha\tilde{s}_c^{k+1} + (1-\alpha)s_c^k - s_c^{k+1}\right), \tag{11}
$$

$$
y_b^{k+1} \leftarrow y_b^k + \rho_b\left(\alpha\tilde{s}_b^{k+1} + (1-\alpha)s_b^k - s_b^{k+1}\right), \tag{12}
$$

where $\alpha \in (0, 2)$ is the relaxation parameter for the Douglas-Rachford splitting method as in [29], [31], [32]. If we set $\lambda^0 = 0$, then $\lambda^k = 0$ for all k afterwards. Note that, the minimization step (6) is an equality constrained QP that is always solvable, similar to [29], [31], [33]. The minimization over s_c and s_b is equivalent to a projection step onto the cone $\mathcal{C}$ and the box set $\mathcal{B}$.

Algorithm 1 ADMM algorithm for CP($\underline{x}, \bar{x}$)

Require:

Initial values $x^0, s_c^0, s_b^0, y_c^0, y_b^0$, problem data P, q, A, b, and parameters $\sigma > 0, \rho_c > 0, \rho_b > 0, \alpha \in (0, 2)$. Assign a bounded constraint $\mathcal{B} = \mathcal{D} \cap \{x | x_{\mathbb{I}} \in [\underline{x}, \bar{x}]\}$ to x.

1: **while** *termination criteria not satisfied* **do**
2: $(\tilde{x}^{k+1}, \tilde{s}_c^{k+1}, \tilde{s}_b^{k+1}) \leftarrow$ solve equality constrained QP (6)
3: $x^{k+1} \leftarrow \alpha \tilde{x}^{k+1} + (1-\alpha) x^k$
4: $s_c^{k+1} \leftarrow \Pi_{\mathcal{C}} \left(\alpha \tilde{s}_c^{k+1} + (1-\alpha) s_c^k + \frac{1}{\rho_c} y_c^k \right)$
5: $s_b^{k+1} \leftarrow \Pi_{\mathcal{B}} \left(\alpha \tilde{s}_b^{k+1} + (1-\alpha) s_b^k + \frac{1}{\rho_b} y_b^k \right)$
6: $y_c^{k+1} \leftarrow y_c^k + \rho_c \left(\alpha \tilde{s}_c^{k+1} + (1-\alpha) s_c^k - s_c^{k+1} \right)$
7: $y_b^{k+1} \leftarrow y_b^k + \rho_b \left(\alpha \tilde{s}_b^{k+1} + (1-\alpha) s_b^k - s_b^{k+1} \right)$
8: **end while**

Termination and infeasibility detection criteria follow that in [29].

C. Branch and Bound

The branch-and-bound method computes the optimal solution x in (3) by exploring the integer combinations in a tree. It repeatedly branches on any entry of x in the integer index set $\mathbb{I}$ and solves the continuous relaxation (4) until the global optimum is found, see Figure 1. However, the size of the corresponding tree increases exponentially with the number of integer variables. To alleviate the computational burden, pruning is key to reduce the computational time for B&B algorithms. A classical B&B algorithm is summarized in Algorithm 2. It starts with the continuous relaxation (4), denoted as CP($\underline{l}, \bar{u}$) and saved as the root node of tree $\mathcal{T}$. For each round, we pick a node CP($\underline{x}, \bar{x}$) from the tree $\mathcal{T}$ and solve it with a solution $\tilde{x}$ and a lower bound $f(\tilde{x}, \tilde{s}_c, \tilde{s}_b)$ by Algorithm 1. Infeasibility of CP($\underline{x}, \bar{x}$) infers that the subproblem, (3) restricted by the bounded constraint $\{x | x_{\mathbb{I}} \in [\underline{x}, \bar{x}]\}$, is also infeasible. The node would be pruned if the lower bound $f(\tilde{x}, \tilde{s}_c, \tilde{s}_b)$ is larger than the current upper bound U as CP($\underline{x}, \bar{x}$) can't yield a potentially feasible lower objective value than U. If the corresponding solution $\tilde{x}$ satisfies all integer constraints in (3) with $f(\tilde{x}, \tilde{s}_c, \tilde{s}_b) < U$, we update the best solution x^* and the upper bound for (3). Otherwise, we branch an element of $\tilde{x}$ that violates an integer constraint in (2) and add new nodes into the tree $\mathcal{T}$. Note that the quality of an upper bound is closely related to the branching method in a B&B algorithm. In this paper, we branch on the entry with the maximal fractional part.

III. Early Termination for Mixed Integer ADMM

Early termination requires generation of a dual feasible point for implementation. The dual feasible point can provide a valid lower bound for the optimal dual cost, and therefore a lower bound for the optimal primal cost due to weak duality of the relaxed convex problem. If the lower bound for a given node is greater than the existing upper bound corresponding to a feasible solution, we know the optimum would not be better than the current best solution

Algorithm 2 B&B for bounded MICP (3)

Require:

Initialize upper bound $U \leftarrow +\infty$, node tree $\mathcal{T} \leftarrow \text{CP}(\underline{l}, \bar{u})$

while $\mathcal{T} \neq \emptyset$ **do**
  Pick and remove $\text{CP}(\underline{x}; \bar{x})$ from $\mathcal{T}$
  $\tilde{x}, f(\tilde{x}, \tilde{s}_c, \tilde{s}_b) \leftarrow \text{CP}(\underline{x}; \bar{x})$
  if $\text{CP}(\underline{x}; \bar{x})$ is infeasible **then**
    prune current node
  else if $f(\tilde{x}, \tilde{s}_c, \tilde{s}_b) > U$ **then**
    prune current node
  else if $\tilde{x}$ is *integer feasible* **then**
    $U \leftarrow f(\tilde{x}, \tilde{s}_c, \tilde{s}_b), x^* \leftarrow \tilde{x}$
    fathom nodes in $\mathcal{T}$ with lower bound $> U$
  else
    branch node $\text{CP}(\underline{x}, \bar{x})$
  end if
end while

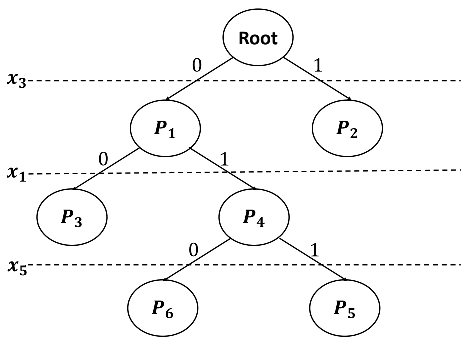

Figure 1: Illustration of branch and bound. The integer index set is $\mathbb{I} = \{1, 3, 5\}$ with the branching order $3 \rightarrow 1 \rightarrow 5$.

and the node computation can be terminated. However, it is hard to design the counterpart for mixed integer ADMM or primal-dual interior point method (IPM) [23], since they do not generate dual feasible points until the algorithm converges. In this section, we first introduce the dual formulation of the convex relaxed problem (4) and then describe our early termination technique for mixed integer ADMM.

A. Dual Formulation

Following the discussion of the dual formulation of a CP in [30], the dual of (4) is

$$\max_{x, y, y_x} -\frac{1}{2} x^\top P x + b^\top y_c - \sigma_{\mathcal{C}}(y_c) - \sigma_{\mathcal{B}}(y_b) \tag{13a}$$

$$\text{s.t} \quad Px + q - A^\top y_c + y_b = 0 \tag{13b}$$

$$y_c \in (\mathcal{C}^{\infty})^{\circ}, \;\; y_b \in \mathbb{R}^n \tag{13c}$$

In (13a), the support function $\sigma_{\mathcal{B}}(y_b)$ is explicit, i.e.

$$\sigma_{\mathcal{B}}(y_b) = u^{\top} y_b^{+} + l^{\top} y_b^{-}, \tag{14}$$

where

$$y_b^{+} = \max\{y_b, 0\}, \quad y_b^{-} = \min\{y_b, 0\}.$$

Moreover, since $y_c \in (\mathcal{C}^{\infty})^{\circ}$ the support function $\sigma_{\mathcal{C}}(y_c)$ is also easily computable when $\mathcal{C}$ is a cone or a box constraint. Since the dual cost (13a) is easily computable, the remaining issue is to ensure dual feasibility, i.e. (13b), (13c) when designing a valid early termination technique. We elaborate on this point in the next subsection.

B. Correction

Early termination relies on finding a dual feasible solution, which does not hold in general for ADMM iterates (y_c^k, y_b^k). However, we have a partial dual feasibility result of ADMM iterates (y_c^k, y_b^k) and summarize it in the following lemma.

Lemma III.1. *The iterates in Algorithm 1 satisfy:*

$$y_c^k \in N_{\mathcal{C}}(s_c^k) \subset (\mathcal{C}^{\infty})^{\circ}, \;\; y_b^k \in N_{\mathcal{B}}(s_b^k) \tag{15}$$

Proof. $y_c^k \in N_{\mathcal{C}}(s_c^k), y_b^k \in N_{\mathcal{B}}(s_b^k)$ have been proved in [30]. Note that the polar of the tangent cone $T_{\mathcal{C}}(x)$ is the normal cone $N_{\mathcal{C}}(x)$ if $\mathcal{C}$ is regular at x, see Corollary 6.30 in [34]. Such a regularity condition is satisfied for a closed and convex set due to Theorem 6.9 in [34]. Since $\mathcal{C}^{\infty} \subset R_{\mathcal{C}}(x) \subset T_{\mathcal{C}}(x)$ for any $x \in \mathcal{C}$ by the definition of the recession cone $\mathcal{C}^{\infty}$, then $N_{\mathcal{C}}(x) = (T_{\mathcal{C}}(x))^{\circ} \subset (\mathcal{C}^{\infty})^{\circ}$. □

In Lemma III.1, we prove that $y_c^k \in N_{\mathcal{C}}(s_c^k) \subset (\mathcal{C}^{\infty})^{\circ}$ for every iteration, which means the dual feasibility constraint (13c) is automatically satisfied. More problematic is the feasibility of the equality constraint (13b) for (x^k, y_c^k, y_b^k). To satisfy (13b), we define the dual residual as

$$r^k := Px^k + q - A^{\top} y_c^k + y_b^k. \tag{16}$$

Our goal is then to offset the dual residual r^k via a correction of (x^k, y_c^k, y_b^k). We adopt a similar approach to [23], where the correction is obtained from solving an auxiliary optimization problem. Compared with the correction method in [23], our method ensures dual feasibility at all times and can also be used in linear programming (LP), i.e. when $P = 0$. Moreover, our approach is applicable to general mixed integer conic programming beyond MIQP. The correction on (x^k, y_c^k, y_b^k) relies on the following optimization problem,

$$\begin{aligned} \min \quad & \frac{\alpha}{2}||\Delta x^k||^2 + \frac{\gamma}{2}||\Delta y_b^k||^2 \\ \text{s.t.} \quad & P\Delta x^k + \Delta y_b^k = -r^k. \end{aligned} \tag{17}$$

We allow changes in x^k, y_b^k but keep y_c^k unchanged because the satisfaction of the conic constraint $y_c \in (C^\infty)^\circ$ is hard to preserve when modifying y_c^k. Since (17) is a convex QP with equality constraints, its solution is equivalent to solving the linear system

$$\begin{bmatrix} \alpha I & -P \\ -P & -\frac{1}{\gamma} I \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \gamma \Delta y_b^k \end{bmatrix} = \begin{bmatrix} 0 \\ r^k \end{bmatrix}, \tag{18}$$

which is a quasi-definite linear system and can be solved efficiently via $LDL^\top$ factorization since the matrix on the left hand side of (18) is unchanged for any convex subproblem in B&B. Alternatively, we can apply the Conjugate Gradient (CG) method to the following linear equation

$$(\frac{\gamma}{\alpha} P^2 + I) \Delta y_b^k = -r^k \tag{19}$$

instead of the $LDL^\top$ decomposition, and then recover Δx by

$$\Delta x^k = \frac{\gamma}{\alpha} P \Delta y_b^k. \tag{20}$$

The estimated dual cost is then given by

$$\underline{D}^k = -\frac{1}{2}(x^k + \Delta x^k)^\top P (x^k + \Delta x^k) + b^\top y_c^k - \sigma_{\mathcal{C}}(y_c^k) - \sigma_{\mathcal{B}}(y_b^k + \Delta y_b^k). \tag{21}$$

The ratio γ/α determines whether Δx^k or Δy_b^k should be favoured. If γ/α is set to 0 or $P = 0$, (17) simplifies to $\Delta y_b^k := -r^k$. If $\gamma/\alpha \to +\infty$, (17) reduces to $\Delta x^k = P^{-1} r^k$ when P is nonsingular.

To summarize this section, we give our branch and bound (B&B) algorithm for MICP with early termination in Algorithm 3. Steps 4 to 20 solve a convex relaxation CP($\underline{x}, \bar{x}$) as in Algorithm 1, which corresponds to solving a node in a B&B algorithm, and the remaining parts are the standard B&B algorithm. The novelty is the introduction of early termination (step 14-16 and step 21-22) for ADMM-based MICP solver within a B&B framework.

Algorithm 3 B&B for MICP with early termination

Require:
Initialize upper bound $U \leftarrow +\infty$, node tree $\mathcal{T} \leftarrow \mathrm{CP}(\underline{l}, \bar{u})$

while $\mathcal{T} \neq \emptyset$ **do**
  Pick and remove $\mathrm{CP}(\underline{x}, \bar{x})$ from $\mathcal{T}$
  Initial values $x^0, s_c^0, s_b^0, y_c^0, y_b^0$, problem data P, q, A, b, and parameters $\sigma > 0, \rho_c > 0, \rho_b > 0, \alpha \in (0,2)$. Assign a bounded constraint $\mathcal{B} = \mathcal{D} \cap \{x | x_{\mathbb{I}} \in [\underline{x}, \bar{x}]\}$ to x.
  while *infeasibility of* $\mathrm{CP}(\underline{x}, \bar{x})$ *is not detected* **do**
    $(\tilde{x}^{k+1}, \tilde{s}_c^{k+1}, \tilde{s}_b^{k+1}) \leftarrow$
      solve equality constrained QP (6)
    $x^{k+1} \leftarrow \alpha \tilde{x}^{k+1} + (1-\alpha) x^k$
    $s_c^{k+1} \leftarrow \Pi_{\mathcal{C}}\left(\alpha \tilde{s}_c^{k+1} + (1-\alpha) s_c^k + \frac{1}{\rho_c} y_c^k\right)$
    $s_b^{k+1} \leftarrow \Pi_{\mathcal{B}}\left(\alpha \tilde{s}_b^{k+1} + (1-\alpha) s_b^k + \frac{1}{\rho_b} y_b^k\right)$
    $y_c^{k+1} \leftarrow y_c^k + \rho_c\left(\alpha \tilde{s}_c^{k+1} + (1-\alpha) s_c^k - s_c^{k+1}\right)$
    $y_b^{k+1} \leftarrow y_b^k + \rho_b\left(\alpha \tilde{s}_b^{k+1} + (1-\alpha) s_b^k - s_b^{k+1}\right)$
    Compute the estimated dual cost $\underline{D}^k$ via (17) or (21)
    if $\underline{D}^k \geq U$ **then**
      terminate early for $\mathrm{CP}(\underline{x}, \bar{x})$
    end if
    if *termination criteria is satisfied* **then**
      return optimal solution $\hat{x} = x^{k+1}, \hat{s}_c = s_c^{k+1}, \hat{s}_b = s_b^{k+1}$ and $f(\hat{x}, \hat{s}_c, \hat{s}_b)$
    end if
  end while
  if $\mathrm{CP}(\underline{x}, \bar{x})$ terminate early **then**
    prune current node
  else if $\mathrm{CP}(\underline{x}, \bar{x})$ is infeasible **then**
    prune current node
  else if $f(\hat{x}, \hat{s}_c, \hat{s}_b) > U$ **then**
    prune current node
  else if $\hat{x}$ is *integer feasible* **then**
    $U \leftarrow f(\hat{x}, \hat{s}_c, \hat{s}_b), x^* \leftarrow \hat{x}, s_c^* \leftarrow \hat{s}_c$
    fathom nodes in $\mathcal{T}$ with lower bound $> U$
  else
    branch node $\mathrm{CP}(\underline{x}, \bar{x})$
  end if
end while

IV. Numerical Results

We implement Algorithm 3 both with and without the early termination condition in step 13-16 in Julia. Both algorithms use the same B&B strategies as in Section II A of the miOSQP algorithm [25]. Each relaxed problem in B&B is solved using the method [31]. All tests are implemented on a personal computer of Inter Core i7-9700 CPU @ 3.00GHz with 16GB RAM.

A. Random MIQPs

We generate a positive semidefinite matrix $P \in \mathbb{S}_{+}^{n}$ via $P = QDQ^{\top}$, where $Q \in \mathbb{R}^{n\times r}$ is the first r columns of an orthonormal matrix generated by QR decomposition of a normally-distributed random square matrix with distribution $\mathcal{N}(0,1)$. In addition $D \in \mathbb{R}^{r\times r}$ is a diagonal matrix where each diagonal entry is generated uniformly from the interval $[1, 11]$, ensuring moderate condition number of P. The vector q is generated randomly via $\mathcal{N}(0,1)$. The constraint matrix A is generated randomly by $\mathcal{N}(0,1)$ with sparsity 0.7 and b is set to 0. We set the constraint set $\mathcal{C}$ as a box constraint $[l_b, u_b]$ where l_b is generated uniformly on $[-2,-1] \in \mathbb{R}^m$ and u_b is generated uniformly on $[2,3] \in \mathbb{R}^m$. At the same time, the bounded constraint for x is $[l, u]$, where l is generated uniformly on $[-2,-1] \in \mathbb{R}^n$ and u is generated uniformly on $[2,3] \in \mathbb{R}^n$, except integer variables set to binary. The settings above ensure the problem is always integer feasible. For the correction step (17), we set the ration $\frac{\gamma}{\alpha} = 100$. Since the early termination requires a valid upper bound, we enable it once a feasible solution is obtained in the B&B process. We check for early termination every 25 iterations.

We test different combinations of n, m, p, r, where n is the number of total variables, m is the number of linear constraints in $\mathcal{C}$, p is the number of integer variables among n variables and r is the rank of P. Each problem is solved 20 times and we record the percentage of nodes that trigger early termination in Algorithm 3 on average. The results are shown in Table I. We count nodes that trigger early termination successfully after finding the first feasible solution. For all cases, at least 10% nodes benefit from early termination and more than 20% for small MIQPs. Future work would focus on the effectiveness of our early termination in combination with other MIQP techniques.

n	m	p	r	Percentage (%)
50	25	5	25	33.83
50	25	5	50	21.08
50	100	5	25	23.37
50	100	5	50	21.14
100	50	10	50	22.82
100	50	10	100	12.72
100	200	10	50	12.12
100	200	10	100	11.06

Table I: Percentage of nodes that trigger early termination after finding the first feasible solution.

B. Mixed Integer Model Predictive Control

We consider mixed integer model predictive control (MIMPC) for current reference tracking in power converters [3]. The MIMPC problem can be formulated as a MIQP

$$\begin{aligned} \min \quad & \sum_{t=0}^{T} \gamma^t l(x_t) + \gamma^\top V(x_T) \\ \text{s.t.} \quad & x_0 = x_{\text{init}}, \\ & x_{t+1} = Ax_t + Bu_t, \\ & ||u_t - u_{t-1}||_\infty \le 1, \\ & u_t \in \{-1, 0, 1\}^3 \times \{0, 1\}^3, \end{aligned} \tag{22}$$

where γ is a discount factor and T is the time horizon. The quadratic state penalty cost $l(x_t)$ is for current tracking and $V(x_T)$ is a final stage cost using approximate dynamic programming (ADP). The initial state is x_{init} and the system dynamics is $x_{t+1} = Ax_t + Bu_t$ where the state x_t has dimension 12 representing the internal motor currents and voltages and the input vector u_t has dimension 6 including the three semiconductor devices positions with integer values $\{-1, 0, 1\}$ and three additional binary components required to model the system. $\|u_t - u_{t-1}\|_\infty \le 1$ avoids shoot-through in the inverter positions (changes from -1 to 1 or vice-versa) that can damage the components. By eliminating $x_t, t \in \{1, \dots, T\}$ via the state dynamics, problem (22) reduces to a problem depending only on input variables $u_0, \dots, u_{T-1}$ and the initial state x_0, see details in [3].

We set $T = 2$ for the time horizon and simulate closed-loop MIMPC for 100 consecutive intervals. Figure 2 compares the performance of early terminated MIQP with fully solved MIQP. Since a valid upper bound U is required for early termination, we start to count QP iterations only when the first feasible solution of (22) is found. Here, we define one loop of steps 4-19 in Algorithm 3 as a QP iteration. We find that, for most intervals, the first integer feasible solution coincides with the optimum and has no further QP iterations, which is irrelevant to early termination. For the remaining intervals, early termination is effective to reduce the number of QP iterations, about 16% reduction of QP iterations after finding the first feasible solution.

V. Conclusion

Contribution: We have successfully developed an early termination technique for the mixed integer programming based on ADMM, under a mild boundedness assumption that is reasonable for many real-world applications. Compared with the correction method for QP in [23], our method guarantees the dual feasibility of the estimated point for every iteration. Moreover, our early termination strategy is applicable for mixed integer ADMM algorithm in various mixed integer conic problems.

Future Work: At the same time, we also have some issues remained for future work. One issue is about the asymmetric box constraint $x \in [l, u]$, which may cause different sensitivity in the correction step. Another issue is the dependence on the boundedness assumption, Assumption II.1, to guarantee the dual feasibility for each iteration. A potential direction

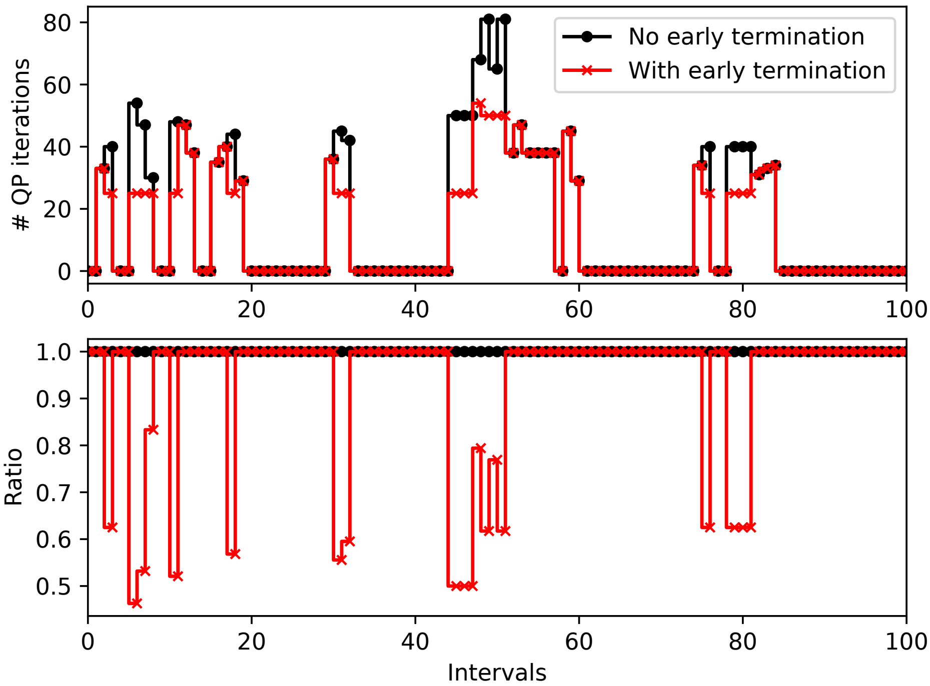

Figure 2: Early termination vs. fully solved MIMPC

is to exploit the degree of freedom from the constraint $Ax + s = b$, where some rows can be rewritten as box constraints. Such constraints have unconstrained dual variables and can be utilized for correction.

VI. Extension

A. Degree of freedom

In this part, we try to generalize the Assumption II.1 w.r.t. the rank information from the data matrices.

The core of the Assumption II.1 is that, the residual $r^k := Px^k + q - A^\top y^k$ could be compensated via the correction $(\delta x^k, \delta y^k)$ on any generated primal-dual iterates (x^k, y^k) while keeping the corrected primal-dual iterates dual feasible. Suppose Ξ is the index set where the corresponding entries are unconstrained, i.e. $y_\Xi \in \mathbb{R}^{|\Xi|}$, we have the following Theorem:

Assumption VI.1. *Our dual correction strategy is applicable as long as rank*$([P \mid (A_{\Xi,\cdot})^\top]) = n$.

Previously, we define a bounded set $\mathcal{D}$ in Assumption II.1 such that $(A_{\Xi,\cdot})^\top$ contains a submatrix $I \in \mathbb{R}^{n\times n}$ which guarantees the Assumption VI.1 automatically. However, we need to introduce the full dimension of x instead of integer indices in $\mathbb{I}$, which not only increases the dimension of the linear system but the correction is also sensitive to the bound we make in $\mathcal{D}$. So it is better to exploit existing constraint sets rather than introducing additional box constraints.

According to (13c), the dual variable y is unconstrained as long as the convex set $\mathcal{B}$ is bounded. In practice, there are some other special constraints other than the box constraint that we could exploit in the operator splitting methods if their projection steps are efficient.

1) Unit norm ball: Three typical norm balls are l_1, l_2, l_∞ unit balls, and the efficient projection steps have been summarized in Section 6.5 in [35]:

- l_1 norm: If $\|s_i\|_1 \leq 1$, it is done. Otherwise, we find λ such that

$$\sum_{j=1}^{d_i} (|s_{i,j}| - \lambda)_+ = 1.$$

- l_2 norm:

$$\Pi_{\mathcal{C}_i}(s_i) = \begin{cases} s_i/\|s_i\|_2, & \|s_i\|_2 \geq 1 \\ s_i, & \|s_i\|_2 \leq 1. \end{cases}$$

- l_∞ norm:

$$\Pi_{\mathcal{C}_i}(s_i) = \begin{cases} 1, & s_i > 1 \\ s_i, & |s_i| \leq 1 \\ -1, & s_i < -1. \end{cases}$$

Note that the ball $\mathcal{B}$ with the l_∞ norm is a special box constraint.

2) Zero set: When $\mathcal{C}_i$ becomes $\{0\}^{|d_i|}$, we could get $A_i x = b_i$, which is a linear equality constraint.

B. Absolute value constraints

$$|a^\top x| \leq b \quad \rightarrow \quad -b \leq a^\top x \leq b$$

1) Matrix norm ball:

- Frobenius norm
- spectral norm

Could it generate an unconstrained $Y \in \mathbb{R}^{m \times n}$?

2) Cutting planes: **Disjunctive cuts:** Generate a pair of cut that can be written into a box constraint in this Gurobi talk, around 45:00.

C. Extension to the interior-point method

Note that the main idea behind our correction strategy is to make the current primal-dual iterate (x^k, s^k, y^k) dual feasible. The similar idea could also be extended to primal-dual interior point methods (IPM). However, the previous correction relies on Assumption II.1 where the projection onto a convex compact set is computationally cheap. However, we introduce logarithmically homogeneous self-concordant barrier functions in a primal-dual interior point method for the inequality constraints, which requires $\mathcal{C}$ to be a cone. To deal with it, a slight modification allows us to exploit boundedness information for the dual correction.

Consider we solve a convex relaxation satisfying Assumption II.1 in each node computation,

$$\begin{aligned} \min \quad & \frac{1}{2} x^\top P x + q^\top x \\ \text{s.t.} \quad & Ax + s = b, \\ & s \in \mathcal{C}, l \leq x \leq u. \end{aligned} \tag{23}$$

We split the box constraint into two nonnegative cones,

$$\begin{aligned} \min \quad & \frac{1}{2} x^\top P x + q^\top x \\ \text{s.t.} \quad & \begin{bmatrix} A \\ I \\ -I \end{bmatrix} x + \begin{bmatrix} s \\ s_- \\ s_+ \end{bmatrix} = \begin{bmatrix} b \\ u \\ -l \end{bmatrix}, \\ & s \in \mathcal{C}, s_- \geq 0, s_+ \geq 0. \end{aligned}$$

Following the dualization in (13), we obtain the dual of (23),

$$\begin{aligned} \max_{x,y,y_l,y_u} \quad & -\frac{1}{2} x^\top P x + b^\top y + y_+^\top u - y_-^\top l - \sigma_{\mathcal{C}}(y) \\ \text{s.t} \quad & Px + q - A^\top y - y_+ + y_- = 0 \\ & y \in (\mathcal{C}^\infty)^\circ, \ y_- \leq 0, y_+ \leq 0, \end{aligned} \tag{24}$$

where the support function values for y_l, y_u are indeed 0. Note that, if we define $y = y_+ - y_-$, y is an unconstrained variable that is applicable for the dual correction. Suppose we only make corrections on y_- and y_+ where the effective term is

$$y_+^\top u - y_-^\top l = y_+^\top (u - l) + (y_+ - y_-)^\top l = y_+^\top (u - l) + \Delta y^\top l.$$

where $\Delta y := y_+ - y_-$. Note that we have $y_+^\top (u-l) \leq 0$ due to $y_- \leq 0, y_+ \leq 0, u - l \geq 0$. To maximize the dual objective in (24) at any iteration k, we choose $\Delta y^k = Px^k + q - A^\top y^k$, $y_+^k = \min\{0, \Delta y^k\}$ and $y_-^k = y_+^k - \Delta y^k$ such that (x^k, y^k, y_+^k, y_-^k) is a dual feasible point for (24).

D. Modification for the optimization-based correction

In our previous optimization-based correction (17), it has a conditioning number P^2 for computing Δy_b. If we change the cost for Δx to $\Delta x^\top P \Delta x$ as in [23], the new optimization problem becomes

$$\begin{aligned} \min \quad & \frac{1}{2} \Delta x^{k\top} P \Delta x^k + \frac{\gamma}{2} ||\Delta y_b^k||^2 \\ \text{s.t.} \quad & P\Delta x^k + \Delta y_b^k = -r^k. \end{aligned} \tag{25}$$

The corresponding KKT condition of (25) is

$$\begin{aligned} P\Delta x^k - P\lambda &= 0 \\ \gamma \Delta y_b^k - \lambda &= 0 \\ P\Delta x^k + \Delta y_b^k + r^k &= 0, \end{aligned}$$

which reduces to

$$\begin{bmatrix} P & I \\ I & -\gamma \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta y_b^k \end{bmatrix} = \begin{bmatrix} -r^k \\ 0 \end{bmatrix}, \tag{26}$$

if we keep $\Delta x^k = \lambda$. (26) is also equivalent to

$$\Delta y_b^k = -(\gamma P + I)^{-1} r^k, \Delta x^k = \gamma \Delta y_b^k. \tag{27}$$

Hence, we see the conditioning number is reduced to P instead of P^2 in (19).

References

[1] G. Cornuéjols, J. Peña, and R. Tütüncü, *Mixed Integer Programming Models: Portfolios with Combinatorial Constraints*, 2nd ed. Cambridge University Press, 2018.

[2] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[3] B. Stellato, T. Geyer, and P. J. Goulart, "High-speed finite control set model predictive control for power electronics," *IEEE Transactions on Power Electronics*, vol. 32, no. 5, pp. 4007–4020, 2017.

[4] K. Yonekura and Y. Kanno, "Global optimization of robust truss topology via mixed integer semidefinite programming," *Optimization and Engineering*, vol. 11, no. 3, pp. 355–379, 2010.

[5] X. Zheng, H. Chen, Y. Xu, Z. Li, Z. Lin, and Z. Liang, "A mixed-integer sdp solution to distributionally robust unit commitment with second order moment constraints," *CSEE Journal of Power and Energy Systems*, vol. 6, no. 2, pp. 374–383, 2020.

[6] R. Deits and R. Tedrake, "Efficient mixed-integer planning for uavs in cluttered environments," in *IEEE International Conference on Robotics and Automation, ICRA 2015, Seattle, WA, USA, 26-30 May, 2015*. IEEE, 2015, pp. 42–49.

[7] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2021. [Online]. Available: https://www.gurobi.com

[8] M. ApS, *MOSEK Modeling Cookbook*, 2021. [Online]. Available: https://docs.mosek.com/MOSEKModelingCookbook-a4paper.pdf

[9] G. Gamrath, D. Anderson, K. Bestuzheva, W.-K. Chen, L. Eifler, M. Gasse, P. Gemander, A. Gleixner, L. Gottwald, K. Halbig, G. Hendel, C. Hojny, T. Koch, P. Le Bodic, S. J. Maher, F. Matter, M. Miltenberger, E. Mühmer, B. Müller, M. E. Pfetsch, F. Schlösser, F. Serrano, Y. Shinano, C. Tawfik, S. Vigerske, F. Wegscheider, D. Weninger, and J. Witzig, "The SCIP Optimization Suite 7.0," Optimization Online, Technical Report, March 2020.

[10] T. Gally, M. E. Pfetsch, and S. Ulbrich, "A framework for solving mixed-integer semidefinite programs," *Optimization Methods and Software*, vol. 33, no. 3, pp. 594–632, 2018.

[11] C. Coey, M. Lubin, and J. P. Vielma, "Outer approximation with conic certificates for mixed-integer convex problems," *Mathematical Programming Computation*, vol. 12, no. 2, pp. 249–293, 2020.

[12] M. Lubin, "Mixed-integer convex optimization : outer approximation algorithms and modeling power," Ph.D. dissertation, Massachusetts Institute of Technology, 2017.

[13] T. Achterberg, R. E. Bixby, Z. Gu, E. Rothberg, and D. Weninger, "Presolve reductions in mixed integer programming," *INFORMS Journal on Computing*, vol. 32, no. 2, p. 473–506, 2020.

[14] T. Berthold, "Primal heuristics for mixed integer programs," Ph.D. dissertation, Technische Universität Berlin, 2006.

[15] T. Berthold, A. Lodi, and D. Salvagnin, "Ten years of feasibility pump, and counting," *EURO Journal on Computational Optimization*, vol. 7, no. 1, pp. 1–14, 2019.

[16] E. Danna, E. Rothberg, and C. L. Pape, "Exploring relaxation induced neighborhoods to improve mip solutions," *Mathematical Programming*, vol. 102, no. 1, pp. 71–90, 2005.

[17] E. Rothberg, "An evolutionary algorithm for polishing mixed integer programming solutions," *INFORMS Journal on Computing*, vol. 19, no. 4, pp. 534–541, 2007.

[18] R. Fletcher and S. Leyffer, "Numerical experience with lower bounds for miqp branch-and-bound," *SIAM J. on Optimization*, vol. 8, no. 2, p. 604–616, 1998.

[19] V. V. Naik and A. Bemporad, "Embedded mixed-integer quadratic optimization using accelerated dual gradient projection," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10 723–10 728, 2017, 20th IFAC World Congress.

[20] A. Bemporad, "Solving mixed-integer quadratic programs via nonnegative least squares," *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 73–79, 2015.

[21] D. Axehill and A. Hansson, "A mixed integer dual quadratic programming algorithm tailored for mpc," in *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006, pp. 5693–5698.

[22] C. Buchheim, M. D. Santis, S. Lucidi, F. Rinaldi, and L. Trieu, "A feasible active set method with reoptimization for convex quadratic mixed-integer programming," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1695–1714, 2016.

[23] J. Liang, S. D. Cairano, and R. Quirynen, "Early termination of convex qp solvers in mixed-integer programming for real-time decision making," *IEEE Control Systems Letters*, vol. 5, no. 4, pp. 1417–1422, 2021.

[24] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[25] B. Stellato, V. V. Naik, A. Bemporad, P. Goulart, and S. Boyd, "Embedded mixed-integer quadratic optimization using the osqp solver," in *2018 European Control Conference (ECC)*, 2018, pp. 1536–1541.

[26] R. Takapoui, N. Moehle, S. Boyd, and A. Bemporad, "A simple effective heuristic for embedded mixed-integer quadratic programming," in *2016 American Control Conference (ACC)*, 2016, pp. 5619–5625.

[27] A. Alavian and M. C. Rotkowitz, "Improving admm-based optimization of mixed integer objectives," in *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, 2017, pp. 1–6.

[28] K. Mihić, M. Zhu, and Y. Ye, "Managing randomization in the multi-block alternating direction method of multipliers for quadratic optimization," *Mathematical Programming Computation*, vol. 13, no. 2, pp. 339–413, 2021.

[29] M. Garstka, M. Cannon, and P. Goulart, "Cosmo: A conic operator splitting method for convex conic problems," *Journal of Optimization Theory and Applications*, vol. 190, no. 3, pp. 779–810, 2021.

[30] G. Banjac, P. Goulart, B. Stellato, and S. Boyd, "Infeasibility detection in the alternating direction method of multipliers for convex optimization," *Journal of Optimization Theory and Applications*, vol. 183, no. 2, pp. 490–519, 2019.

[31] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.

[32] J. Eckstein and D. P. Bertsekas, “On the douglas—rachford splitting method and the proximal point algorithm for maximal monotone operators,” *Mathematical Programming*, vol. 55, no. 1, pp. 293–318, 1992.

[33] B. O’donoghue, E. Chu, N. Parikh, and S. Boyd, “Conic optimization via operator splitting and homogeneous self-dual embedding,” *Journal of Optimization Theory and Applications*, vol. 169, no. 3, p. 1042–1068, 2016.

[34] R. Rockafellar and R. J.-B. Wets, *Variational Analysis*. Springer Verlag, 1998.

[35] N. Parikh and S. Boyd, “Proximal algorithms,” *Found. Trends Optim.*, vol. 1, no. 3, p. 127–239, 2014.

Appendix

Lemma A.1. *The first-order optimality conditions for Problem (4) are*

$$\begin{aligned} Ax + s_c = b, x = s_b, \\ s_c \in \mathcal{C}, s_b \in \mathcal{B}, \\ Px + q - A^\top y_c + y_b = 0, \\ y_c \in N_\mathcal{C}(s_c),\ \ y_b \in N_\mathcal{B}(s_b). \end{aligned} \tag{28}$$

Proof. Problem (4) can be rewritten as

$$\begin{aligned} \min\ & f(x, s_c, s_b) := \frac{1}{2}x^\top Px + q^\top x + I_\mathcal{C}(s_c) + I_\mathcal{B}(s_b) \\ \text{s.t}\quad & \begin{bmatrix} A \\ -I \end{bmatrix} x + \begin{bmatrix} s_c \\ s_b \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \end{aligned} \tag{29}$$

and the Lagrangian of (4) is defined as

$$\begin{aligned} \mathcal{L}(x, s_c, s_b, y_c, y_b) =& \frac{1}{2}x^\top Px + q^\top x - y_c^\top (Ax + s_c - b) - y_b^\top(-x + s_b) \\ &+ I_\mathcal{C}(s_c) + I_\mathcal{B}(s_b). \end{aligned} \tag{30}$$

The optimality condition for (29) is then (28) by taking partial derivatives to x, s_c, s_b, y_c, y_b and the existence of indicator functions $I_\mathcal{C}(s_c), I_\mathcal{B}(s_b)$. □

Lemma A.2. *The dual of Problem (4), CP($\underline{l}, \bar{u}$), is given by (13a) to (13c).*

Proof. The dual function can be derived from the Lagrangian (30) by

$$\begin{aligned} & g(x, s, s_x, y, y_x) \\ :=& \inf_{x, s_c, s_b} \mathcal{L}(x, s_c, s_b, y_c, y_b) \\ =& \inf_{x, s_c \in \mathcal{C}, s_b \in \mathcal{B}} \frac{1}{2}x^\top Px + q^\top x - y_c^\top(Ax + s_c - b) - y_b^\top(-x + s_b) \\ =& \ b^\top y_c + \inf_x \left(\frac{1}{2}x^\top Px + (-A^\top y_c + q + y_b)^\top x\right) - \sup_{s_c \in \mathcal{C}} y_c^\top s_c - \sup_{s_b \in \mathcal{B}} y_b^\top s_b. \end{aligned}$$

Note the minimum of the Lagrangian above is attained when $Px + q - A^\top y_c + y_b = 0$ and $y_c \in (\mathcal{C}^\infty)^\circ$, which is the domain of support function $\sigma_\mathcal{C}(y_c)$. Hence, the dual problem of (4) can be written as (13a)-(13c). □

The termination criteria and infeasibility detection follows that in [29]–[31]. Since the boundedness consumption is made in Assumption II.1, the problem (1) is lower bounded, which means we can omit dual infeasility detection and use primal infeasibility detection only.

A. Termination Criteria

We define the primal and dual residuals

$$r_{\text{prim}} := [Ax + s_c - b; x - s_b] \tag{31}$$

$$r_{\text{dual}} := Px + q - A^\top y_c + y_b \tag{32}$$

of the problem. As the scaling is not implemented to problem (1), We terminate the relaxed convex problem of each node if:

$$||r_p^k||_\infty \le \epsilon_{\text{abs}} \tag{33}$$

$$||r_d^k||_\infty \le \epsilon_{\text{abs}}, \tag{34}$$

without the relative dependence on each term inside (31), (32) as in [29], [31].

B. Infeasibility Detection

As in [29], primal infeasibility is detected if there exists a direction such that the dual cost in (13a) would increase to infinity along this direction. Hence, the set that provides primal infeasibility of each node is defined as

$$\mathcal{D} = \{(y_c, y_b) \in \mathbb{R}^m \times \mathbb{R}^n |\ A^\top y_c - y_b = 0, \sigma_{\mathcal{C}}(y_c) + \sigma_{\mathcal{B}}(y_b) - b^\top y_c < 0\} \tag{35}$$

The existence of some $(y_c, y_b) \in \mathcal{D}$ certifies that problem (1) is primal infeasible, even if we omit Assumption II.1. Furthermore, [30] showed that the successive differences

$$\begin{aligned} \delta x^k = x^k - x^{k-1}, \delta s_c^k = s_c^k - s_c^{k-1}, \delta s_b^k = s_b^k - s_b^{k-1}, \\ \delta y_c^k = y_c^k - y_c^{k-1}, \delta y_b^k = y_b^k - y_b^{k-1} \end{aligned}$$

converge. Hence, for primal infeasible problems in node solving, $(\delta y_c, \delta y_b) = \lim_{k\to\infty}(\delta y_c^k, \delta y_b^k)$ will satisfy condition (35).

Since the complexity of the correction step (17) is equivalent to solving a linear system (18) and requires computation of the dual cost (21). The primal infeasibility is claimed if

$$\begin{aligned} ||A^\top \delta y_c^k - \delta y_b^k||_\infty / ||(\delta y_c^k, \delta y_b^k)||_\infty \le \epsilon_{\text{p,inf}}, \\ \sigma_{\mathcal{C}}(\delta y_c^k) + \sigma_{\mathcal{B}}(\delta y_b^k) - b^\top \delta y_c^k \le \epsilon_{\text{p,inf}}. \end{aligned}$$