ALADIN-β: A Distributed Optimization Algorithm for Solving MPCC Problems

Yifei Wang, Shuting Wu, Genke Yang, Jian Chu, Apostolos I. Rikos and Xu Du*

Abstract—**Mathematical Programs with Complementarity Constraints (MPCC) are critical in various real-world applications but notoriously challenging due to non-smoothness and degeneracy from complementarity constraints. The ℓ_1-Exact Penalty-Barrier enhanced `IPOPT` improves performance and robustness by introducing additional inequality constraints and decision variables. However, this comes at the cost of increased computational complexity due to the higher dimensionality and additional constraints introduced by the centralized formulation. To mitigate this, we propose a distributed structure-splitting reformulation that decomposes these inequality constraints and auxiliary variables into independent sub-problems. Furthermore, we introduce Augmented Lagrangian Alternating Direction Inexact Newton (ALADIN)-β, a novel approach that integrates the ℓ_1-Exact Penalty-Barrier method with ALADIN to efficiently solve the distributed reformulation. Numerical experiments demonstrate that even without a globalization strategy, the proposed distributed approach achieves the fast convergence while maintaining high precision.**

I. INTRODUCTION

Mathematical Programs with Complementarity Constraints (MPCC) [1] are a class of optimization problems incorporating complementarity constraints. They have become a valuable modeling framework in control theory, particularly for optimization-based control problems involving non-smooth dynamical systems [2] and hybrid systems [3]. Key challenges in optimal control, such as trajectory optimization for contact-aware robotic manipulators [4], logic-based control in chemical processes [5], and mixed Boolean optimization for smart grids [6], can be systematically reformulated as MPCC through a unified framework [7], thereby leveraging complementarity structures to encode non-smooth physical and logical constraints. To address degeneracies in such problems, several solution methods have been proposed [8]. Notably, a method was introduced that bypasses direct solution of the original problem by utilizing sequential convex

programming to solve penalty sub-problems derived from linear complementarity quadratic programs, with penalty parameters adaptively adjusted through homotopy [9], [10]. Building on these foundations, a subsequent study extends the original method to tackle a general mixed-integer non-linear problem arising from antenna selection optimization in a sequential quadratic programming (SQP) fashion [11]. Additionally, [12] applied an ℓ_1-exact penalty strategy to handle complementarity constraints, coupled with the penalty parameter update strategy to address degenerate problems with `IPOPT`. This integration demonstrates improved speed and robustness on standard MPCC benchmarks by introducing additional inequality constraints and decision variables. However, as a centralized approach, this ℓ_1-strategy inherently struggles with increased computational complexity due to higher dimensionality and additional constraints. As problem sizes grow, memory constraints and processing bottlenecks further limit its scalability. This motivates the exploration of distributed optimization (DO) for efficient MPCC solving.

DO algorithms have become essential for tackling large-scale engineering problems [13]–[16] by mitigating the *curse of dimensionality* through decomposition into parallelizable sub-problems, improving scalability and reducing communication overhead. They typically follow two main approaches: primal and dual decomposition [17]. Dual decomposition methods, particularly Alternating Direction Method of Multipliers (ADMM) [16] and Augmented Lagrangian Alternating Direction Inexact Newton (ALADIN) [18]–[21], offer rigorous theoretical convergence guarantees and exhibit practical effectiveness in DO. Theoretically, ALADIN synthesizes the distributed coordination efficiency of ADMM with the high precision of SQP, guaranteeing global convergence for convex problems [22] and local convergence for non-convex problems [18]. Recent advancements like ALADIN-Prox [23], integrate proximal regularization with ALADIN, extending this to global convergence in non-convex settings with preserved complexity. In practice, ALADIN exhibits exceptional performance in large-scale engineering applications, primarily categorized into three types: a) *spatial-splitting-based* optimization problems (e.g., optimal power flow) [14], [15], [24], b) *time-splitting-based* optimization problems (e.g., model predictive control (MPC) and moving horizon estimation (MHE)) [25]–[28], and c) the distributed reconstruction of non-summation centralized optimization [29], also known as *structure-splitting-based* optimization. The remarkable numerical performance of ALADIN relies on the fulfillment of regularity conditions such as the Linear

* Corresponding author.

Yifei Wang, Genke Yang and Jian Chu are with the Ningbo Artificial Intelligence Institute and the Department of Automation, Shanghai Jiao Tong University, Ningbo 315000, China, and also with the School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, Shanghai 200240, China. E-mails: {`yifeiw4ng`, `gkyang`, `chujian`}`@sjtu.edu.cn`.

Shuting Wu is with School of Mathematics and Statistics, North China University of Water Resources and Electric Power, Zhengzhou, China, and is also with the HNAS Institute of Mathematics, Henan Academy of Science, Zhengzhou, China. E-mail: `wushuting0126@163.com`.

Apostolos I. Rikos and Xu Du are with the Artificial Intelligence Thrust of the Information Hub, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China. Apostolos I. Rikos is also affiliated with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong, China. E-mails: {`apostolosr`, `michaelxudu`}`@hkust-gz.edu.cn`.

Independence Constraint Qualification (LICQ) and Second-Order Sufficiency Conditions (SOSC), yet these conditions are inherently violated in degenerate problems like MPCC. Specifically, MPCC often violate standard constraint qualifications (CQ) at every feasible point, due to the non-convexity and non-smoothness introduced by the complementarity constraints. To the best of our knowledge, existing research on ALADIN has not addressed MPCC. This work proposes an enhanced ALADIN framework tailored for MPCCs, aiming to bridge this methodological gap.

Building on the ℓ_1-penalty method [12] and the structure-splitting-based optimization of ALADIN [29], we propose a novel DO algorithm, ALADIN-β, for solving MPCC problems. First, following the approach in [12], the ℓ_1-exact penalty method is applied to handle the non-smooth and non-convex complementarity constraints, and then a universally applicable structure-splitting paradigm is proposed for the resulting general nonlinear programming (NLP) problems, see Section III. Next, we propose ALADIN-β for solving the distributed problems derived from the structure-splitting process, see Section IV. Finally, representative numerical examples are compared with existing state-of-the-art solvers to validate the effectiveness and efficiency of ALADIN-β, see Section V.

II. PRELIMINARIES

This section begins with a review of NLP with orthogonality constraints. We then revisit the foundation of the ℓ_1-exact penalty-barrier-based interior-point method, an efficient centralized approach for solving MPCC. Finally, we provide an overview of the ALADIN algorithm, a powerful algorithm designed for large-scale distributed optimization.

A. Problem Formulation

An NLP with orthogonality constraints can be formulated as follows,

$$\min_{x \in \mathbb{R}^n} \quad f(x) \tag{1a}$$

$$\text{s.t.} \quad g(x) = 0, \tag{1b}$$

$$x \leq 0. \tag{1c}$$

Here, $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}$ are all assumed to be twice continuously differentiable. Moreover, equation (1b) defines the orthogonality constraints of the form $g(x) = G(x)^\top H(x)$, where $G : \mathbb{R}^n \to \mathbb{R}^{n_g}$, $H : \mathbb{R}^n \to \mathbb{R}^{n_g}$ are twice continuously differentiable. For simplicity, this work focuses on linear inequality constraints. However, leveraging the ALADIN framework [18], future extensions will address more general inequality constraints.

Remark 1 *MPCC are considered as a class of degenerate NLPs, where redundant constraints violate CQs[1]. Typically,*

the complementarity constraints in MPCC, formulated as $0 \leq G(x) \perp H(x) \geq 0$, can be equivalently expressed via smooth constraints: $G(x)^\top H(x) = 0$, $G(x) \geq 0$ and $H(x) \geq 0$. Active set methods resolve small-scale degeneracies effectively, yet their combinatorial complexity escalates with dimensionality. Conversely, interior-point methods like `IPOPT` [31] regularize constraints globally under localized degeneracy assumptions but struggle with numerical instability due to inconsistent constraint linearizations.

B. The ℓ_1-Exact Penalty-Barrier-Based Reformulation

To tackle the aforementioned challenges in solving problem (1), we review an improved interior-point method in this section. This method systematically mitigates degeneracy-induced issues while preserving numerical robustness across problem scales.

By incorporating the inequality constraint (1c) into the objective function using barrier terms, we formulate a sequence of barrier problems whose solutions asymptotically converge to those of (1) as the barrier parameter μ approaches zero[2]. To further penalize the equality constraint (1b), we yield the combined non-smooth ℓ_1-penalty term[3] with a parameter ρ,

$$\min_{x \in \mathbb{R}^n} \quad f(x) - \mu \sum_{i=1}^{n} \ln(-x_i) + \rho \|g(x)\|_1, \tag{2}$$

where $\mu \in \mathbb{R}_{>0}$ is the barrier parameter and $\rho \in \mathbb{R}_{>0}$ is the penalty parameter. However, the non-smoothness induced by the ℓ_1-penalty term complicates the minimization process, necessitating specialized non-smooth optimization strategies.

To address this, [12] introduces auxiliary variables p and n to augment the equality constraints, resulting in a smooth reformulation of problem (2):

$$\min_{x \in \mathbb{R}^n; p,n \in \mathbb{R}^m} \quad f(x) - \mu \sum_{i=1}^{n} \ln(-x_i) + \rho(p+n)^\top e \tag{3}$$

$$\text{s.t.} \quad g(x) - p + n = 0, \quad p \geq 0, \quad n \geq 0,$$

where $e = [1, 1, \cdots, 1]^\top \in \mathbb{R}^m$ denotes a vector of appropriate dimension filled with ones. Crucially, the reformulated equality constraints maintain full-rank properties, ensuring the LICQ holds even at rank-deficient points of the barrier problem. This enables robust computation of steps for degenerate problems. Replacing the proposed ℓ_1-exact penalty-barrier phase in the `IPOPT` solver has been shown to significantly improve success rates for degenerate problems and accurately solve a practical MPCC problem in a single optimization stage [12]. This approach avoids numerical challenges associated with non-smooth terms but comes at the cost of increased problem dimensions and the need for an additional penalty parameter update strategy.

---

[1]Common CQs include LICQ and the weaker Mangasarian-Fromovitz Constraint Qualification (MFCQ) [30]. While LICQ (sufficient but not necessary) ensures the existence of bounded and unique Lagrange multipliers by requiring linearly independent active constraint gradients, the weaker MFCQ guarantees only the boundedness of the multipliers. Despite being less stringent, MFCQ plays a crucial role in the convergence of most NLP solvers due to its wider applicability in degenerate problems.

[2]Under the MFCQ, the sequence of local minimizers for the barrier-augmented function asymptotically converges to the optimal solution $x^\star$ of (1) as $\mu \to 0$.

[3]To relax the restrictive LICQ, we reformulate problem (1) into an exact ℓ_1-penalty-barrier framework, where the equivalence to the original problem is preserved if $\rho \geq \rho^\star := \|(\lambda^\star, z^\star)\|_\infty$ with $\lambda^\star, z^\star$ denoting optimal dual variables [30].

C. Fundamentals of ALADIN for QP

In this section, we review how ALADIN, a DO algorithm, handles large-scale QPs with inequality constraints [29]. Before presenting ALADIN, we define the local update notation: $(\cdot)^+$ denotes the value after the update, while $(\cdot)^-$ represents the value before the update.

Considering a strictly convex quadratic programming problem

$$\min_{x \in \mathbb{R}^n} \quad \tilde{f}(x) \atop \text{s.t.} \quad \underline{x} \le x \le \overline{x}, \tag{4}$$

with $\tilde{f}(x) = \frac{1}{2}x^\top Q x$, where $Q \succ 0$, $\underline{x}$ and $\overline{x}$ denote the lower and upper bound of x, respectively. By introducing an auxiliary variable z, the inequality constraints of problem (4) can be incorporated into the objective function in a reformulated manner,

$$\min_{x,z} \quad \tilde{f}(x) + \tilde{g}(z) \quad \text{s.t.} \quad x = z \mid \lambda, \tag{5}$$

where the function

$$\tilde{g}(z) \overset{\text{def}}{=} \begin{cases} 0 & \text{if} \quad \underline{z} \le z \le \overline{z} \\ \infty & \text{otherwise} \end{cases} \tag{6}$$

is defined as the indicator function, with $\underline{z} = \underline{x}$ and $\overline{z} = \overline{x}$, and λ denotes the dual variable with respect to the coupling constraint. The customized ALADIN [29] for solving problem (4) initializes primal-dual variables (x, z, λ) and selects positive-definite matrices $\Sigma \succ 0$ and $K \succ 0$, proceeding as follows,

$$\begin{cases} v^+ = \arg\min_v \tilde{f}(v) + \lambda^\top v + \frac{1}{2}\|v - x\|_\Sigma^2; \\ w^+ = \arg\min_w \tilde{g}(w) - \lambda^\top w + \frac{1}{2}\|w - z\|_K^2; \\ (x^+, z^+, \lambda^+) \\ = \begin{cases} \min_{x^+, z^+} & \frac{1}{2}\left\| \begin{bmatrix} x^+ - v^+ \\ z^+ - w^+ \end{bmatrix} \right\|_{H,K}^2 + \begin{bmatrix} \nabla \tilde{f}(v^+) \\ \partial \tilde{g}(w^+) \end{bmatrix}^\top \begin{bmatrix} x^+ \\ z^+ \end{bmatrix} \\ \text{s.t.} & x^+ = z^+ \mid \lambda^+. \end{cases} \end{cases} \tag{7}$$

Here, $\nabla \tilde{f}$ and $\partial \tilde{g}$ denote the gradient and subgradient of $\tilde{f}$ and $\tilde{g}$, respectively, while the matrices $H \succ 0$ and $K \succ 0$ are local Hessian approximations of $\tilde{f}$ and $\tilde{g}$. Note that, ALADIN alternates between parallel sub-problems solving at sub-nodes to update v^+ and w^+, and coordination at the central node to determine (x^+, z^+, λ^+), iterating until convergence. This study demonstrates the significant potential of ALADIN in solving large-scale problems, focusing on solving convex QPs [29, Section IV-B]. Its extension to non-convex optimization problems will be presented in subsequent sections.

III. STRUCTURE-SPLITTING-BASED REFORMULATION

Building on Sections II-B and II-C, we introduce a structure-splitting reformulation of problem (3) by incorporating auxiliary variables. The optimization variables are grouped into independent blocks, resulting in a generalized representation of the structure-splitting formulation as follows.

$$\min_{\alpha,\beta,\gamma} \quad \phi(\alpha,\gamma) + \varphi(\beta) + \psi(\gamma) \tag{8a}$$
$$\text{s.t.} \quad g(x) - q + m = 0, \tag{8b}$$
$$x = \beta, \quad m = n, \quad p = q. \tag{8c}$$

The composite variables are structured as $\alpha = [x^\top, q^\top, m^\top]^\top$ and $\gamma = [p^\top, n^\top]^\top$, where q and m act as local replicas of the auxiliary variables p and n, respectively, to enforce consensus among distributed sub-problems. Furthermore, the auxiliary variable β represents the local variables associated with the log-barrier term for the inequality constraints in (3). To enhance numerical stability, a relaxation parameter r is integrated into the formulation, strategically designed to mitigate abrupt gradient-driven oscillations near constraint boundaries. Furthermore, the objective function (8a) is decomposed into three distinct components, expressed as follows,

$$\phi(\alpha,\gamma) \overset{\text{def}}{=} f(x) + \frac{1}{2}\|q - p\|_P^2 + \frac{1}{2}\|m - n\|_M^2, \tag{9a}$$

$$\varphi(\beta) \overset{\text{def}}{=} -\sum_{i=1}^n \mu_i \ln(r - \beta_i), \tag{9b}$$

$$\psi(\gamma) \overset{\text{def}}{=} \rho(p + n)^\top e - \sum_{i=1}^{n_g} \mu_i (\ln(r+p_i) + \ln(r+n_i)), \tag{9c}$$

with $r > 0$, $\mu_i > 0$, $P \succ 0$, $M \succ 0$. Sub-problem (9a) is augmented by two proximal terms in q and m to regularize the update step sizes near optimality. Inspired by [29], the inequality constraints in (3) are reformulated as *relaxed log-barrier* objectives, eliminating the need for explicit feasibility detection. Refer to [29] for details.

By introducing the coupling matrices

$$A_1 \overset{\text{def}}{=} \begin{bmatrix} I_{|x|} & \mathbf{0}_{|x| \times n_g} & \mathbf{0}_{|x| \times n_g} \\ \mathbf{0}_{n_g \times |x|} & I_{n_g} & \mathbf{0}_{n_g \times n_g} \\ \mathbf{0}_{n_g \times |x|} & \mathbf{0}_{n_g \times n_g} & I_{n_g} \end{bmatrix},$$
$$A_2 \overset{\text{def}}{=} \begin{bmatrix} -I_{|x|} \\ \mathbf{0}_{n_g \times |x|} \\ \mathbf{0}_{n_g \times |x|} \end{bmatrix}, \quad A_3 \overset{\text{def}}{=} \begin{bmatrix} \mathbf{0}_{|x| \times n_g} & \mathbf{0}_{|x| \times n_g} \\ -I_{n_g} & \mathbf{0}_{n_g \times n_g} \\ \mathbf{0}_{n_g \times n_g} & -I_{n_g} \end{bmatrix}, \tag{10}$$

Problem (8) can then be represented as,

$$\min_{\alpha,\beta,\gamma} \quad \phi(\alpha,\gamma) + \varphi(\beta) + \psi(\gamma) \atop \text{s.t.} \quad G(\alpha) = 0 \qquad\qquad |\kappa, \atop A_1\alpha + A_2\beta + A_3\gamma = 0 \quad |\lambda, \tag{11}$$

where $G(\alpha) = g(x) - q + m$, κ represents the dual variable of the first constraint and λ denotes the dual variable of the coupling constraint.

Remark 2 *Drawing inspiration from (5), we employ an analogous technique to reformulate the inequality-constrained problem* (3) *into a distributed reformulation* (11) *by incorporating inequality constraints into the objective function. This reformulation strategically eliminates the computational overhead associated with infeasibility detection in constrained optimization. Note that, although the formulation*

of problem (8) *introduces new variables and a modified objective function, its summation structure enables distributed solving.*

IV. ALADIN-β

In order to solve the reformulated distributed ℓ_1-penalty-barrier problem (11), we develop an efficient solution approach leveraging the ALADIN algorithm and detail its implementation strategy.

A. Algorithm Development

We propose a novel algorithm, ALADIN-β, to solve (11). The algorithm begins with an initial guess for all primal and dual variables $(\alpha, \beta, \gamma, \lambda)$ in (11). It then iterates through the following steps until the termination criteria are met. The main steps of ALADIN-β are outlined below.

1) Choose proper initial values for the barrier and penalty parameter $\mu, \rho \geq 0$ and positive semi-definite scaling matrices $\Sigma_i \in \mathbb{S}_+^{n_x}$ for all $i \in \{1, 2, 3\}$.

2) Solve the decoupled optimization problems in parallel,

- For Sub-problem 1:

$$\hat{\alpha} = \arg\min_{\alpha} \ \phi(\alpha, \gamma^-) + \lambda^\top A_1 \alpha + \frac{1}{2}\|\alpha - \alpha^-\|_{\Sigma_1}^2 \quad (12)$$
$$\text{s.t. } G(\alpha) = 0 \mid \kappa.$$

- For Sub-problem 2:

$$\hat{\beta} = \arg\min_{\beta} \ \varphi(\beta) + \lambda^\top A_2 \beta + \frac{1}{2}\|\beta - \beta^-\|_{\Sigma_2}^2. \quad (13)$$

- For Sub-problem 3:

$$\hat{\gamma} = \arg\min_{\gamma} \ \psi(\gamma) + \lambda^\top A_3 \gamma + \frac{1}{2}\|\gamma - \gamma^-\|_{\Sigma_3}^2. \quad (14)$$

3) Evaluate the sensitivity information for each sub-problem, obtaining the gradient g_i and Hessian H_i for all $i \in \{1, 2, 3\}$ based on their respective local optimal solutions, expressed as follows,

$$\nabla_\alpha \phi(\hat{\alpha}) = \Sigma_1(\alpha^- - \hat{\alpha}) - A_1^\top \lambda - \nabla_\alpha G(\hat{\alpha})^\top \kappa, \quad (15a)$$
$$\nabla_\beta \varphi(\hat{\beta}) = \Sigma_2(\beta^- - \hat{\beta}) - A_2^\top \lambda, \quad (15b)$$
$$\nabla_\gamma \psi(\hat{\gamma}) = \Sigma_3(\gamma^- - \hat{\gamma}) - A_3^\top \lambda. \quad (15c)$$

Then choose symmetric Hessian approximations

$$H_1 \approx \nabla_\alpha^2 \{\phi(\hat{\alpha}, \gamma^-) + \kappa^\top G(\hat{\alpha})\} \succ 0, \quad (16a)$$

$$H_2 \approx \nabla_\beta^2 \{\varphi(\hat{\beta})\} = \text{diag}\left(\frac{\mu_i}{(r - \hat{\beta}_i)^2}\right) \succ 0, \quad (16b)$$

$$H_3 \approx \nabla_\gamma^2 \{\psi(\hat{\gamma})\} = \text{diag}\left(\frac{\mu_i}{(r - \hat{p}_i)^2} + \frac{\mu_i}{(r - \hat{n}_i)^2}\right) \succ 0. \quad (16c)$$

4) Aggregate the sensitivity information from all sub-problems to construct and solve the consensus QP problem subject to equality constraints,[4]

---

[4] For QP problems with equality constraints, we can derive closed-form solutions for updating the primal and dual variables using the first-order KKT conditions. This allows the primal updates to be distributed to the sub-nodes for internal updating, thereby further enhancing computational efficiency for large-scale problems.

$$\min_{\alpha^+, \beta^+, \gamma^+} \frac{1}{2} \left\| \begin{bmatrix} \alpha^+ - \hat{\alpha} \\ \beta^+ - \hat{\beta} \\ \gamma^+ - \hat{\gamma} \end{bmatrix} \right\|_H^2 + \begin{bmatrix} \nabla_\alpha \phi(\hat{\alpha}) \\ \nabla_\beta \varphi(\hat{\beta}) \\ \nabla_\gamma \psi(\hat{\gamma}) \end{bmatrix}^\top \begin{bmatrix} \alpha^+ - \hat{\alpha} \\ \beta^+ - \hat{\beta} \\ \gamma^+ - \hat{\gamma} \end{bmatrix}$$
$$\text{s.t.} \quad C (\alpha^+ - \hat{\alpha}) = 0,$$
$$A_1 \alpha^+ + A_2 \beta^+ + A_3 \gamma^+ = 0 \mid \lambda^{\text{QP}}. \quad (17)$$

Here the matrix H is a block diagonal matrix composed of $H_i, \forall i \in \{1, 2, 3\}$ from (16). Moreover, $C = \nabla_\alpha G(\alpha)^\top = \left(\frac{\partial g(x)}{\partial x}, -I_{n_g}, I_{n_g}\right)$ denotes the Jacobian matrix of the local equality constraints.

5) Update the primal variables $\alpha^+ \to \alpha^-, \beta^+ \to \beta^-, \gamma^+ \to \gamma^-$, as well as $\lambda = \lambda + \theta(\lambda^{\text{QP}} - \lambda)$ with $\theta = 1$ for a full-step update[5]. Verify whether the stopping conditions of the algorithm are satisfied. If not, update the barrier parameter μ_j, ρ and go to Step 2); otherwise, terminate the iteration.

Remark 3 *The proposed algorithm ALADIN-β systematically decomposes the original problem into independent sub-problems, enabling parallel computation. Sub-problem* (12) *utilizes the values of p and n from the previous iteration to solve the primary objective and equality constraints. As detailed in Section II-B, its constraints are reformulated to eliminate dependence on the LICQ. This property allows ALADIN-β to avoid various numerical issues that arise when solving systems with degeneracies. The subsequent two sub-problems (see* (13), (14)) *are introduced to iteratively refine the results from Sub-problem 1. Specifically, Sub-problem* (13) *enforces inequality constraints through log-barrier penalty terms, while Sub-problem* (14) *adaptively adjusts p and n to tighten equality constraint satisfaction. These auxiliary sub-problems, designed with inequality-constrained log-barrier formulations, exhibit simplified structures ensuring analytical closed-form solutions and guaranteed positive definiteness of their Hessian matrices. Moreover, incorporating proximal terms for q and m in the objective function of Sub-problem* (9a) *not only improves its convexity but also enhances the stability of algorithm convergence across iterations.*

Remark 4 *Based on the reformulation in Section II-B, ALADIN-β extends standard ALADIN by relaxing locally LICQ and SOSC requirements [18]–[20], improving robustness to degeneracies and enabling application to MPCCs. It retains ALADIN's distributed structure by decomposing the problem into parallelizable sub-problems and solving a simplified high-dimensional QP. In strongly coupled yet structurally simple problems, ALADIN-β outperforms the ℓ_1-exact penalty-barrier phase of interior-point methods in distributed settings. Our smoothing-based MPCC reformulation [12] broadens ALADIN's applicability while preserving its local convergence properties. A full convergence analysis will be provided in an extended version of this work.*

---

[5] For simplicity, only the full-step update is presented here. If the original problem is highly nonlinear or the initial point is far from the optimal solution, techniques such as line search or other globalization strategies can be employed to improve convergence performance, as detailed in [18], [23].

V. Numerical Experiment

This section presents simulation results demonstrating the advantages of ALADIN-β for MPCC problems and compares it with two variants of the ℓ_1-penalized `IPOPT` solver [12], referred to as `IPOPT1` and `IPOPT2`[6], as well as a baseline `IPOPT-vanilla` that directly solves the original MPCC problem (1) without reformulation. Notably, we emphasize the QP step, as ALADIN tolerates inexact subproblem solutions without compromising convergence, leading to reduced computational effort supported by theoretical analysis [20] and numerical evidence [28]. In `IPOPT1`, the penalty parameter μ is updated after a single QP step, while in `IPOPT2`, it is updated only after fully solving the current barrier problem. The implementation of ALADIN-β is based on `Casadi-v3.6.7` with `IPOPT` and `MATLAB 2024b`. Moreover, to eliminate the influence of heuristic tuning, all the implemented algorithms update the parameters at a fixed rate, where the barrier parameter $\mu = 10$ (see (9)) decreases by a factor of 0.2, and the penalty parameter $\rho = 10$ increases by a factor of 4 at each update.

To clearly demonstrate the numerical performance of ALADIN-β, we adopt a canonical MPCC example (see page 21 of `https://www.syscop.de/files/2023ss/nonsmooth_school/kirches_MPECs_2.pdf`), presented as follows:

$$\min_{\hat{x}, \tilde{x} \in \mathbb{R}^n} \quad \frac{1}{2}\|\hat{x} - e\|_2^2 + \frac{1}{2}\|\tilde{x} - e\|_2^2$$
$$\text{s.t.} \quad \hat{x}^\top \tilde{x} = 0, \quad (18)$$
$$x \succeq 0.$$

Here, $\hat{x}$ and $\tilde{x}$ are n-dimensional column vectors, with $x = [\hat{x}^\top, \tilde{x}^\top]^\top$. Despite the convexity of the objective function and inequality constraints, the equality constraints introduce non-convexity and non-smoothness, violating CQs and posing challenges for standard numerical algorithms.

Figure 1 illustrates the proposed ALADIN-β algorithm solving problem (18), where $\hat{x}, \tilde{x} \in \mathbb{R}_+$. The problem features two local minima at $(0,1)$ and $(1,0)$, along with a relatively weak local maximum at the origin.

Here, we fix the total number of ALADIN-β iterations to 100 and set the initial point to $(1,1)$. The trajectory first moves toward the origin to satisfy the complementarity constraints and, once these are nearly met, pivots toward objective minimization, ultimately converging to the local optimum $(1,0)$ of (18).

Figure 2 presents the convergence behavior of `IPOPT1`, `IPOPT2`, `IPOPT-vanilla`, and ALADIN-β for solving the MPCC problem (18) with $\hat{x}, \tilde{x} \in \mathbb{R}^{10}$. The upper subplot shows the convergence of the primal variable x. `IPOPT2`
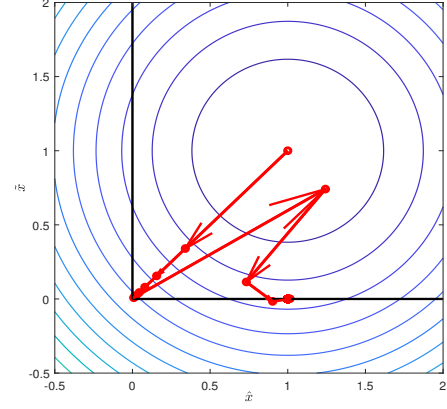

Fig. 1: Iteration process of ALADIN-β on the two-dimensional instance of problem (18).

exhibits slow convergence in the first 90 iterations, followed by linear convergence to a precision of 10^{-8} after approximately 180 iterations. In contrast, ALADIN-vanilla and ALADIN-β both exhibit quadratic convergence, attaining 10^{-9} accuracy in 42 iterations and 10^{-10} accuracy within the fewest iterations, respectively. The lower subplot tracks the complementarity constraint residuals of (18). ALADIN-β satisfies these constraints to 10^{-16} precision within approximately 20 iterations, equiring the fewest iterations among all methods. Notably, `IPOPT`-based methods incorporate adaptive globalization strategies (such as the filter method) to maintain convergence beyond 10^{-16} precision, achieving higher accuracy at a consistent convergence rate compared to ALADIN-β. Despite both `IPOPT1` and ALADIN-β updating parameters after each QP step, the convergence curves of `IPOPT1` in Figure 2 show slow convergence.

In DO, the way local information is exchanged and utilized typically limits the ability of the algorithm to achieve higher convergence precision. However, as illustrated in Figure 2, the proposed ALADIN-β, by appropriately utilizing the first- and second-order information as equation (15) and (16), converges to an acceptable accuracy of 10^{-10} in the fewest iterations, outperforming all the `IPOPT`-based methods. Notably, as demonstrated in the presentation of ALADIN (see `https://www.uiam.sk/~oravec/apvv_sk_cn/slides/aladin.pdf`, page 61), the ALADIN algorithm significantly reduces CPU time per iteration for MPC problems compared to conventional numerical methods [25]. Since ALADIN-β inherits the structure of the standard ALADIN (see Remark 3), it also holds the potential to reduce the computational time required to solve MPCC problems.

VI. Conclusion

This work presents ALADIN-β, a novel distributed algorithm that integrates exact-penalty methods to efficiently and robustly solve MPCC problems in parallel. By systematically avoiding failures caused by violated CQs in ill-conditioned problems, ALADIN-β relaxes the LICQ requirements of classical ALADIN, expanding its applicability. Numerical
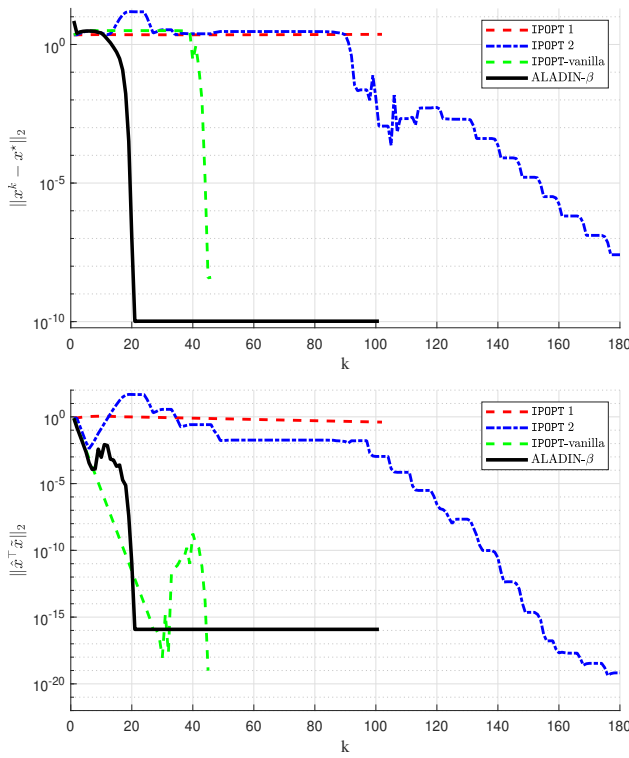
---

[6]In ALADIN β, the parameters ρ and μ are updated at each iteration. In contrast, `IPOPT2` [12] updates these parameters only after solving the current barrier problem to optimality for fixed values of ρ and μ. This fundamental discrepancy in parameter update strategies renders a direct performance comparison between the two methods inherently unfair. To establish a more equitable basis for comparison, we introduce `IPOPT1`, which follows the algorithmic framework proposed in [12] but incorporates parameter updates for ρ and μ at every iteration, thereby aligning its update scheme with that of ALADIN-β.

Fig. 2: Convergence comparison among ALADIN-β and two types of `IPOPT`.

experiments on a benchmark MPCC demonstrates its superior convergence speed and accuracy compared to state-of-the-art NLP solver variants. Future work will focus on formalizing convergence proofs, incorporating globalization strategies, and developing adaptive parameter tuning to enhance robustness for complex MPCCs.

REFERENCES

[1] S. Leyffer, "Mathematical programs with complementarity constraints," *SIAG/OPT Views-and-News*, vol. 14, no. 1, pp. 15–18, 2003.

[2] A. Nurkanović, "Numerical methods for optimal control of nonsmooth dynamical systems," Ph.D. dissertation, University of Freiburg, 2023.

[3] A. J. van der Schaft and J. M. Schumacher, "Complementarity modeling of hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 483–490, 1998.

[4] S. Katayarna, T. Taniai, and K. Tanaka, "Quasistatic contact-rich manipulation via linear complementarity quadratic programming," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 203–210.

[5] S. R. Kazi, K. Wang, and L. T. Biegler, "On optimal control of hybrid dynamical systems using complementarity constraints," *arXiv preprint arXiv:2503.03879*, 2025.

[6] X. Du, S. Zhu, Y. Wang, B. Han, and X. He, "Optimal demand shut-offs of AC microgrid using AO-SBQP method," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 1431–1436, 2023.

[7] A. Nurkanović, A. Pozharskiy, and M. Diehl, "Solving mathematical programs with complementarity constraints arising in nonsmooth optimal control," *Vietnam Journal of Mathematics*, pp. 1–39, 2024.

[8] Y. Kim, S. Leyffer, and T. Munson, "MPEC methods for bilevel optimization problems," *Bilevel Optimization: Advances and Next Challenges*, pp. 335–360, 2020.

[9] J. Hall, A. Nurkanović, F. Messerer, and M. Diehl, "A sequential convex programming approach to solving quadratic programs and optimal control problems with linear complementarity constraints," *IEEE Control Systems Letters*, vol. 6, pp. 536–541, 2022.

[10] J. Hall, A. Nurkanović, F. Messerer, and M. Diehl, "LCQPow: a solver for linear complementarity quadratic programs," *Mathematical Programming Computation*, 2024.

[11] S. Zhu and X. Du, "Alternating direction based sequential boolean quadratic programming method for transmit antenna selection," in *IEEE Conference on Decision and Control*, 2022, p. 7491–7496.

[12] D. Thierry and L. Biegler, "The ℓ_1-exact penalty-barrier phase for degenerate nonlinear programming problems in ipopt," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6496–6501, 2020.

[13] A. Nedić, *Distributed Optimization Over Networks*. Springer International Publishing, 2018, pp. 1–84.

[14] A. Engelmann, Y. Jiang, T. Mühlpfordt, B. Houska, and T. Faulwasser, "Toward distributed OPF using ALADIN," *IEEE Transactions on Power Systems*, vol. 34, no. 1, pp. 584–594, 2019.

[15] X. Du, A. Engelmann, Y. Jiang, T. Faulwasser, and B. Houska, "Distributed state estimation for AC power systems using Gauss-Newton ALADIN," in *IEEE Conference on Decision and Control*, 2019, pp. 1919–1924.

[16] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[17] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, "DLM: decentralized linearized alternating direction method of multipliers," *IEEE Transactions on Signal Processing*, vol. 63, no. 15, pp. 4051–4064, 2015.

[18] B. Houska, J. Frasch, and M. Diehl, "An augmented Lagrangian based algorithm for distributed nonconvex optimization," *SIAM Journal on Optimization*, vol. 26, no. 2, pp. 1101–1127, 2016.

[19] X. Du and J. Wang, "Distributed consensus optimization with consensus ALADIN," in *American Control Conference*, 2025 (accepted for publication).

[20] X. Du, X. Zhou, S. Zhu, and A. I. Rikos, "Convergence theory of flexible aladin for distributed optimization," in *European Control Conference*, 2025 (accepted for publication).

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

[22] B. Houska and Y. Jiang. Springer, 2021, ch. Distributed Optimization and Control with ALADIN, pp. 135–163.

[23] X. Du, J. Wang, X. Zhou, and Y. Mao, "A bi-level globalization strategy for non-convex consensus ADMM and ALADIN," *arXiv preprint arXiv:2309.02660*, 2023.

[24] A. Engelmann, Y. Jiang, B. Houska, and T. Faulwasser, "Decomposition of nonconvex optimization via bi-level distributed ALADIN," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 4, pp. 1848–1858, 2020.

[25] D. Kouzoupis, R. Quirynen, B. Houska, and M. Diehl, "A block based ALADIN scheme for highly parallelizable direct optimal control," in *American Control Conference*, 2016, pp. 1124–1129.

[26] J. Shi, Y. Jiang, J. Oravec, and B. Houska, "Parallel mpc for linear systems with state and input constraints," *IEEE Control Systems Letters*, vol. 7, pp. 229–234, 2023.

[27] G. Stomberg, A. Engelmann, and T. Faulwasser, "Decentralized nonconvex optimization via bi-level SQP and ADMM," in *IEEE Conference on Decision and Control*, 2022, pp. 273–278.

[28] S. Wu, Y. Wang, J. Wang, A. I. Rikos, and X. Du, "A time splitting based optimization method for nonlinear MHE," in *IEEE Conference on Decision and Control*, 2025 (accepted for publication).

[29] B. Houska and J. Shi, "Distributed MPC with ALADIN-a tutorial," in *American Control Conference*, 2022, pp. 358–363.

[30] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, New York, 2006.

[31] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.