Mix-CALADIN: A Distributed Algorithm for Consensus Mixed-Integer Optimization

Boyu Han [*] **Xu Du** [*] **Karl H. Johansson** [**] **Apostolos I. Rikos**[† *,***]

[*] *The Artificial Intelligence Thrust of the Information Hub, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China, (e-mail: {boyuhan,michaelxudu,apostolosr}@hkust-gz.edu.cn).*
[**] *The Division of Decision and Control Systems, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden, (e-mail: kallej@kth.se)*
[***] *The Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong, China.*

Abstract: This paper addresses distributed consensus optimization problems with mixed-integer variables, with a specific focus on Boolean variables. We introduce a novel distributed algorithm that extends the Consensus Augmented Lagrangian Alternating Direction Inexact Newton (CALADIN) framework by incorporating specialized techniques for handling Boolean variables without relying on local mixed-integer solvers. Under the mild assumption of Lipschitz continuity of the objective functions, we establish rigorous convergence guarantees for both convex and nonconvex mixed-integer programming problems. Numerical experiments demonstrate that the proposed algorithm achieves competitive performance compared to existing approaches while providing rigorous convergence guarantees.

Keywords: Mixed Integer Programming, Distributed Optimization, Consensus ALADIN

1. INTRODUCTION

Mixed-integer programming (MIP) has attracted increasing attention in recent years due to its wide range of applications, including express transport network design (Zhu and You (2025)), wireless antenna selection (Zhu and Du (2022)), and optimal demand shut-offs in AC microgrids Du et al. (2023). However, as the scale and complexity of optimization problems continue to grow, solving MIPs on a single computing node has become increasingly impractical (Doostmohammadian et al. (2025)).

Traditional MIP algorithms are primarily centralized and include classical methods such as branch-and-bound, cutting-plane, and branch-and-cut. For a comprehensive overview of these techniques, readers are referred to Wolsey (2020) and Conforti et al. (2014). In addition, several heuristic approaches have demonstrated strong empirical performance without formal convergence guarantees, such as the semidefinite relaxation (SDR) method proposed by Luo et al. (2010) and the ϵ-approximation algorithm for mixed-integer quadratic programming (MIQP) presented by Pia (2023). However, most of the aforementioned studies focus on centralized computation, which often suffers from high memory demands and long computation times in large-scale settings. Therefore, developing distributed algorithms for solving MIPs has become an increasingly urgent and essential research direction.

Distributed optimization algorithms can be broadly categorized into two classes: (i) primal decomposition and (ii) dual decomposition. This paper focuses primarily on solving MIP problems using distributed optimization algorithms of the dual decomposition type.

Existing Literature. Both primal and dual decomposition methods have been widely used for mixed-integer programs (MIPs). In primal decomposition, Kuwata and How (2010) proposed a cooperative distributed algorithm for trajectory optimization using mixed-integer linear programming (MILP), which sequentially optimizes subproblems with low-dimensional parametric perturbations to ensure robustness and monotonic improvement. Similarly, Camisa et al. (2021) developed a distributed MILP framework that guarantees finite-time feasible solutions with low sub-optimality. Testa et al. (2019) achieved convergence by iteratively generating cutting planes and exchanging active constraints. However, these works are limited to MILP. More recently, the Partially Distributed Optimization Algorithm (PaDOA) Murray et al. (2021) extended primal decomposition to mixed-integer convex programs (MICP) by alternating between decoupled mixed-integer subproblems and a centralized MILP master problem, achieving global optimality within ϵ-accuracy. Nonetheless, PaDOA still relies on centralized MILP solvers. Dual-decomposition-based distributed algorithms for mixed-integer programming have received even greater attention. Within this framework, Vujanic et al. (2016) and Falsone

[⋆] [†] Corresponding author
[⋆⋆]The work of B.H., X.D., and A.I.R. was supported by the Guangzhou-HKUST(GZ) Joint Funding Scheme (Grant No. 2025A03J3960). The work of A.I.R. was also supported by the Guangdong Provincial Project (Grant No. 2024QN11G109).

et al. (2019) proposed dual-based decentralized MILP algorithms using adaptive constraint tightening for finite-time feasible convergence. Dong and Xia (2024) extended to mixed-integer nonlinear programming (MINLP) with a convexification procedure. However, all these methods rely on external MIP solvers for subproblems, limiting scalability. As a representative dual decomposition method, ADMM has been widely studied for distributed mixed-integer optimization. Sun et al. (2024) applied ADMM to MILP with rigorous convergence analysis. Zhu and You (2025) used ADMM for express transport network design (MILP). Liu and Stursberg (2022) showed that, with sufficiently large penalty parameters, ADMM achieves zero duality gap for MICP, a strong convergence result. Yet all these methods require solving MIP subproblems, which is computationally expensive. Only a few works avoid centralized MIP solvers: Le and Malikopoulos (2025) addressed an MIQP traffic problem using ADMM with Big-M, but is numerically unstable; Takapoui et al. (2020) directly round integer variables, but are heuristic without convergence guarantees. ALADIN Houska et al. (2016); Du and Wang (2025) incorporates SQP into ADMM, guaranteeing global convergence for convex problems and local convergence for nonconvex ones. Its mixed-integer extensions Murray and Hagenmeyer (2020) provide theoretical guarantees for convex and nonconvex MIPs, yet rely on centralized MIP solvers in practice. To the best of our knowledge, no existing work solves distributed mixed-integer optimization with convergence guarantees without relying on MIP solvers. This motivates a distributed solver that (i) operates without local MIP solvers, and (ii) provides rigorous convergence guarantees for convex and nonconvex MINLPs.

Main Contributions. Motivated by these challenges, we propose Mix-CALADIN, a Consensus-ALADIN-based algorithm for Boolean mixed-integer optimization. To our knowledge, it is the first such method that (i) efficiently solves distributed mixed-integer problems, (ii) operates without local mixed-integer solvers, and (iii) provides local convergence guarantees. Our contributions are as follows.
A. Two-stage algorithm. We propose a novel two-stage distributed optimization algorithm for solving distributed mixed-integer programming problems. Stage I solves a relaxed continuous problem via Consensus ALADIN, yielding a deterministic lower bound and a high-quality initial point for Stage II. Inspired by Hall et al. (2021); Du et al. (2023), Stage II adds, at the coordinator, a constraint keeping relaxed integer variables in $[0, 1]$ and a penalty term driving them toward Boolean values, ensuring convergence.
B. Convergence guarantees. Stage I inherits Consensus ALADIN's properties: global linear convergence for convex L-smooth problems (Du and Wang, 2025, Theorem 2) and local linear convergence for nonconvex problems (Du and Wang, 2025, Theorem 3). For Stage II, under Lipschitz continuity, convergence holds when the augmented Lagrangian parameter ρ exceeds the Lipschitz constant (Theorem 1). Numerical simulations confirm the theory.

2. PROBLEM FORMULATION

We consider the following distributed consensus mixed-integer optimization problem with N agents:

$$\begin{aligned} \min_{\substack{x_i,\, z\in\mathbb{R}^{n_c+n_d}\\ \forall i\in\{1,\dots,N\}}} \quad & \sum_{i=1}^{N} f_i(x_i) \\ \text{s.t.} \quad & x_i = z, \end{aligned} \tag{1}$$

where $x_i = [y_i^\top, b_i^\top]^\top, y_i \in \mathbb{R}^{n_c}, b_i \in \{0,1\}^{n_d}$. Each agent i has a local objective function $f_i : \mathbb{R}^{n_c+n_d} \to \mathbb{R}$ and a decision variable $x_i = [y_i^\top, b_i^\top]^\top$, where $y_i \in \mathbb{R}^{n_c}$ are continuous variables and $b_i \in \{0,1\}^{n_d}$ are Boolean variables. The global consensus variable $z = [z_c^\top, z_d^\top]^\top$ comprises continuous components $z_c \in \mathbb{R}^{n_c}$ and Boolean components $z_d \in \{0,1\}^{n_d}$.

3. PRELIMINARIES OF CALADIN

This section provides the preliminary knowledge for our work. We begin with an overview of CALADIN, an algorithm for large-scale distributed consensus optimization with continuous variables, which lays the foundation for our approach.

CALADIN, see Du and Wang (2025), Du et al. (2025b) is a distributed algorithm for solving consensus optimization problems of the form (1) without boolean part, where all local variables x_i and the global consensus variable z are n-dimensional vectors. The augmented Lagrangian is given by: $\mathcal{L}(x, z, \lambda) := \sum_{i=1}^{N} f_i(x_i) + \lambda_i^\top(x_i - z) + \frac{\rho}{2}\|x_i - z\|^2$, where $x = [x_1^\top, \dots, x_N^\top]^\top$ collects the primal variables and $\lambda = [\lambda_1^\top, \dots, \lambda_N^\top]^\top$ collects the dual variables.

The CALADIN algorithm proceeds as follows:

$$\begin{cases} x_i^{[k+1]} = \arg\min\limits_{x_i} f_i(x_i) + \left(\lambda_i^{[k]}\right)^\top x_i \\ \qquad\qquad\qquad + \dfrac{\rho}{2}\|x_i - z^{[k]}\|^2, \quad \forall i \in \mathcal{V}, \\ g_i = \nabla f_i(x_i^{[k+1]}), \quad H_i \approx \nabla^2 f_i(x_i^{[k+1]}) \succ 0, \\ (z^{[k]}, \lambda^{[k]}) = \begin{cases} \min\limits_{\Delta x_i\in\mathbb{R}^n} & \sum\limits_{i=1}^{N} \Delta x_i^\top H_i \Delta x_i + g_i^\top \Delta x_i \\ \quad \text{s.t.} & x_i^{[k+1]} + \Delta x_i = z \mid \lambda_i, \quad \forall i. \end{cases} \end{cases} \tag{2}$$

The algorithm consists of three main steps: (1) each agent minimizes its augmented Lagrangian with respect to local variables; (2) all agents compute local gradients g_i and positive definite Hessian approximations H_i; (3) a coordinator solves a consensus quadratic program (QP) using the collected local information to update the global variable z and dual variables λ_i.

For continuous variables, our previous work Du and Wang (2023, 2025); Du et al. (2025a)) have established local convergence guarantees for nonconvex problems and global convergence for convex problems. The aim of this paper is to extend the CALADIN framework to handle mixed-integer optimization problems as formulated in (1).

4. MIX-CALADIN

This section presents a novel distributed algorithm for solving problem (1).

4.1 Algorithm Development

In this section, we present our proposed two-stage distributed mixed-integer algorithm, detailed below as Algorithm 1.

In Stage I, we consider a continuous relaxation of problem (1), constructed by relaxing the Boolean constraint. Crucially, this relaxed problem yields a theoretically guaranteed lower bound for the original problem (1), and this guarantee holds irrespective of the convexity of the functions f_i. The implementation of Stage I follows the established method described in (2) and is therefore not repeated here. Notably, the coordination step in (5) admits a closed-form solution, which reduces the computational burden compared to the original CALADIN formulation.

Algorithm 1 Mix-CALADIN

Initialization. Randomly choose dual variable $\lambda_i \in \mathbb{R}^{n_c+n_d}$ for each node $i \in \mathcal{V}$, and global variable $z \in \mathbb{R}^{n_c+n_d}$. Set $\beta > 1$.
Stage I Iteration.

1. Paralleled solve local NLP without subsystem coupling:
$$x_i^{[k+1]} = \arg\min_{x_i} f_i(x_i) + (\lambda_i^{[k]})^\top x_i + \frac{\rho_1}{2}\left\|x_i - z^{[k]}\right\|^2 \tag{3}$$
2. Evaluate the Hessian and gradient:
$$g_i = \nabla f_i(x_i^{[k+1]}), \quad H_i \approx \nabla^2 f_i(x_i^{[k+1]}) \succ 0. \tag{4}$$
3. The coordination:
$$\begin{cases} z^{[k+1]} = \left(\sum_{i=1}^N H_i\right)^{-1}\left(\sum_{i=1}^N H_i x_i^{[k+1]} - g_i\right) \\ \lambda_i^{[k+1]} = H_i(x_i^{[k+1]} - z^{[k+1]}) - g_i. \end{cases} \tag{5}$$

If $\|z^{[k+1]} - z^{[k]}\| \leq \epsilon$, then switch to Stage II with $\tilde{z}^* = z^{[k+1]}$
Stage II
Initialization. Set $z^1 = \tilde{z}^*$ from Stage I. Set $\alpha = 1, k = 1$.
Iteration.

1. Inner level: Do iteration k:
   - For each agent: evaluate the gradient and transmit it to the coordinator:
   $$g_i = \nabla f_i(z^{[k]}). \tag{6}$$
   - For coordinator: receive g_i from each agent and transmit $z^{[k]}$ to each agent.
   $$\begin{aligned} z^{[k+1]} = \arg\min_z \quad & \frac{N\rho_2}{2}\left\|z - z^{[k]}\right\|^2 + z^\top \sum_{i=1}^N g_i \\ & + \alpha(\mathbf{1} - 2z_d^{[k]})^\top z_d \\ \text{s.t.} \quad & 0 \leq z_d \leq 1. \end{aligned} \tag{7}$$

   If $\|z^{[k+1]} - z^{[k]}\| \leq \epsilon_{inner}$, then output to the *outer level* $z^* = z^{[k+1]}$. Else set *k=k+1* and go to the *inner level.*
2. Outer level: If $(1 - z_d^*)^\top z_d^* \geq \epsilon_{outer}$, then the coordinator sets $\alpha = \beta\alpha, k = k+1, z^{[k]} = z^*$ and go to the *inner level.* Else terminate operation.
3. Output: z^*.

Stage II of Algorithm 1 is initialized using the solution from Stage I by setting $z^{[1]} = \tilde{z}^*$ and implements a bi-level optimization framework. The inner level is inspired by (Wu et al., 2025, Algorithm 3)), where information exchange is managed solely by a coordinator while each agent performs only gradient evaluation. Specifically, at each iteration, the inner level executes two steps: (i) each agent computes its local gradient g_i at $z^{[k]}$ according to (6), and (ii) the coordinator solves the convex quadratic program in (7) with box constraint $0 \leq z_d \leq 1$, employing the term $\alpha(\mathbf{1}-2z_d^{[k]})^\top z_d$ to drive the continuous variable z_d toward Boolean values as α increases, a strategy motivated by Zhu and Du (2022); Du et al. (2023). The inner level iterates until $\|z^{[k+1]}-z^{[k]}\| \leq \epsilon_{\text{inner}}$, outputting $z^* = z^{[k+1]}$ to the outer level or proceeding to iteration $k+1$ otherwise. In the outer level, if $(\mathbf{1} - z_d^{[k+1]})^\top z_d^{[k+1]} \geq \epsilon_{\text{outer}}$, the penalty parameter is increased via $\alpha \leftarrow \beta\alpha$ ($\beta > 1$) and the inner level is restarted with $k \leftarrow k+1$ and $z^{[k]} = z^*$; otherwise, z^* is returned as the final solution. The theoretical convergence analysis is provided in the subsequent section. Note that, Stage I of Algorithm 1 can be replaced by other distributed optimization algorithms to initialize Stage II. However, since CALADIN offers favorable convergence guarantees and numerical performance for both convex and non-convex problems (Du and Wang (2025)), this paper focuses on CALADIN.

Remark 1. ***Smoothness Approximation for Boolean Variables:*** To enforce convergence of z_d to Boolean solutions, one could incorporate the nonconvex term $\alpha(\mathbf{1} - z_d)^\top z_d$ into (7) with sufficiently large α. However, this approach may lead to numerical instability. Inspired by Hall et al. (2021); Zhu and Du (2022); Du et al. (2023), we instead use the convex approximation $\alpha(\mathbf{1} - 2z_d^{[k]})^\top z_d$ at each iterate $z_d^{[k]}$ to maintain numerical robustness while achieving the desired convergence behavior. Note that the convergence of the centralized linear complementarity QP (LCQP) discussed in Hall et al. (2021) relies on step-size adjustment for the primal variable, i.e., a globalization strategy. In contrast, due to the inherent distributed optimization nature of ALADIN, no such globalization strategy is required. Moreover, aside from Le and Malikopoulos (2025) (numerically unstable) and Takapoui et al. (2020) (heuristic, no convergence guarantee), no existing distributed optimization algorithm operates without a centralized mixed-integer programming solver.

4.2 Convergence Analysis

This section presents the convergence analysis of Algorithm 1. Before stating the convergence theorems, we first introduce the following assumptions, which serve as the foundation for our convergence analysis.

Assumption 1. For each agent i, the local cost function $f_i : \mathbb{R}^n \to \mathbb{R}$ is closed, proper, and L-smooth, i.e., there exists a Lipschitz constant $L > 0$ such that for all $x_a, x_b \in \mathbb{R}^n$:
$$f_i(x_a) \leq f_i(x_b) + \nabla f_i(x_b)^\top (x_a - x_b) + \frac{L}{2}\|x_a - x_b\|^2. \tag{8}$$

Assumption 2. For each agent i, the local cost function $f_i : \mathbb{R}^n \to \mathbb{R}$ is closed, proper, and strongly convex, i.e., there exists a constant $\mu_i > 0$ such that for all $x_a, x_b \in \mathbb{R}^n$:
$$f_i(x_\alpha) + \nabla f_i(x_\alpha)^\top (x_\beta - x_\alpha) + \frac{\mu_i}{2}\|x_\beta - x_\alpha\|^2 \leq f_i(x_\beta). \tag{9}$$

Assumption 3. For each agent i, the local cost function $f_i : \mathbb{R}^n \to \mathbb{R}$ is closed, proper, twice continuously differentiable, and satisfies the second-order sufficient condition at a local minimizer x^* of (1) without boolean part, i.e., $\nabla^2 f_i(x^*) \succ 0$.

Assumptions 1, 2, and 3 are essential for the convergence analysis of Stage I of Algorithm 1. Assumption 3 provides local convergence guarantees for smooth non-convex problems. Assumptions 1 and 2 provide global convergence theory for smooth convex problems. Detailed convergence analysis can be found in (Du and Wang, 2025, Theorem 2 and 3). Furthermore, Assumption 1 is also used to establish the convergence of Stage II. All these assumptions are common in numerical optimization.

We now establish the convergence theory for Stage II of Algorithm 1. Theorem 1 shows the convergence of the inner level. It relies on the monotonic decrease of the energy function

$$E(z^{[k]}) = \sum_{i=1}^{N} f_i(z^{[k]}) + \alpha\gamma^{[k]}, \tag{10}$$

where $\gamma^{[k]} := (\mathbf{1} - z_d^{[k]})^\top z_d^{[k]}$. The proof is available in Han et al. (2026). It will also appear in an extended version of this paper.

Theorem 1. ***Local Convergence of Stage II of Algorithm 1:*** Suppose each agent i has a local cost function f_i satisfying Assumption 1. For fixed $\rho_2 > L$, the energy function satisfies:

$$E(z^{[k+1]}) < E(z^{[k]}). \tag{11}$$

With the inner-loop convergence of Stage II secured by Theorem 1, the role of the outer loop is to increase α, a strategy that drives the entire algorithm toward a solution satisfying the Boolean constraint. Specifically, this increase is triggered once the change in the inner-loop variable z is below ϵ_{inner}, a criterion which ensures that the inner-loop has sufficiently converged before tightening the constraint. Following (Hall et al., 2021, Theorem 1) and (Ralph* and Wright, 2004, Section 5), it can be shown that as α gradually increases and becomes sufficiently large, the Boolean constraint on the variables will be satisfied and the outer level of Stage II will converge to a local optimum of problem (1).

5. NUMERICAL SIMULATION AND RESULTS

To validate Mix-CALADIN for mixed-integer programs, we adapt the sensor localization problem from Du and Wang (2025) to distributed consensus mixed-integer settings and conduct numerical simulations on two representative cases.

The first case studies a nonconvex sensor localization problem with mixed-Boolean constraints, designed to demonstrate the algorithm's capability in handling challenging nonconvex optimization landscapes. The problem is formally formulated as follows:

$$\begin{aligned}
\min_{\substack{x_i,\, z,\\ \forall i \in \{1,\dots,N\}}} \quad & \sum_{i=1}^{N} \Big(\frac{1}{2}\|y_i - \zeta_i^\alpha\|^2 + \frac{1}{2}\|b_i - \zeta_i^\beta\|^2 \\
& + \frac{1}{2}\sum_{j=1}^{n_c} \big((y_i[j] - b_i[j])^2 - \zeta_i^\gamma[j] \big)^2 \Big) \\
\text{s.t.} \quad & x_i = z, \\
& x_i = [y_i^\top, b_i^\top]^\top, \\
& y_i \in \mathbb{R}^{n_c}, \quad b_i \in \{0,1\}^{n_d}.
\end{aligned} \tag{12}$$

The number of agents is $N = 20$, the dimensions of the continuous and boolean parts of the local variable x_i are $n_c = 10$ and $n_d = 10$, respectively. The parameter vectors $\zeta_i^\alpha \in \mathbb{R}^{n_c}$, $\zeta_i^\beta \in \mathbb{R}^{n_d}$, $\zeta_i^\gamma \in \mathbb{R}^{n_d}$ are drawn randomly from a standard normal distribution. Note that the element-wise operation $(y_i - b_i)^2$ in the objective function requires $n_c = n_d$. Thus, the dimension of ζ_i^γ is n_d (or n_c).

The second case studies a convex objective function, allowing for a performance benchmark against established distributed optimization methods, specifically the projection-based ADMM technique Takapoui et al. (2020). This approach does not rely on mixed-integer programming solvers; instead, it first employs a standard continuous variable solution method and then applies rounding. The parameter set $(N, n_c, n_d, \zeta_i^\alpha, \zeta_i^\beta)$ is shared with the first problem, with the removal of the nonconvex term being the only difference. This problem takes the form:

$$\begin{aligned}
\min_{\substack{x_i,\, z,\\ \forall i \in \{1,\dots,N\}}} \quad & \sum_{i=1}^{N} \Big(\frac{1}{2}\|y_i - \zeta_i^\alpha\|^2 + \frac{1}{2}\|b_i - \zeta_i^\beta\|^2 \Big) \\
\text{s.t.} \quad & x_i = z, \\
& x_i = [y_i^\top, b_i^\top]^\top, \\
& y_i \in \mathbb{R}^{n_c}, \quad b_i \in \{0,1\}^{n_d}.
\end{aligned} \tag{13}$$

For a fair comparison, the parameters of all algorithms are meticulously tuned. The penalty parameter ρ_1 for the augmented Lagrangian function (3), common to both Stage I of Algorithm 1 and projection-based ADMM, is set to 10^5 for nonconvex problem and 10 for convex problem. The penalty parameter ρ_2 for the QP function (7) is set to 10^5 for nonconvex problem and 10 for convex problem.

Fig. 1 illustrates the convergence behavior of the global variable z in Stage I of Algorithm 1 applied to both nonconvex and convex optimization problems. The vertical axis shows the residual norm $\|z^{[k]} - \tilde{z}^*\|^2$ on a logarithmic scale, while the horizontal axis tracks the iteration count up to 200. Both curves exhibit linear convergence rates, which aligns well with the theoretical findings presented in (Du and Wang, 2025, Theorem 2 and 3). It is worth noting that the two curves do not start from the same point since the underlying problems are fundamentally different, with one being convex and the other nonconvex, consequently leading to distinct optimal solutions $\tilde{z}^*$. Stage I facilitates effective convergence across both convex and nonconvex problems of the form (1) by relaxing the mixed-integer variables into a continuous space. This performance aligns with the theoretical expectations established in (Du and Wang, 2025, Theorem 2 and 3), respectively. These results provide a reference point via relaxation, establishing a lower bound for the subsequent Stage II of our algorithm.

Fig. 2 illustrates the convergence behavior of the global variable z in Stage II of Algorithm 1 for nonconvex and convex optimization problems. The results demonstrate that the algorithm effectively converges while the residual is toward zero in both cases, with the convex problem converging after 199 iterations and the nonconvex problem after 237 iterations.

Fig. 3 and Fig. 4 present the energy function (10) profiles throughout the optimization process for the convex and nonconvex problems, respectively. In Fig. 3, the distinc-

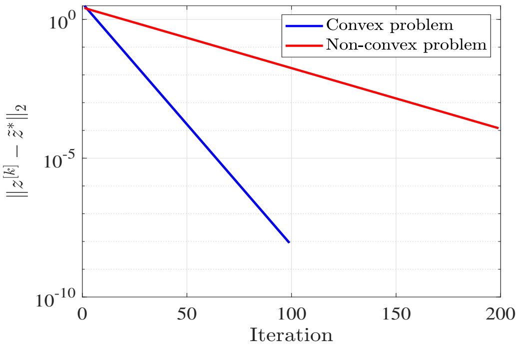

Fig. 1. A comparative analysis of Stage I performance in Algorithm 1 is conducted for both convex and nonconvex problems, with parameter settings $\rho_1 = 10$ for convex cases (13) and $\rho_1 = 10^5$ for nonconvex cases (12).

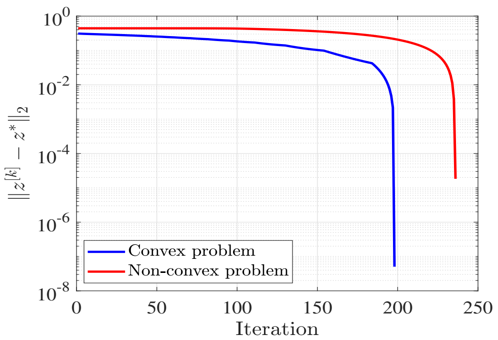

Fig. 2. A comparative analysis of Stage II performance in Algorithm 1 is conducted for both convex and nonconvex problems, with parameter settings $\rho_2 = 10$ for the convex case (13) and $\rho_2 = 10^5$ for the nonconvex cases (12).

tive staircases pattern is observed in the convex case, characterized by 20 abrupt increases followed by gradual declines. This behavior is attributed to the outer-loop updates of the parameter α: each increase in α temporarily elevates the energy function value, after which the inner-loop optimization reduces it toward optimality. This process terminates in satisfaction of the Boolean constraint. In Fig. 4, the nonconvex problem shows more frequent rises in the energy function value within the first 102 iterations, indicating frequent exits from the inner-loop due to insufficient α values. Once α becomes sufficiently large after 102 iterations, the energy function decreases monotonically, converging near its initial value by iteration 237 and satisfying the Boolean constraint.

Figure 5 shows the comparative performance between the projection-based ADMM Takapoui et al. (2020) and Algorithm 1 in a convex optimization problem. It is important to note that ADMM is not generally applicable to problems containing nonconvex terms. Furthermore, the projection-based method serves as a heuristic baseline and lacks convergence theory guarantees. As shown by the blue curve, the projection-based ADMM method converges to a local minimum. In contrast, Algorithm 1 (red curve) proceeds in two distinct stages: in Stage I, it rapidly converges to the relaxed solution of the problem with continuous relaxation for the Boolean constraint, providing a high-quality initial point for Stage II. The introduction of exact Boolean constraint in Stage II leads to a temporary increase in the objective value, reflecting the formulation's shift towards feasibility. After 199 iterations, Algorithm 1 converges to a solution that is superior to that attained by the projection-based ADMM approach, demonstrating both improved performance and more rigorous convergence behavior with theoretical guarantee.

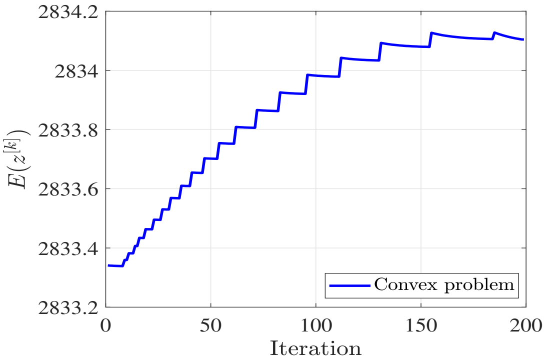

Fig. 3. Evolution of the energy function $E(z^{[k]})$ (see (10)) during Stage II of Algorithm 1 for the convex problem (13).

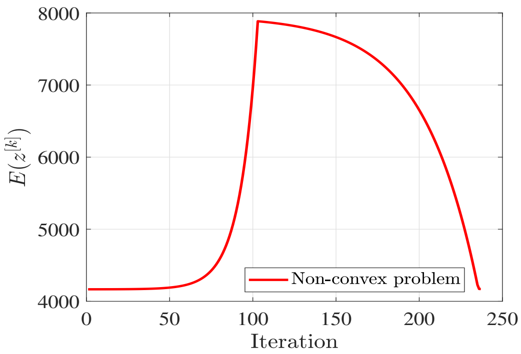

Fig. 4. Evolution of the energy function $E(z^{[k]})$ during Stage II of Algorithm 1 for the nonconvex problem (12).

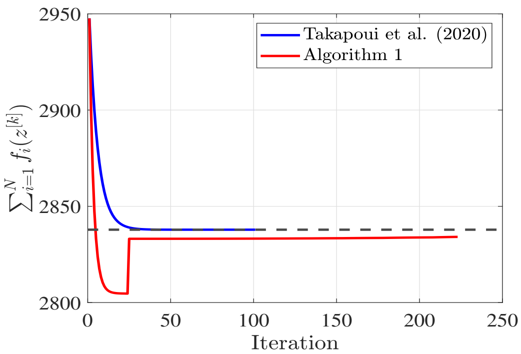

Fig. 5. Convergence comparison of Algorithm 1 and Takapoui et al. (2020).

6. CONCLUSION

In this paper, we presented a novel distributed algorithm for consensus mixed-integer optimization named Mix-CALADIN. Our algorithm efficiently handles both convex and nonconvex optimization problems while eliminating the dependency on conventional mixed-integer solvers. Numerical simulations validate the algorithm's performance and demonstrate its advantages over existing approaches in the literature.

REFERENCES

Camisa, A., Notarnicola, I., and Notarstefano, G. (2021). Distributed primal decomposition for large-scale MILPs. *IEEE Transactions on Automatic Control*, 67(1), 413–420.

Conforti, M., Cornuéjols, G., and Zambelli, G. (2014). *Integer Programming*, volume 271. Springer.

Dong, S. and Xia, W. (2024). A distributed algorithm for large-scale multi-agent MINLPs. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, 3266–3271. IEEE.

Doostmohammadian, M., Aghasi, A., Pirani, M., Nekouei, E., Zarrabi, H., Keypour, R., Rikos, A.I., and Johansson, K.H. (2025). Survey of distributed algorithms for resource allocation over multi-agent systems. *Annual Reviews in Control*, 59, 100983.

Du, X., Johansson, K.H., and Rikos, A.I. (2025a). Decentralized optimization via RC-ALADIN with efficient quantized communication. In *2025 IEEE 64th Conference on Decision and Control (CDC)*, 4357–4363. IEEE.

Du, X. and Wang, J. (2023). Consensus ALADIN: A framework for distributed optimization and its application in federated learning. *arXiv preprint arXiv:2306.05662*.

Du, X. and Wang, J. (2025). Distributed consensus optimization with consensus ALADIN. In *2025 American Control Conference (ACC)*, 2949–2955.

Du, X., Zhou, X., Zhu, S., and Rikos, A.I. (2025b). Convergence theory of flexible ALADIN for distributed optimization. In *2025 European Control Conference (ECC)*, 1907–1912.

Du, X., Zhu, S., Wang, Y., Han, B., and He, X. (2023). Optimal demand shut-offs of ac microgrid using AO-SBQP method. *IFAC-PapersOnLine*, 56(2), 1431–1436.

Falsone, A., Margellos, K., and Prandini, M. (2019). A decentralized approach to multi-agent MILPs: finite-time feasibility and performance guarantees. *Automatica*, 103, 141–150.

Hall, J., Nurkanović, A., Messerer, F., and Diehl, M. (2021). A sequential convex programming approach to solving quadratic programs and optimal control problems with linear complementarity constraints. *IEEE Control Systems Letters*.

Han, B., Du, X., Johansson, K.H., and Rikos, A.I. (2026). Mix-CALADIN: A distributed algorithm for consensus mixed-integer optimization. *arXiv preprint arXiv:2604.14897*.

Houska, B., Frasch, J., and Diehl, M. (2016). An augmented lagrangian based algorithm for distributed nonconvex optimization. *SIAM Journal on Optimization*, 26(2), 1101–1127.

Kuwata, Y. and How, J.P. (2010). Cooperative distributed robust trajectory optimization using receding horizon MILP. *IEEE Transactions on Control Systems Technology*, 19(2), 423–431.

Le, V.A. and Malikopoulos, A.A. (2025). Distributed mixed-integer quadratic programming for mixed-traffic intersection control. *arXiv preprint arXiv:2504.04618*.

Liu, Z. and Stursberg, O. (2022). Distributed solution of mixed-integer programs by ADMM with closed duality gap. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, 279–286. IEEE.

Luo, Z.Q., Ma, W.K., So, A.M.C., Ye, Y., and Zhang, S. (2010). Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27(3), 20–34.

Murray, A., Faulwasser, T., Hagenmeyer, V., Villanueva, M.E., and Houska, B. (2021). Partially distributed outer approximation. *Journal of Global Optimization*, 80(3), 523–550.

Murray, A. and Hagenmeyer, V. (2020). Convergence of mixed-integer ALADIN. In *Proceedings of the 4th International Conference on Algorithms, Computing and Systems*, 51–54.

Pia, A.D. (2023). An approximation algorithm for indefinite mixed integer quadratic programming. *Mathematical Programming*, 201(1), 263–293.

Ralph*, D. and Wright, S.J. (2004). Some properties of regularization and penalization schemes for MPECs. *Optimization Methods and Software*, 19(5), 527–556.

Sun, K., Sun, M., and Yin, W. (2024). Decomposition methods for global solution of mixed-integer linear programs. *SIAM Journal on Optimization*, 34(2), 1206–1235.

Takapoui, R., Moehle, N., Boyd, S., and Bemporad, A. (2020). A simple effective heuristic for embedded mixed-integer quadratic programming. *International journal of control*, 93(1), 2–12.

Testa, A., Rucco, A., and Notarstefano, G. (2019). Distributed mixed-integer linear programming via cut generation and constraint exchange. *IEEE Transactions on Automatic Control*, 65(4), 1456–1467.

Vujanic, R., Esfahani, P.M., Goulart, P.J., Mariéthoz, S., and Morari, M. (2016). A decomposition method for large scale MILPs, with performance guarantees and a power system application. *Automatica*, 67, 144–156.

Wolsey, L.A. (2020). *Integer programming*. John Wiley & Sons.

Wu, S., Wang, Y., Wang, J., Rikos, A.I., and Du, X. (2025). A time splitting based optimization method for nonlinear MHE. In *2025 IEEE 64th Conference on Decision and Control (CDC)*, 5335–5341. IEEE.

Zhu, R. and You, K. (2025). Distributed mixed-integer ADMM for express transport network design. *IFAC-PapersOnLine*, 59(4), 210–215.

Zhu, S. and Du, X. (2022). Alternating direction based sequential boolean quadratic programming method for transmit antenna selection. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, 7491–7496. IEEE.