A Unified Early Termination Technique for Primal-dual Algorithms in Mixed Integer Conic Programming

Yuwen Chen*, Catherine Ning, Paul Goulart

***Abstract*—We propose an early termination technique for mixed integer conic programming within branch-and-bound based solvers. Our approach generalizes previous early termination results for ADMM-based solvers to a broader class of primal-dual algorithms, including both operator splitting and interior point methods. The complexity for checking early termination is $\mathcal{O}(n)$ for each termination check assuming a bounded problem domain. We show that this domain restriction can be relaxed for problems whose data satisfies a simple rank condition, in which case each check requires an $\mathcal{O}(n^2)$ solve using a linear system that is factored only once at the root node. We further show how this approach can be used in hybrid model predictive control problems with bounded inputs. Numerical results show that our method leads to a moderate reduction in the computational time required for branch-and-bound conic solvers with interior-point based subsolvers.**

***Index Terms*—Optimization algorithms, mixed integer programming, model predictive control.**

I. Introduction

A. Literature review

Mixed integer conic programming (MICP) is a powerful tool for modelling many real-world applications, e.g. hybrid model predictive control [1], portfolio optimization [2], power electronics [3] and robust truss topology [4]. The branch-and-bound (B&B) method is the technique most commonly used to search for optimal solutions in mixed integer programming (MIP). It solves a sequence of relaxed convex subproblems, while the number of such problems increases exponentially w.r.t. the number of integer variables.

Many techniques have been developed to speed up MIP computation. Cutting plane methods are widely used and can significantly reduce the number of nodes that a B&B solver must visit. Presolving [5] is a collection of problem reduction operations applied before solving an MIP, including bound strengthening, coefficient strengthening, constraint reduction and conflict analysis. In addition to presolving an MIP, one can also apply many heuristic methods to accelerate the computation. Most acceleration methods can be broadly classified into two types, start and improvement heuristics [6], both of which are crucial for pruning nodes in B&B algorithms. Start heuristics aim to find a feasible solution as early as possible when the B&B algorithm starts, e.g. feasibility pump methods [7]. On the other hand, improvement heuristics search for feasible points of better objective value based on information from feasible points already obtained, e.g. RINS [8] and the crossover method [9].

Pruning is usually an effective method to reduce the total number of nodes to be solved in B&B. Suppose U is the upper bound corresponding to the value of the best integer feasible solution so far. After updating the upper bound U with a new integer feasible point, one can prune any unevaluated nodes that are known to have an optimal value or a lower bound that is greater than U. Consequently, if a dual feasible point of a relaxed problem within a B&B search can be generated prior to convergence with its dual objective already larger than the current upper bound U, then one can stop the node computation immediately before solving it to optimality. This is called *early termination* and has been implemented in dual feasible algorithms like active-set methods [10], [11], [12], [13]. However, many central ideas in dual feasible methods, such as the use of basic feasible solutions, are not easily generalizable to conic programming.

At the heart of any B&B method is an optimization algorithm for solving convex problems. Many state-of-the-art conic optimization algorithms are primal-dual methods, and most can be classified into two types: second-order methods such as the interior point method (IPM) [14], and first-order methods such as the operator splitting method (OSM) [15]. Both of them start from an infeasible initial point, and attain a feasible point when the algorithm converges to a global optimum and generate a certificate of infeasibility otherwise. This makes early termination difficult since primal-dual methods do not typically reach a dual feasible point until the algorithm converges at optimality. Recently, [16] proposed a heuristic method to generate a dual feasible point for a specialized primal-dual IPM, but the feasibility of dual iterates is still not theoretically guaranteed and it applies only to mixed-integer quadratic programming.

B. Contributions and organization

In this paper we generalize an early termination strategy for MICP, initially proposed for ADMM [17], to any primal-dual optimization method. We develop efficient methods to find a dual feasible point for early termination at each iteration. We relax the boundedness assumption in [17] to a more general rank condition on the problem data that is applicable to many real-world scenarios. We propose a simple correction step that costs $\mathcal{O}(n)$ flops for bounded problems, and a more general optimization-based one costing $\mathcal{O}(n^2)$ flops at each iteration once we obtain a factorization at the start of an MICP. Both costs are relatively small compared to the factorization time $\mathcal{O}(n^3)$ per iteration in IPMs and no worse than the per iteration cost of OSMs. We also show that mixed-integer model predictive control (MIMPC) with bounded input satisfies the condition for the optimization-based correction.

Section II provides background on conic optimization. Section III presents our early termination strategy for mixed

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

integer conic programming and describes how to implement it in both OSMs and IPMs. Section IV discusses the algorithmic structure and the computation complexity of early termination. Numerical results are shown in Section V and conclusions are summarized in Section VI.

C. Notation

We denote the set of $n \times n$ symmetric matrices by $\mathbb{S}^n$ and the set of positive semidefinite matrices by $\mathbb{S}^n_+$. We denote the number of elements in the discrete set $\mathbb{I}$ as $|\mathbb{I}|$. The norm $\|\cdot\|$ is the Euclidean norm. The projection of $x \in \mathbb{R}^n$ onto the set $\mathcal{C}$ is denoted $\Pi_{\mathcal{C}}(x)$. The support function of $\mathcal{C}$ is

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

We denote the dual cone $\mathcal{K}^*$ and polar cone $\mathcal{K}^\circ$ of a convex cone $\mathcal{K}$ by $\mathcal{K}^* := \{y \in \mathbb{R}^n \mid \sup_{x \in \mathcal{K}} \langle x, y \rangle \geq 0\}$, and $\mathcal{K}^\circ := \{y \in \mathbb{R}^n \mid \sup_{x \in \mathcal{K}} \langle x, y \rangle \leq 0\}$, respectively.

II. Background

A. Problem formulation

We will consider MICPs in the general form:

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

where $G \in \mathbb{R}^{p \times n}, A \in \mathbb{R}^{m \times n}, h \in \mathbb{R}^p, b \in \mathbb{R}^m$ and $\mathcal{K}$ is a proper cone. The vector $x \in \mathbb{R}^n$ is the decision variable with interval bounds defined by $\bar{l}, \bar{u} \in \mathbb{R}^n$, 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 $q \in \mathbb{R}^n$. We denote the continuous relaxation of (1) within a B&B solver as

$$\begin{aligned} \min_{x,s} \quad & \frac{1}{2} x^\top P x + q^\top x \\ \text{s.t.} \quad & Gx = h \qquad\qquad \mathrm{CP}(l,u) \\ & Ax + s = b, \ s \in \mathcal{K}, \\ & l \leq x \leq u, \end{aligned} \tag{2}$$

where the integer relaxation of $\mathcal{Z}$ is incorporated into the box constraint $\bar{l} \leq l \leq x \leq u \leq \bar{u}$.

B. Dual form for OSMs

Following [17], the dual of the continuous relaxation (2) is

$$\begin{aligned} \max_{x,y,y_b,z} \quad & -\frac{1}{2} x^\top P x - h^\top z + b^\top y - \sigma_{[l,u]}(y_b) \\ \text{s.t.} \quad & Px + q + G^\top z - A^\top y + y_b = 0, \\ & x \in \mathbb{R}^n, y \in \mathcal{K}^\circ, y_b \in \mathbb{R}^n, z \in \mathbb{R}^p, \end{aligned} \tag{3}$$

where the support function $\sigma_{[l,u]}(y_b)$ is explicit, i.e.

$$\sigma_{[l,u]}(y_b) = u^\top y_b^+ + l^\top y_b^-, \tag{4}$$

where $y_b^+ = \max\{y_b, 0\}, y_b^- = \min\{y_b, 0\}$, which is suitable to generate a correction for early termination of any MIP based on an OSM solver, e.g. OSQP [18] and PDHG [19].

C. Dual form for primal-dual IPMs

For IPMs that rely on logarithmically homogeneous self-concordant barrier (LHSCB) functions [14], there is no standard explicit barrier function for box constraints. We instead reformulate the box constraint $l \leq x \leq u$ into two nonnegative inequalities $x \geq l, x \leq u$ that have well-defined barrier functions, and obtain the alternative dual formulation:

$$\begin{aligned} \max_{x,y,y_+,y_-,z} \quad & -\frac{1}{2} x^\top P x - h^\top z - b^\top y - u^\top y_+ + l^\top y_- \\ \text{s.t.} \quad & Px + q + G^\top z + A^\top y + y_+ - y_- = 0 \\ & x \in \mathbb{R}^n, y \in \mathcal{K}^*, \ y_- \geq 0, y_+ \geq 0, z \in \mathbb{R}^p, \end{aligned} \tag{5}$$

where $\mathcal{K}^* = -\mathcal{K}^\circ$ for a proper cone $\mathcal{K}$. If we define $y_b := y_+ - y_-$ for (5), then we find that the dual form for IPMs (5) is equivalent to its counterpart (3) for OSMs. We can therefore design a unified dual correction mechanism for both IPMs and OSMs, which we describe in Section III.

The primal-dual IPM typically requires factorization of a matrix in the form

$$K := \begin{bmatrix} P & G^\top & A^\top \\ G & 0 & 0 \\ A & 0 & -H^k \end{bmatrix} \tag{6}$$

to compute the search direction for every iteration k, where H^k is a scaling matrix that depends on the choice of cones but which is always positive semidefinite. By adding small perturbation to diagonals of K, the matrix can become quasi-definite and be factorized by LDL decomposition with complexity $\mathcal{O}((n+p+m)^3)$ [20], [21]. An IPM always generates a sequence $(x^k, s^k, z^k, y^k, y_b^k)$ such that $s^k \in \mathcal{K}$ and $y^k \in \mathcal{K}^*$, which is the same as in OSMs [18], [19].

D. Branch and bound

The B&B method computes an optimal solution x in (1) by exploring different integer combinations in a tree. It repeatedly branches on entries of x in the integer index set $\mathbb{I}$ and solves continuous convex relaxation subproblems in the form of (2) until a global optimizer is found. Meanwhile, B&B always maintains a global upper bound U corresponding to the value of the best integer feasible solution of (1) found so far. This upper bound is very useful in pruning unsolved nodes, and we will explore early evaluation of this bound inside each convex subproblem in the rest of the paper.

III. Early Termination for Primal-dual Algorithms

In this section we first review the early termination technique we proposed in [17], arguing that it is also applicable in other OSMs (Section III-A), and then tailor it for primal-dual IPMs (Section III-B). We also relax our boundedness assumption and improve the optimization-based correction discussed in [17] (Section III-C) and discuss how we can apply it to hybrid MPC problems (Section III-D).

The key to our proposed early termination method is to utilize the current dual iterate, which has a conic feasible y^k from a primal-dual algorithm (either an OSM or an IPM), and then *remove linear dual residuals by adding corrections* *to unconstrained dual variables y_b in* (3) *or* (5). We thereby obtain a dual feasible solution for (3) or (5) and generate the corresponding dual cost for early termination. To ensure our early termination always works, we first make the following boundedness assumption as in [17]:

Assumption 3.1: The domain of x in the MIP relaxation (2) is bounded, i.e. $l, u \in \mathbb{R}^n$ are both finite.

The assumption is applicable to many real world scenarios, e.g. x is an 0-1 switching signal or subjected to some physical limitations, like in some QP problems where $\|x\|$ is bounded. We will show how to relax this assumption in Section III-C.

A. Correction for OSMs

ADMM can generate iterates $y^k \in \mathcal{K}^\circ, \forall k \geq 0$ in [22]. For any dual iterates (x^k, y^k, y_b^k, z^k) generated by ADMM, we can offset the residual

$$r^k := Px^k + q + G^\top z^k - A^\top y^k + y_b^k \tag{7}$$

by setting $\Delta y_b^k = -r^k$ so that $(x^k, y^k, y_b^k + \Delta y_b^k, z^k)$ is a dual feasible point for (3), which is suitable for the early termination technique proposed in [17]. A useful property of ADMM is that every y^k is in the conic constraint set $\mathcal{K}^\circ$. However, such a property can be generalized to any OSM because we always tackle a conic constraint $s \in \mathcal{K}$ by either projection to the polar cone $\mathcal{K}^\circ$, i.e. $\Pi_{\mathcal{K}^\circ}(v^k)$, or projection to $\mathcal{K}$, i.e. $\Pi_{\mathcal{K}}(v^k)$. The former is what we want for early termination directly, as in a primal-dual hybrid gradient (PDHG) solver [19]. For the latter, due to the Moreau decomposition [23, §2.5],

$$v = \Pi_{\mathcal{K}}(v) + \Pi_{\mathcal{K}^\circ}(v), \ \forall v, \tag{8}$$

we can generate an "equivalent" dual iterate $(I - \Pi_{\mathcal{K}})(v) \in \mathcal{K}^\circ$, which gives the y^k we obtain in ADMM [17]. We can therefore generalize the early termination method we proposed in [17] to any OSM within a B&B solver.

B. Correction for primal-dual IPMs

The main idea behind our correction strategy is to adjust the iterate $(x^k, y^k, y_+^k, y_-^k, z^k)$ to be dual feasible via the correction on the dual of box constraint. A similar idea can be applied to primal-dual IPMs, which also generate dual-feasible conic iterates y^k for every iteration k. Suppose we define $\Delta y_b := \Delta y_+ - \Delta y_-$ with $\Delta y_+, \Delta y_- \geq 0$ for the IPM dual formulation (5). We can verify Δy_b is an unconstrained variable for the dual correction. If we only make corrections on y_- and y_+, leaving other variables fixed, then the change of dual cost in (5) becomes

$$\begin{aligned} -\Delta y_+^\top u + \Delta y_-^\top l &= \Delta y_+^\top (l - u) + (\Delta y_- - \Delta y_+)^\top l \\ &= \Delta y_+^\top (l - u) - \Delta y_b^\top l. \end{aligned} \tag{9}$$

Note that we have $\Delta y_+^\top (l-u) \leq 0$ due to $\Delta y_+ \geq 0, l - u \leq 0$. Meanwhile, the linear residual is

$$r^k := Px^k + q + G^\top z^k + A^\top y^k + y_+^k - y_-^k \tag{10}$$

before the correction. To maximize the dual objective in (5) given $\Delta y_b^k = -r^k$, we set $\Delta y_+^k, \Delta y_-^k$ as

$$\Delta y_+^k = \max\{0, \Delta y_b^k\}, \quad \Delta y_-^k = \Delta y_+^k - \Delta y_b^k. \tag{11}$$

Hence, $(x^k, y^k, y_+^k + \Delta y_+^k, y_-^k + \Delta y_-^k, z^k)$ is a dual feasible point and we can enable early termination checking via (5).

C. Optimization-based correction

In Section III-A and III-B we applied a correction to potentially every entry of y_+^k, y_-^k to ensure dual feasibility, which explains the need for Assumption 3.1. However, the crux of our early termination strategy is to offset the linear residual r^k in (10) via corrections on unconstrained dual variables, which means we can exploit other dual variables beyond box constraints. If we allow for corrections to the unconstrained dual variables x, y_b, z in early termination, then Assumption 3.1 can be generalized to the following:

Assumption 3.2: $[P, I_{\mathcal{B}}^\top, G^\top]$ has rank n, i.e. full row-rank, where $\mathcal{B}$ is the set of entries that have explicit bounded constraints $l_{\mathcal{B}} \leq x_{\mathcal{B}} \leq u_{\mathcal{B}}$ and $I_{\mathcal{B}}$ is the incidence matrix from the span of x to entries in $\mathcal{B}$, i.e. $x_{\mathcal{B}} = I_{\mathcal{B}} x$.

Given Assumption 3.2 we can always generate a dual feasible correction $(\Delta x^k, \Delta y_b^k, \Delta z^k)$ since the linear system

$$P\Delta x^k + I_{\mathcal{B}}^\top \Delta y_{\mathcal{B}}^k + G^\top \Delta z^k = -r^k. \tag{12}$$

always has a solution. It is also a generalization for setting $\Delta y_b^k = -r^k$ discussed in Section III-B, which is useful if some entries of l, u for box constraints are infinite or the difference $u - l$ is so large that the corrected dual cost is excessively sensitive to the correction Δy_b^k.

Due to the existence of different coefficients for the support function $\sigma_{[l,u]}(y_b)$ in (3), or $-u^\top y_+ + l^\top y_-$ in (5), we divide the optimization-based correction into two steps. For the first step, we solve the optimization problem

$$\begin{aligned} \min_{\Delta x^k, \Delta z^k, \Delta y_{\mathcal{B}}^k} \ & \frac{1}{2} \Delta x^{k\top} P \Delta x^k + (Px^k)^\top \Delta x^k + h^\top \Delta z^k \\ & + \frac{\eta}{2} \|\Delta y_{\mathcal{B}}^k\|^2 + \frac{\gamma}{2} \|\Delta z^k\|^2 \\ \text{s.t.} \ \ & P\Delta x^k + I_{\mathcal{B}}^\top \Delta y_{\mathcal{B}}^k + G^\top \Delta z^k = -r^k, \end{aligned} \tag{13}$$

which produces a correction $(\Delta x^k, \Delta y_{\mathcal{B}}^k, \Delta z^k)$ while maximizing the corrected dual cost w.r.t. $\Delta x^k, \Delta z^k$ with regularizations for $\Delta y_{\mathcal{B}}^k, \Delta z^k$. The corresponding KKT condition of (13) is

$$\begin{bmatrix} P & I_{\mathcal{B}}^\top & G^\top \\ I_{\mathcal{B}} & -\eta I & 0 \\ G & 0 & -\gamma I \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta y_{\mathcal{B}}^k \\ \Delta z^k \end{bmatrix} = \begin{bmatrix} -r^k \\ -I_{\mathcal{B}} x^k \\ h - Gx^k \end{bmatrix} \tag{14}$$

if we set $\lambda^k = x^k + \Delta x^k$. The matrix on the left-hand side does not depend on the active node, and hence only needs to be factored once at the initialization of an MIP solver and can be reused later for any node's computation. Meanwhile, solving (14) is computationally efficient compared to the factorization step (6) of an IPM in every iteration, or not worse than the computation of an OSM per iteration. For the second step, we complete Δy_b^k by setting $\Delta y_j = 0$ for any index $j \notin \mathcal{B}$. If an IPM is used, we compute $\Delta y_+^k, \Delta y_-^k$ from Δy_b^k as described in (11) Section III-B, and $(x^k + \Delta x^k, y^k, y_+^k + \Delta y_+^k, y_-^k + \Delta y_-^k, z^k + \Delta z^k)$ is a dual feasible point for early termination.

D. Applications in control

A common type of MIP arising in control engineering is optimal control with discrete-valued inputs as encountered in hybrid MPC problems, which takes the form:

$$\begin{aligned} \min_{x,u} \quad & \sum_{t=0}^{T-1} (x_t^\top Q_t x_t + u_t^\top R_t u_t) + x_T^\top Q_T x_T + 2q_T^\top x_T \\ \text{s.t.} \quad & x_{t+1} = \bar{A} x_t + \bar{B} u_t,\ x_0 = x_{init}, \\ & u_t \in \mathcal{U}_t, \quad \forall t = 0, 1, \dots, T-1, \end{aligned} \tag{15}$$

where $x_{init} \in \mathbb{R}^{n_x}$ is the initial state, $(\bar{A}, \bar{B})$ models the system dynamics, and $\mathcal{U}_t$ describes the constraints for each input $u_t \in \mathbb{R}^{n_u}$. The set $\mathcal{U}_t$ can be composed entirely or in part by discrete valued constraints. Our optimization-based correction is suitable for the hybrid-MPC (15) due to the following theorem:

Theorem 3.3: Assumption 3.2 is satisfied in the hybrid MPC problem (15) when $\mathcal{U}_t$ is bounded for $t = 0, \dots, T-1$.

Proof: Suppose $x := [x_0; \dots; x_T; u_0; \dots; u_{T-1}]$. The corresponding block components of $[P, I_{\mathcal{B}}^\top, G^\top]$ become

$$P = \begin{bmatrix} Q & \\ & R \end{bmatrix}, I_{\mathcal{B}} = \begin{bmatrix} 0_{n_u T \times n_x(T+1)} & I_{n_u T} \end{bmatrix},$$

$$G = \left[\begin{array}{ccccc|cccc} I & & & & & 0 & & & \\ \bar{A} & -I & & & & \bar{B} & & & \\ & \bar{A} & -I & & & & \ddots & & \\ & & \ddots & \ddots & & & & \ddots & \\ & & & \bar{A} & -I & & & & \bar{B} \end{array}\right],$$

where $Q = \text{diag}(Q_0, \dots, Q_T), R = \text{diag}(R_0, \dots, R_{T-1})$ are block diagonal. Hence, $[P, I_{\mathcal{B}}^\top, G^\top]$ can be reordered into an upper triangular matrix in the form

$$\left[\begin{array}{cccccc|cc} I_{n_x} & & \bar{A}^\top & & & & \vdots & \vdots \\ & I_{n_u} & \bar{B}^\top & & & & \vdots & \vdots \\ & & -I_{n_x} & & & & \vdots & \vdots \\ & & & \ddots & & \bar{A}^\top & \vdots & \vdots \\ & & & & I_{n_u} & \bar{B}^\top & \vdots & \vdots \\ & & & & & -I_{n_x} & \vdots & \vdots \end{array}\right].$$

The matrix above is full row rank since every diagonal term is either 1 or -1. Hence, $[P, I_{\mathcal{B}}^\top, G^\top]$ is full row rank and the Assumption 3.2 is satisfied. ∎

Note that the system (14) is also banded for the sparse formulation (15) and we can exploit its structure to accelerate the computation as in [24], which reduces the cost per iteration from $\mathcal{O}((m_x N)^3)$ to $\mathcal{O}(N(n_x + n_u)^3)$. Moreover, our method can be applied directly to outer approximation (OA) [25] if the OA only replaces the conic constraint $\mathcal{K}$ with some linear constraints without introducing new variables. Otherwise, we can combine it with bound strengthening techniques [5] to satisfy Assumption 3.2.

IV. Algorithm and Complexity of Computation

We next summarize how to implement early termination in a B&B method, which corresponds to steps 3-17 in Algorithm 1. For every iteration k in a node CP$(\underline{x}, \bar{x})$, we can obtain a primal-dual iterate $(x^k, s^k, y^k, y_b^k, z^k)$ from an OSM or an IPM with an approximate dual cost D^k. Note that this iterate is conic feasible but doesn't satisfy the dual linear constraint, i.e. (3) or (5). We then check whether the algorithm finds an optimal solution $\hat{x}$ or detects the infeasibility of CP$(\underline{x}, \bar{x})$ (steps 5-10). These steps are inherent to a primal-dual algorithm even without early termination and do not incur any additional cost. We then activate early termination when we find the approximate dual cost is larger than the current upper bound, i.e. $D^k \geq U$ (step 11). This heuristic follows [16] since D^k is close to the optimal solution of CP$(\underline{x}, \bar{x})$ when the dual linear residual r^k is small enough, and can save computation time on early termination.

Once early termination is enabled, we then compute a feasible correction $(\Delta x^k, \Delta y_b^k, \Delta z^k)$ using one of the methods discussed in Section III-A, III-B or Section III-C and compute the dual cost $\underline{D}^k$ at the dual feasible point $(x^k + \Delta x^k, y^k, y_b^k + \Delta y_b^k, z^k + \Delta z^k)$ (step 12). If $\underline{D}^k > U$, we know the optimum of CP$(\underline{x}, \bar{x})$ is larger than $\underline{D}^k$ due to weak duality, and hence larger than U, which indicates we can stop the node computation and prune this node immediately (step 14). Otherwise, we continue computing until we solve CP$(\underline{x}, \bar{x})$ and proceed with the standard B&B method (steps 18-27).

Algorithm 1 B&B for MICP with early termination

Require:
Initialization: $U \leftarrow +\infty$, node tree $\mathcal{T} \leftarrow$ CP(l, u)

```
1:  while T ≠ ∅ do
2:    Pick and remove CP(x̲, x̄) from T
3:    for k = 1, 2 ... do
4:      Generate (x^k, s^k, y^k, y_b^k, z^k) and an estimated dual
        cost D^k from OSMs or IPMs
5:      if termination criteria is satisfied then
6:        return optimal solution x̂ = x^k and f(x̂)
7:      end if
8:      if infeasibility of CP(x̲, x̄) is detected then
9:        return CP(x̲, x̄) infeasible
10:     end if
11:     if D^k ≥ U then
12:       Compute the corrected dual cost D̲^k via (x^k +
          Δx^k, y^k, y_b^k + Δy_b^k, z^k + Δz^k)
13:       if D̲^k ≥ U then
14:         return CP(x̲, x̄) terminates early
15:       end if
16:     end if
17:   end for
18:   if CP(x̲, x̄) terminates early or is infeasible then
19:     prune current node
20:   else if f(x̂) > U then
21:     prune current node
22:   else if x̂ is integer feasible then
23:     U ← f(x̂), x* ← x̂
24:     prune nodes in T with lower bound > U
25:   else
26:     branch node CP(x̲, x̄)
27:   end if
28: end while
```

Let us now consider the computational complexity of early termination. The estimated dual cost D^k, the iterate (x^k, y^k, y_b^k, z^k) and the residual r^k (10) are already computed from a primal-dual algorithm and therefore do not incur any extra cost for checking early termination. We recompute corrections $\Delta x, \Delta z, \Delta y_b$ and the corrected cost $\underline{D}^k$ at every time we check for early termination. Indeed, we compute the cost change $\Delta D^k := \underline{D}^k - D^k$ first and then the corrected cost via $\underline{D}^k = D^k + \Delta D^k$, which is more efficient than computing $\underline{D}^k$ directly. The correction (9) requires an additional $\mathcal{O}(n)$ flops to generate a feasible dual cost. For an optimization-based correction (13), we need no more than $\mathcal{O}((2n+p)^2)$ flops to solve the linear system (14) if we save the factorization of the matrix in (14) from the start of an MICP. Both correction costs are relatively small compared to the $\mathcal{O}(n+p+m)^3$ flops per IPM iteration. For an OSM, each early termination check is no more costly than the original computation per iteration, so that its computational time is negligible inside every M iterations, e.g. $M = 25$ in OSQP [26].

V. Numerical Results

We implement Algorithm 1 and a counterpart without early termination, i.e. removing steps 3-17 in Algorithm 1. Both were written in Julia with every convex relaxation solved by the IPM solver `Clarabel.jl` [21]. Tests are implemented on Intel Core i7-9700 CPU @3.00GHz, 16GB RAM. Experiments for OSMs can be found in [17].

A. Mixed integer model predictive control

We next consider the following hybrid MPC problem current reference tracking in power electronics [3]:

$$\begin{aligned} \min_{x,u} \quad & \sum_{t=0}^{T-1} \gamma^t l(x_t) + \gamma^T V(x_T) \\ \text{s.t.} \quad & x_0 = x_{\text{init}}, \\ & x_{t+1} = \bar{A}x_t + \bar{B}u_t, \ ||u_t - u_{t-1}||_\infty \leq 1, \\ & u_t \in \{-1, 0, 1\}^6, \ \forall t = 0, 1, \dots, T-1, \end{aligned} \tag{16}$$

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 computed using approximate dynamic programming. The initial state is x_{init} and the system dynamics is $x_{t+1} = \bar{A}x_t + \bar{B}u_t$ with $x_t \in \mathbb{R}^{12}$ representing the internal motor currents, voltages and the input $u_t \in \mathbb{R}^6$ including three semiconductor devices positions with integer values $\{-1, 0, 1\}$ and three additional binary components required to model the system. The ramp rate constraint $\|u_t - u_{t-1}\|_\infty \leq 1$ 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, (16) reduces to a problem depending only on the input variables $u_0, \dots, u_{T-1}$ and the initial state x_0; we refer readers to [3] for details. We set $T = 8$ for the time horizon and simulate closed-loop MIMPC for 100 consecutive intervals. Figure 1 compares the performance of B&B with and without early termination. We apply the simple early termination introduced in Section III-B. We start to count time only after the first feasible solution of (16), and consequently a finite upper bound U is found. This ensures that the reductions show in Figures 1 and 2 represent the reduction in computation cost relative to an idealized omniscient early termination scheme. For all 100 intervals, early termination has produced a noticeable reduction in computational time, averaging to about 20%.

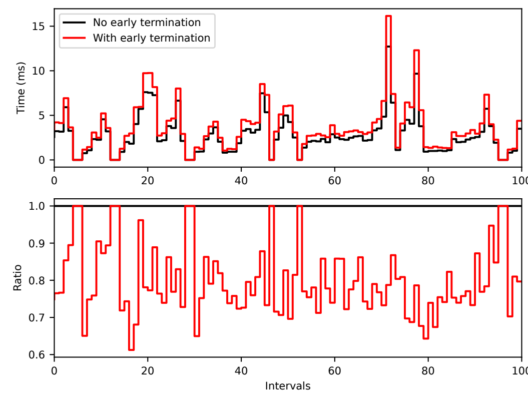

Figure 1: MIMPC $T = 8$, reduced dense form

We implement another experiment for (16) using the sparse form discussed in Section III-D with the optimization-based correction of $\eta = \gamma = 1$. Figure 2 shows it also reduces the computational time about 15% to 20% over 100 intervals.

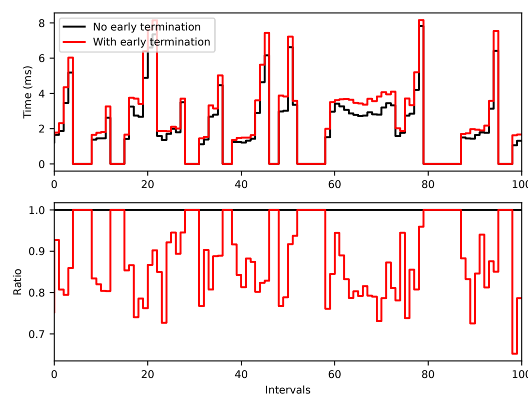

Figure 2: MIMPC $T = 8$, sparse form

B. Portfolio optimization

We also test our proposed early termination technique on a portfolio optimization problem, which can be formulated as a mixed integer second-order cone (SOC) programming [2],

$$\begin{aligned} \min_{x,b,l} \quad & r^\top x \\ \text{s.t.} \quad & x^\top \Lambda x \leq \rho, \\ & \sum\nolimits_{i=1}^n x_i = 1, \quad \sum\nolimits_{i=1}^n b_i \leq K, \\ & L_{min} \leq \sum\nolimits_{i=1}^n l_i \leq L_{max}, \ b \leq Hl, \ l \leq H^\top b, \\ & l_j \in \{0, 1\}, \ \text{for } j \in \{1, ..., L\} \\ & -b_i \leq x_i \leq b_i, b_i \in \{0, 1\}, i \in \{1, ..., n\}. \end{aligned}$$

There are n assets in total, categorized into L industry sectors, with the mapping from assets to sectors captured by matrix

$H \in \mathbb{R}^{n \times L}$. We define $x \in \mathbb{R}^n$ as the fractions of portfolio value held in each asset: $x_i > 0$ and $x_i < 0$ denote buying and selling (i.e. shorting) respectively, and must sum to unity. The vector $r \in \mathbb{R}^n$ is the expected return for n assets, and Λ is the covariance for market volatility and restricted below a certain level ρ, which is formulated as a SOC constraint. The binary vectors $b \in \mathbb{R}^n$ and $l \in \mathbb{R}^L$ denote whether we invest in an asset, respectively in a sector or not. The number of assets we can invest in is upper-bounded by K and the number of sectors is limited to $[L_{min}, L_{max}]$.

We use the early termination strategy as in Section III-B and choose $n = 20, L = 3, L_{min} = 1, L_{max} = L, \rho = 100, K = 10$ and simulate the portfolio problem over 100 consecutive days. We use data on returns from the S&P 500 for the period 2015-2020 grouped according to GICS sector. Estimates for r and Λ were computed using standard statistical methods; see [27, §13]. Figure 3 shows the early termination can reduce computation time about 10%-15% after we find the first integer feasible solution, which shows that our early termination method can also be effective for MICPs.

VI. Conclusion

We generalized our early termination technique of ADMM in [17] to primal-dual algorithms including operator splitting methods and interior point methods in MICPs. We showed how to utilize existing dual iterates inside either an OSM or an IPM to generate a dual feasible point for early termination with little additional efforts, and we provided two sufficient conditions for two proposed early termination techniques respectively. We also show that the optimization-based correction can be directly applied to an MIMPC if the input is bounded. Numerical results showed the proposed early termination can reduce the total computational time in MICPs effectively.

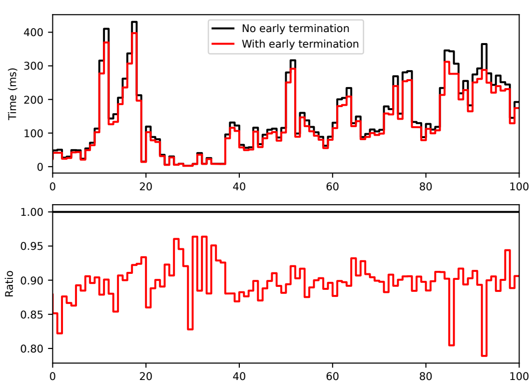

Figure 3: Portfolio Optimization

References

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

[2] H. Y. Benson and Ümit Sağlam, *Mixed-Integer Second-Order Cone Programming: A Survey*. INFORMS, 2014, pp. 13–36.

[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] 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.

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

[7] 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.

[8] 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.

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

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

[11] 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.

[12] 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.

[13] 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.

[14] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994.

[15] 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.

[16] 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.

[17] Y. Chen and P. Goulart, "An early termination technique for ADMM in mixed integer conic programming," in *European Control Conference (ECC)*, 2022, pp. 60–65.

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

[19] D. Applegate, M. Díaz, O. Hinder, H. Lu, M. Lubin, B. O'Donoghue, and W. Schudy, "Practical large-scale linear programming using primal-dual hybrid gradient," *Advances in Neural Information Processing Systems*, vol. 34, pp. 20 243–20 257, 2021.

[20] A. Domahidi, E. Chu, and S. Boyd, "ECOS: An SOCP solver for embedded systems," in *2013 European Control Conference (ECC)*. IEEE, 2013, pp. 3071–3076.

[21] P. Goulart and Y. Chen. (2022) Clarabel.jl. [Online]. Available: https://oxfordcontrol.github.io/ClarabelDocs/stable/

[22] 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.

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

[24] C. V. Rao, S. J. Wright, and J. B. Rawlings, "Application of interior-point methods to model predictive control," *Journal of Optimization Theory and Applications*, vol. 99, no. 3, pp. 723–757, 1998.

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

[26] 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.

[27] R. J. Vanderbei, *Linear Programming: Foundations and Extensions*. Springer, 2020.