*Journal of Industrial and Management Optimization*

https://www.aimspress.com/journal/jimo

JIMO, 22(12): 6908–6935.
DOI:10.3934/jimo.2026230
Received: 27 August 2026
Revised: 12 November 2026
Accepted: 19 November 2026
Published: 26 November 2026

*Research article*

# Consensus multiplier methods for heterogeneous reverse-convex programs

**Zhonghui Xue*** **and Yu Chen**

Shanghai Publishing and Printing College, Shanghai, China

* **Correspondence:** Email: hnlgxzh@163.com.

**Abstract:** We studied finite-horizon cooperative packing with heterogeneous linear dynamics, convex local constraints, quadratic costs, and reverse-convex pairwise disk clearance. Nonlinear disk indicators give an exact consensus lifting; fixed contact halfspaces give a convex inner approximation. For a specified synchronous local/copy multiplier iteration, a weighted distance to a saddle of the fixed-halfspace problem decreases and bounds the multipliers. A retained-edge dual procedure implements the joint weighted copy projection using neighbor exchanges, with convergence even when contact rows are redundant. A feasible normal-refresh outer iteration has original-disk Karush–Kuhn–Tucker cluster points under an explicit uniform strict-point condition, allowing active local input and corridor bounds. Separately, a jammed antipodal polygon is a strict saddle in a reduced position model. Its fixed-point statement requires coincident historical copies and specified multipliers; its stale-normal estimate is local. Using five shared feasible starts, we compared the implemented normal-refresh method with external nonlinear solvers using the same original-disk feasibility, stationarity, complementarity, and multiplier-sign thresholds. The proposed method and two released SciPy solvers passed this screen on all starts; an additional independent ALADIN reference was reported by variant, including unsuccessful runs. The results distinguish surrogate convergence, original-problem stationarity, and constrained saddle behavior.

**Keywords:** nonconvex programming; Karush–Kuhn–Tucker conditions; reverse-convex constraints; unilateral contact; partial asynchrony; alternating direction method of multipliers; distributed optimization; multi-agent systems
**Mathematics Subject Classification:** 90C26, 90C30, 65K05

## 1. Introduction

We consider $N$ agents with unrolled linear dynamics, convex local constraints, and convex quadratic costs. Pairwise disk clearance at each predicted time is a reverse-convex inequality in the extracted planar positions; once dynamics are eliminated, cooperative packing becomes a nonconvex quadratically

constrained quadratic program [1]. Similar structures appear in open-loop subproblems of distributed model predictive control (MPC) [2] and in multi-agent corridor or warehouse coordination [3, 4], where agents may differ in dynamics, feasible sets, and onboard solver cost. This paper studies one heterogeneous finite-horizon instance in an open-loop setting. It does not analyze closed-loop distributed MPC.

The alternating direction method of multipliers (ADMM) splits contacts across copies [5–8]. Convergence results for the nonconvex ADMM formulations depend on the splitting, regularity, penalty parameters, and update schedule [9–13]. Different oracle dimensions do not by themselves violate a common smoothness bound. Delayed information and changing contact halfspaces nevertheless prevent a direct application of a fixed-splitting synchronous theorem. Asynchronous master–worker schemes [14, 15] and decentralized methods [16–18] provide relevant convergence frameworks, with different objective and communication assumptions; see Section 3.

A separate first-order difficulty already appears in the synchronous homogeneous case [1, 19, 20]. Biconvex consensus projects onto frozen-normal halfspaces; the hierarchical ADMM method linearizes disk residuals on copies. On a jammed regular polygon with antipodal targets, the two maps behave differently: the configuration is a Karush–Kuhn–Tucker (KKT) strict saddle [21, Thm. 12.5]. The rotational field is tangent to the frozen contact planes. Those planes exclude the curved rigid rotation, while retaining the first-order direction. A short infeasible local step can leave the nonlinear disks. The strict-saddle avoidance results for unconstrained first-order methods [22, 23] do not apply directly to this constrained KKT point, whose objective gradient is nonzero. Pan and Wu [1] motivated the contact-splitting comparison; Theorem 6.19 proves the saddle statement for a reduced position model, separately from the heterogeneous dynamics. Whether heterogeneity and outdated normals alter this contact-flex behavior remains open in that literature.

Table 1 summarizes prior multiplier schemes with respect to bounded copy delay, heterogeneous unrolled oracles, reverse-convex disks, and a feasible-point trap under frozen normals. The analysis separates model equivalence, an exact inner solver, a feasible normal-refresh outer procedure, and reduced contact geometry (Figure 1). This separation prevents the convergence of a convex surrogate from being mistaken for the convergence of an arbitrary delayed disk heuristic. Section 2 states the main theorems; proofs are given in Section 6 and Appendix B in the accompanying appendix.

The contributions are as follows:

1. The disk-indicator lifting is exact, while a fixed-normal halfspace lifting is a convex inner approximation (Lemma 4.1). Their KKT conditions and those of a penalty problem are kept distinct.
2. Exact synchronous local/copy updates decrease a weighted distance to a saddle of the fixed-halfspace problem and bound the consensus multipliers (Theorem 6.6). A neighbor implementation of the joint weighted copy quadratic program (QP) is given and proved convergent (Proposition 6.15); it retains edge multipliers and does not average separately corrected copies.
3. A feasible normal-refresh outer iteration has original-disk KKT cluster points under an explicit uniform strict-point condition for its convex subproblems (Theorem 6.17). Active input and corridor normals are retained in the limiting KKT system.
4. On a reduced position model, the jammed polygon is a KKT strict saddle. A feasible-point fixed point persists with coincident historical copies and specified multipliers; a single-endpoint infeasible step violates a disk only within an explicit displacement interval (Theorems 6.19 and 6.21).

5. A common original-disk KKT evaluator compares the normal-refresh implementation with external nonlinear solvers on five identical feasible starts (Section 7). The common screen includes feasibility, stationarity, complementarity, and multiplier sign, rather than consensus alone.

For finitely many agents, $\max_i L_i$ is a common smoothness bound. Agent-wise metrics are parameter choices, not a new rate. The lifting, Hildreth projection, and feasible inner-approximation principles have established origins; our contribution is their precise contact-model specification, the reduced saddle and trap analysis, and a checked heterogeneous implementation. General convergence of the original stale edge-placement/averaging template is not claimed. The title and theorem scopes distinguish that template from the algorithms actually analyzed.

**Contact model.** Disk indicators lift exactly; fixed halfspaces restrict the disk program.

**Inner solution.** Synchronous local/copy splitting decreases a saddle distance; retained edge multipliers give a neighbor solver for the joint copy QP.

**Outer stationarity.** Feasible normal refresh gives original-disk KKT cluster points under uniform strict points.

**Contact geometry and tests.** The reduced jam is a KKT strict saddle; five shared-start comparisons use the original-disk KKT screen.

**Figure 1.** Scope of the analysis. The delayed-copy statements concern reduced contact geometry. The convergence statements refer to the explicitly specified inner and outer iterations.

## 2. Main results

The program (4.5), lifting (4.8), and contact maps are defined in Sections 4 and 5. The Lyapunov and stationarity statements are proved in Section 6; the trap geometry is treated in Section 6.3; full proofs are in Appendix B. The main statements are listed in the Introduction; Section 7 reports numerical tests corresponding to each block.

## 3. Related work

Nonconvex ADMM analyses [9–13] establish stationarity under their specified block structure and regularity assumptions. Kurdyka–Łojasiewicz analyses [24, 25] offer convergence tools for tame objectives; the polynomial disk constraints are compatible with that framework, but the required descent and relative-error estimates must still be proved for the chosen iteration. Chang et al. [14] analyzed the bounded-delay master–worker ADMM. Mafakheri et al. [17] gave an asynchronous decentralized nonconvex method based on randomized Douglas–Rachford splitting, while Carnevale et al. [18] proved linear convergence for a strongly convex consensus model with asynchronous agents and packet losses. These results do not directly analyze the delayed frozen-contact maps studied here (Table 1).

Table 1 compares the source papers on four points: whether delay is bounded, whether local oracles may differ, whether reverse-convex disks are present, and whether a feasible-point trap is identified under delayed contact normals. The four columns record features investigated in the cited theorems or

examples, not every constraint a general formulation might admit. The Schedule column describes the schedule investigated, rather than asserting convergence under that schedule. An entry of "no" does not mean the source formulation forbids the feature.

**Table 1.** Comparison with source papers. Het. indicates that the cited work investigated the agent-specific data $(A_i, B_i, \mathcal{Z}_i)$. RC indicates that the cited work investigated reverse-convex pairwise disks. Trap indicates that the cited work discussed a feasible-point fixed point at a KKT strict saddle under the stated contact conditions. An entry of "no" means the feature is not part of the cited theorem or example.

| Work | Schedule | Het. | RC | Trap |
|---|---|---|---|---|
| Chang et al. [14] | bounded $\tau$ | no | no | no |
| Hong [15][1] | bounded $\tau$ | no | no | no |
| Kumar et al. [16] | bounded $\tau$ | no | no | no |
| Mafakheri et al. [17] | random wake-up | no | no | no |
| Carnevale et al. [18][2] | packet loss | no | no | no |
| Houska et al. [26][3] | none | yes | no | no |
| Pan–Wu [1] | none | no | yes | no |
| Zhang et al. [19] | none | no | yes | no |
| This paper | sync.; bounded ages[4] | yes | yes | yes[4] |

[1] Master–worker ADMM; reverse-convex contact was not treated.
[2] Strongly convex local blocks only; no reverse-convex disks.
[3] Synchronous coordinator QP; coupling linearized at the current iterate, not frozen unilateral contact.
[4] Bounded copy ages occur in the contact-geometry analysis. Convergence concerns the specified synchronous inner and feasible normal-refresh outer iterations. The trap is a reduced-position fixed point at a jammed polygon with coincident copies and aligned normals, not a claim for arbitrary stale copies.

ALADIN [26] coordinates heterogeneous local nonlinear programs through a convex quadratic subproblem whose coupling constraints are linearizations at the current primal iterate. That subproblem is not a projection onto frozen unilateral halfspaces: normals refresh with the linearization point, and inner iterates need not remain inside the nonlinear feasible set of (4.5). ALADIN targets distributed nonlinear optimal control with local convergence under standard constraint qualifications; it does not analyze reverse-convex disk contact, copy age bounded by $\tau$, or the feasible-point trap at a jammed packing (Table 1). Section 7 compares the original-disk KKT screen across sequential least squares programming (SLSQP), `trust-constr`, the normal-refresh multiplier method, and a documented independent ALADIN reference. The latter is not the official MATLAB implementation.

The table is not a performance ranking. The cited ADMM methods motivate the splitting, while Theorem 6.6 uses a standard convex saddle-distance argument. Pan and Wu [1] motivated the contact-map comparison. The present paper distinguishes the fixed-halfspace synchronous convergence result from the delayed edge-wise tests. At the exact reduced-model fixed point, coincident delayed copies preserve the trap. Neither convergence nor rate guarantees transfer automatically to stale contact updates (Remark 6.13).

Hong, Luo, and Razaviyayn [9] and Wang, Yin, and Zeng [10] proved stationarity for a fixed splitting with a uniform smoothness constant. Agent-wise penalties reweight that argument; they do not define a new complexity class [24, 25, 27].

The partially asynchronous model of Bertsekas and Tsitsiklis [28] bounds delay by a finite $\tau$. Chang et al. [14] established KKT stationarity for a nonconvex consensus problem under a master–worker ADMM when the master's proximal weight grows like $\tau^2$, and they noted that a slightly different implementation can diverge even when the problem is convex. Neighbor-wise schemes [16, 17, 29] remove the master. ADMM-tracking gradients [18] add packet-loss robustness under strong convexity. None of these references treats reverse-convex unilateral contact or agent-specific unrolled dynamics.

Biconvex consensus projects onto a frozen-normal polyhedron and, from a strictly feasible start, keeps inner iterates in $\{g \le 0\}$ [1]. The hierarchical ADMM method embeds a linearized inner step in an outer augmented Lagrangian loop; inner iterates need not be feasible [19, 20]. On a jammed regular polygon, the two maps are not interchangeable as first-order contact iterations. The question here is whether delayed contact normals and mixed $(A_i, B_i)$ change this behavior. At the exact saddle, with delayed copies equal to the current copies, the feasible-point map stays at the same fixed point. The finite rigid-rotation path is excluded by the frozen planes even though its first-order field is tangent to them. Theorem 6.21 does not extend this sentence to arbitrary stale copies.

## 4. A heterogeneous reverse-convex contact program

$\|\cdot\|$ is Euclidean unless otherwise noted. For stacked variables, $\langle \lambda, r\rangle = \sum_i \langle \lambda_i, r_i\rangle$. The copy age $d(k) \in \{0, \dots, \tau\}$ is the delay used at event $k$; $\hat{y}^k = y^{k-d(k)}$. Frozen normals $n_e^{t,k}$ are computed from delayed copies as in (5.3). Symbols $P_i$, $g_{ij}^t$, and $\psi_{e,t}$ are fixed for the entire paper.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected connected graph on $N$ agents [30, 31]. Agent $i \in \mathcal{V}$ has type $\theta_i$, state dimension $n_i$, input dimension $m_i$, and discrete linear dynamics

$$x_i(t+1) = A_i x_i(t) + B_i u_i(t), \qquad t = 0, \dots, T-1, \tag{4.1}$$

with a common horizon $T$; any agent-specific sampling period is absorbed into $(A_i, B_i)$. The types used later are a planar single integrator ($n_i = 2$), a planar double integrator ($n_i = 4$), and a planar triple integrator ($n_i = 6$). We stack

$$z_i = \big(u_i(0), \dots, u_i(T-1), x_i(1), \dots, x_i(T)\big) \in \mathbb{R}^{T(m_i+n_i)}. \tag{4.2}$$

Unrolling (4.1) produces the affine equality $G_i z_i = b_i(x_i(0))$. Equivalently, one can eliminate this equality and use the complete input sequence $u_i$ as independent coordinates, with $z_i = z_i^0 + S_i u_i$ and extracted positions $p_i^0 + Q_i u_i$. In that coordinate system, the quadratic cost, proximal metric, and position-map differences are defined on $u_i$; the fixed position offset cancels from differences. The synchronous numerical comparison uses these condensed coordinates. Local convex inequalities (input boxes, speed bounds, and a corridor $|p_{i,y}| \le Y_i$ for selected agents) are written as $F_i z_i \le h_i$. The local set

$$\mathcal{Z}_i := \{z_i : G_i z_i = b_i(x_i(0)),\ F_i z_i \le h_i\} \tag{4.3}$$

is nonempty, compact, and convex. Each tracking cost $f_i$ is a convex quadratic and hence is of class $C^{1,1}$.

Let $P_i$ extract the sequence of planar positions, so $P_i z_i \in \mathbb{R}^{2T}$, and write $P_i^t$ for the block at predicted time $t$. A pair $e = (i, j) \in \mathcal{E}$ must satisfy

$$\|P_i^t z_i - P_j^t z_j\| \geq d_{ij} := r_i + r_j, \qquad t = 1, \ldots, T. \tag{4.4}$$

The feasible set of (4.4) is the complement of an open ball, so it is reverse-convex. Each pair is a unilateral contact of two rigid disks. A frozen unit normal is the first-order linearization of that contact. Static obstacles are frozen disks of the same kind. The radii $r_i$ need not be equal.

The program is

$$\begin{aligned} \min_{(z_i)_{i\in\mathcal{V}}} \quad & \sum_{i\in\mathcal{V}} f_i(z_i) \\ \text{s.t.} \quad & z_i \in \mathcal{Z}_i, && i \in \mathcal{V}, \\ & \|P_i^t z_i - P_j^t z_j\| \geq d_{ij}, && (i,j) \in \mathcal{E},\ t = 1, \ldots, T. \end{aligned} \tag{4.5}$$

Different $(A_i, B_i, n_i, m_i, \mathcal{Z}_i, r_i)$ appear only in the local data of agent $i$. The coupling is exclusively through (4.4). Nonlinear unicycle kinematics are excluded: they would destroy the convexity of $\mathcal{Z}_i$.

We write

$$g_{ij}^t(z) := d_{ij}^2 - \left\|P_i^t z_i - P_j^t z_j\right\|^2 \tag{4.6}$$

for the reverse-convex residual, so (4.4) is $g_{ij}^t(z) \leq 0$. The partial gradient in $z_i$ has the two-block form

$$\nabla_{z_i} g_{ij}^t(z) = -2(P_i^t)^\top (P_i^t z_i - P_j^t z_j) \tag{4.7}$$

and $\nabla_{z_j} g_{ij}^t(z) = 2(P_j^t)^\top (P_i^t z_i - P_j^t z_j)$, with zeros in all other blocks.

For each agent $i$, we introduce a copy $y_i \in \mathbb{R}^{2T}$ of the extracted positions, and write $y = (y_i)_{i\in\mathcal{V}}$. Stack $y_i = (y_i^1, \ldots, y_i^T)$ with $y_i^t \in \mathbb{R}^2$ the planar position at predicted time $t$. For edge $e = (i, j)$ and time $t$, the reverse-convex pair constraint is $\|y_i^t - y_j^t\| \geq d_{ij}$. The lifted program is

$$\begin{aligned} \min_{z,y} \quad & \sum_{i\in\mathcal{V}} f_i(z_i) + \sum_{e\in\mathcal{E}} \sum_{t=1}^{T} \psi_{e,t}(y_i^t, y_j^t) \\ \text{s.t.} \quad & z_i \in \mathcal{Z}_i, && i \in \mathcal{V}, \\ & P_i z_i = y_i, && i \in \mathcal{V}. \end{aligned} \tag{4.8}$$

The symbol $\psi_{e,t}$ is not unique. Three liftings are used below, and later theorems name which one they address.

1. Disk lifting: $\psi_{e,t}$ is the indicator of $\{(y_i^t, y_j^t) : \|y_i^t - y_j^t\| \geq d_{ij}\}$.
2. Halfspace lifting: $\psi_{e,t} = \iota_{H_e^t(n_e^t)}$ with

$$H_e^t(n_e^t) = \{(y_i^t, y_j^t) : (n_e^t)^\top (y_i^t - y_j^t) \geq d_{ij}\}. \tag{4.9}$$

3. Penalty lifting: $\psi_{e,t}$ is a $C^{1,1}$ penalty of the disk residual on a set bounded away from vanishing separation, or a squared-residual penalty as described below. The heuristic edge correction is a different implementation choice and is not an exact proximal step for this penalty.

Equation (4.8) denotes whichever of these three is in force. Karush–Kuhn–Tucker conditions are stated for that choice, not transferred to (4.5) without an extra alignment argument. The residual uses the agent-wise copy: $r_i(z_i, y_i) = P_i z_i - y_i$, and $C_W = 1$ in (4.8). An incident-edge copy $y^t_{e,i}$ is attached to one edge. An equivalent edge-expanded lifting would require $y^t_{e,i} = y^t_i = P^t_i z_i$ at every incidence. The edge-wise implementation instead defines the aggregation

$$y^t_i = \frac{1}{|\mathcal{N}_i|} \sum_{e \ni i} y^t_{e,i}.$$

This average does not impose equality of all incident copies. Before averaging, each edge pair is the output of (5.5) or (5.6). After it, that pair need not remain feasible for the edge. If the residual is instead written on stacked edge copies through a selector $W_i$, then $\|W_i\| \le C_W$. Theorem 6.10 is the stationarity condition of the surrogate used by the proximal copy map, not of the averaged edge copies.

**Lemma 4.1** (Disk lifting and halfspace restriction)**.** *A pair $(z, y)$ is feasible for the disk lifting if and only if $z$ is feasible for* (4.5) *and $y_i = P_i z_i$ for every $i$. If instead $\psi_{e,t} = \iota_{H^t_e(n^t_e)}$ with $\|n^t_e\| = 1$, every feasible pair satisfies* (4.4) *by the Cauchy–Schwarz inequality. Conversely, a disk-feasible pair belongs to its aligned halfspace, obtained by choosing $n^t_e = (y^t_i - y^t_j)/\|y^t_i - y^t_j\|$. For an arbitrary fixed normal, this reverse implication need not hold; alignment is sufficient, and it is necessary at a contact pair with distance exactly $d_{ij}$. The halfspace lifting is therefore an inner approximation of* (4.5)*, not an equivalent formulation.*

Each $P_i$ has full row rank. The dimensions $(n_i, m_i)$ never enter the consensus constraint $P_i z_i = y_i$. The halfspace implication follows from the Cauchy–Schwarz inequality and is independent of the dynamics.

**Lemma 4.2** (MFCQ under position controllability)**.** *Let $z$ be feasible for* (4.5)*. Suppose that, at $z$, every local inequality $F_i z_i \le h_i$ is inactive, and that for each $i$, the map from the input sequence to $(P^t_i z_i)^T_{t=1}$ is surjective (the pair $(A_i, B_i)$ is position-controllable over the horizon). Then the Mangasarian–Fromovitz constraint qualification (MFCQ) holds at $z$: the simultaneous expansion of extracted positions about their time-wise centroid is a strictly feasible linearized direction for every active disk, and it lies in the tangent space of the affine hull of $\prod_i \mathcal{Z}_i$.*

If a corridor or an input bound is active, the disk-expanding direction of Lemma 4.2 may leave $\mathcal{Z}_i$. Strict local interior points can certify the local inequalities alone, but do not by themselves give a direction that expands all active disks. The separate normal-refresh result in Theorem 6.17 instead uses explicit strict contact points in the actual local set and retains active local-bound normals. Its numerical witness checks are reported in Section 7.3.

## 5. Contact maps and the proximal multiplier iteration

The historical contact templates use an event-driven notation. The joint proximal method below has its own fixed-halfspace iteration and the neighbor QP solver in Section 6.1; the two should not be confused. A global counter $k \in \mathbb{N}$ labels events. At iteration $k$, a nonempty subset $\mathcal{A}^k \subset \mathcal{V}$ is awake: only those oracles fire. Agent $i$ uses a delayed copy stack

$$\hat{y}^k := y^{k-d(k)}, \qquad 0 \le d(k) \le \tau, \tag{5.1}$$

with a finite horizon $\tau$ [14, 28]. (The same bound applies agent-wise: $d_i(k) \le \tau$.) Every agent wakes at least once in every window of length $\max\{1, \tau\}$. The synchronous method is $\tau = 0$ and $\mathcal{A}^k = \mathcal{V}$. This is the partially asynchronous computation model; the flop-weighted event clock of Section 7 is its heterogeneous-oracle accounting, not a shared-memory runtime.

Let $\lambda_i$ be the multiplier of $P_i z_i = y_i$ and write $r_i(z_i, y_i) := P_i z_i - y_i$. The weighted augmented Lagrangian of (4.8) is

$$\begin{aligned}\mathcal{L}(z, y, \lambda) = &\sum_{i\in\mathcal{V}} f_i(z_i) + \sum_{e\in\mathcal{E}}\sum_{t=1}^{T} \psi_{e,t}(y_i^t, y_j^t) \\ &+ \sum_{i\in\mathcal{V}} \langle \lambda_i, r_i(z_i, y_i)\rangle + \sum_{i\in\mathcal{V}} \tfrac{\rho_i}{2}\|r_i(z_i, y_i)\|^2.\end{aligned} \tag{5.2}$$

The penalties $\rho_i > 0$ need not be equal: they are the local metric of a Lipschitz-asymmetric splitting. A proximal weight $\gamma_i > 0$ is attached to the $z_i$-update. A copy proximal weight $\eta \ge 0$ can be included in the event skeleton; the convergence result uses $\eta = 0$ and current copies. We write $C_W$ for the operator-norm bound of the copy selector, with $C_W = 1$ in (4.8).

The two templates are two linearizations of the same unilateral contact. The feasible-point local step minimizes over a frozen-normal polyhedron; its edge copy placement is defined explicitly below. The infeasible-point map takes a damped step of the disk residual (contact not restored). Listings of the two edge-wise maps are in Appendix A. They are structural reductions of [1, 19], not line-by-line clones.

### *5.1. Feasible-point map: frozen-contact projection*

The feasible-point template takes $\psi_{e,t} = \iota_{H_e^t(n_e^{t,k})}$ with

$$n_e^{t,k} = \frac{\hat{y}_i^{t,k} - \hat{y}_j^{t,k}}{\|\hat{y}_i^{t,k} - \hat{y}_j^{t,k}\|}, \tag{5.3}$$

or the previous unit vector if the denominator vanishes. For each oriented edge $e = (i, j)$, we use $n_{ij}^{t,k} = n_e^{t,k}$ at endpoint $i$ and $n_{ji}^{t,k} = -n_e^{t,k}$ at endpoint $j$. The local step is the unique minimizer of a strongly convex quadratic on the convex set

$$\hat{\mathcal{Z}}_i^k := \mathcal{Z}_i \cap \{z_i : (n_{ij}^{t,k})^\top (P_i^t z_i - \hat{y}_j^{t,k}) \ge d_{ij} \ \forall j \in \mathcal{N}_i, \ \forall t\}, \tag{5.4}$$

which is closed and convex. Nonemptiness is an additional requirement at each event: a feasible point of (4.5) need not remain feasible for the local cuts formed with stale copies. The historical edge copy formula is not the Euclidean projection onto the halfspace. That projection preserves the component of $p_i^t - p_j^t$ orthogonal to $n_e^{t,k}$. The formula below replaces the whole separation by $d_{ij} n_e^{t,k}$. If $(n_e^{t,k})^\top (p_i^t - p_j^t) < d_{ij}$,

$$\begin{aligned} y_{e,i}^{t,+} &= \tfrac{1}{2}(p_i^t + p_j^t) + \tfrac{1}{2} d_{ij}\, n_e^{t,k}, \\ y_{e,j}^{t,+} &= \tfrac{1}{2}(p_i^t + p_j^t) - \tfrac{1}{2} d_{ij}\, n_e^{t,k}, \end{aligned} \tag{5.5}$$

and $(y_{e,i}^{t,+}, y_{e,j}^{t,+}) = (p_i^t, p_j^t)$ otherwise, where $p_i^t = P_i^t z_i^{k+1}$. Since $\|n_e^{t,k}\| = 1$, the Cauchy–Schwarz inequality yields $\|y_{e,i}^{t,+} - y_{e,j}^{t,+}\| \ge d_{ij}$ for this edge pair, independently of the age of $n_e^{t,k}$. Averaging incident-edge

copies is nonexpansive, but it need not keep that pair inside $H_e^t(n_e^{t,k})$ or inside the disk. Proposition 5.2 stops before averaging. The joint halfspace copy update used in Theorem 6.6 instead solves one convex quadratic program on all copies, with current data, fixed normals, and $\eta = 0$. It differs from (5.5) followed by averaging when edges share an agent. The revised numerical evaluation uses the joint proximal map; the historical edge formulas are retained for the geometry discussion.

**Assumption 5.1** (Strictly feasible start)**.** *A point $z^0 \in \prod_i \mathcal{Z}_i$ with $g_{ij}^t(z^0) < 0$ for every pair and time is given, and the initial copies equal the extracted positions.*

**Proposition 5.2** (Disk inequality before averaging)**.** *Assume Assumption 5.1 and that every feasible-point z-update is the exact minimizer of the local quadratic on a nonempty $\hat{\mathcal{Z}}_i^k$. Then every edge pair produced by* (5.5)*, before averaging, satisfies* (4.4)*. Averaging can destroy the inequality on a given edge. The proposition does not apply to the averaged copies. If a pair is not averaged and $P_i z_i = y_i$ after the event, the corresponding primal positions satisfy* (4.4)*.*

Delay changes which halfspaces are used, not the per-time implication from a unit-normal plane to a disk. The proposition does not assert that the halfspaces remain aligned with the current positions, nor that $\hat{\mathcal{Z}}_i^k$ stays nonempty for an arbitrary stale stack.

### *5.2. Infeasible-point map: residual linearization*

The infeasible-point template does not restore copies onto the disk. The ideal noncoincident-pair map below describes the historical damped correction. The Hessian of $\frac{1}{2}(d - \|\Delta\|)_+^2$ blows up as $\|\Delta\| \to 0$; no uniform smooth-distance argument covers coincident positions. A differentiable penalty analysis would need either separation bounded below by $\delta > 0$ or a replacement such as $\frac{1}{2}(d_{ij}^2 - \|\Delta\|^2)_+^2$, whose gradient is Lipschitz on bounded sets. The convergence theorem below concerns fixed halfspaces and does not cover this penalty map. After the local step on $\mathcal{Z}_i$ (no frozen halfspaces), overlapping pairs in the test map are shifted at each $t$ by a damped correction of size $\beta \in (0, 1)$: if $\|p_i^t - p_j^t\| < d_{ij}$,

$$\begin{aligned} y_{e,i}^{t,+} &= p_i^t + \beta(d_{ij} - \|p_i^t - p_j^t\|)\frac{p_i^t - p_j^t}{\|p_i^t - p_j^t\|}, \\ y_{e,j}^{t,+} &= p_j^t - \beta(d_{ij} - \|p_i^t - p_j^t\|)\frac{p_i^t - p_j^t}{\|p_i^t - p_j^t\|}, \end{aligned} \tag{5.6}$$

and $(y_{e,i}^{t,+}, y_{e,j}^{t,+}) = (p_i^t, p_j^t)$ otherwise. Let $s = \|p_i^t - p_j^t\|$. For $s \in (0, d_{ij})$, the two-sided step gives $\|y_{e,i}^{t,+} - y_{e,j}^{t,+}\| = s + 2\beta(d_{ij} - s)$. The copies still overlap if and only if $\beta < 1/2$. At $\beta = 1/2$, the updated distance equals $d_{ij}$. At $s = 0$, the direction $(p_i^t - p_j^t)/s$ is undefined, so (5.6) does not specify the coincident case. This formula is not a copy minimization covered by Theorem 6.6. The archived historical runs used $\beta = 0.08 < 1/2$ and $\eta = 0$; the corrected benchmark does not use this edge map. The legacy implementation replaces each normalization denominator $s$ by $s + 10^{-12}$, including in the feasible-point copy formula. At exact coincidence, this gives zero direction and zero infeasible correction; it does not define a unit contact normal. The exact unit-normal proposition therefore does not cover that implementation at coincidence. All edges are processed at each legacy event, so sleeping-agent copies can change while their multipliers remain fixed. No descent claim is made for this implementation. The supplementary $\eta > 0$ diagnostic blends the averaged copy with its previous value as $(y_{\text{raw}} + \eta y^k)/(1 + \eta)$; this also differs from a joint proximal QP.

**Algorithm 5.1** Synchronous local and joint-copy iteration on fixed halfspaces

**Require:** Fixed $\mathcal{Y}$, current $(z^k, y^k, \lambda^k)$, $\rho_i > 0$, $\gamma_i > 0$

1: Each agent independently minimizes $f_i(z_i) + \frac{\rho_i}{2}\|P_i z_i - y_i^k + \lambda_i^k/\rho_i\|^2 + \frac{\gamma_i}{2}\|z_i - z_i^k\|^2$ over $z_i \in \mathcal{Z}_i$ to obtain $z_i^{k+1}$

2: Solve the joint copy program (5.7), either directly or by the neighbor iterations (6.7), to obtain $y^{k+1}$

3: Every agent sets $\lambda_i^{k+1} = \lambda_i^k + \rho_i(P_i z_i^{k+1} - y_i^{k+1})$

For the fixed-halfspace proximal mode, let $\mathcal{Y} = \bigcap_{e,t} H_e^t(n_e^t)$ with fixed normals and define

$$y^{k+1} = \operatorname*{argmin}_{y \in \mathcal{Y}} \sum_i \left( \langle \lambda_i^k, P_i z_i^{k+1} - y_i \rangle + \frac{\rho_i}{2}\|P_i z_i^{k+1} - y_i\|^2 \right). \tag{5.7}$$

Its centers are $P_i z_i^{k+1} + \lambda_i^k/\rho_i$. This is a joint convex quadratic program, including each shared copy only once.

Algorithm 5.1 is the iteration analyzed in Theorem 6.6. Its local set is $\mathcal{Z}_i$ and every contact halfspace is assigned to the copy block. The neighbor solver in Section 6.1 implements that joint copy map without a distinguished fusion node; its exact limit is the QP minimizer. Different $(A_i, B_i)$ change only local affine maps and costs. The two edge formulas above are not substituted for the copy QP. Their delayed contact geometry is treated in Section 6.3; no general convergence theorem for their stale edge-placement/averaging implementation is claimed.

## 6. Convergence and stationarity

The synchronous argument compares the proximal steps with a saddle and applies the three-point identity (Theorem 6.6). It does not absorb a delay into $V^k = \mathcal{L} + \Delta^k$. The delay identity (B.2) is an algebraic expansion in the Appendix. The Lyapunov function in Theorem 6.6 is the weighted saddle distance $D^k$. Agent-wise Lipschitz constants enter only through Assumption 6.4 and Proposition 6.14; the trap block (Section 6.3) is geometric and does not invoke Theorem 6.6.

**Assumption 6.1** (Local smoothness)**.** *Each $f_i$ is $C^{1,1}$ with Lipschitz constant $L_i$ of $\nabla f_i$. The constants $L_i$ need not coincide.*

**Assumption 6.2** (Local sets)**.** *Each $\mathcal{Z}_i$ in* (4.3) *is nonempty, compact, and convex, and $G_i$ has full row rank.*

**Assumption 6.3** (Partial asynchrony)**.** *There exists $\tau < \infty$ such that the snapshot delay satisfies $d(k) \le \tau$ and every individual copy satisfies $d_i(k) \le \tau$, including the copy of a sleeping agent. For $q = \max\{1, \tau\}$, every agent wakes at least once in every window of length $q$: $\bigcup_{s=0}^{q-1} \mathcal{A}^{k+s} = \mathcal{V}$. Thus $\tau = 0$ means that all agents update at every iteration.*

**Assumption 6.4** (Weights)**.** *The synchronous proximal iteration uses fixed weights*

$$\rho_i > 0, \qquad \gamma_i > 0. \tag{6.1}$$

*It takes $\eta = 0$. These conditions apply to the frozen-halfspace problem and are not sufficient conditions for convergence of either delayed edge-wise map.*

No delay-squared sufficient-weight condition is established for the edge-wise maps. The dual ascent and the distinction between those maps and a joint proximal update must be accounted for in any such argument. Theorem 6.6 proves a saddle-distance decrease for a fully specified synchronous iteration. The historical delayed edge-wise runs are archived and excluded from the corrected benchmark.

The next lemma only bounds the error in reading stale copies. It does not establish asynchronous convergence.

**Lemma 6.5** (Delay perturbation)**.** *Under Assumption 6.3,*

$$\|\hat{y}^k - y^k\|^2 \le \tau \sum_{s=1}^{\tau} \|y^{k-s+1} - y^{k-s}\|^2.$$

**Theorem 6.6** (Convergence of the synchronous proximal map)**.** *Assume Assumptions 6.1, 6.2, and 6.4, $\tau = 0$, and $\eta = 0$. Every agent is awake at every iteration, and every copy used in the update is the current copy. The halfspaces are frozen, so their intersection $\mathcal{Y}$ is closed and convex. Each $f_i$ is a quadratic that is strongly convex on the affine feasible coordinates: for some $\mu_i > 0$, $\langle \nabla f_i(z) - \nabla f_i(w), z - w\rangle \ge \mu_i \|z - w\|^2$ for $z, w \in \mathcal{Z}_i$. The same theorem may instead be formulated in independent input coordinates after eliminating the dynamics. In that case, both the proximal norm and $D^k$ use those coordinates, as in the numerical proximal run. This changes the proximal metric from the full-trajectory Euclidean metric. The position map is then affine; $P_i$ in differences and adjoints denotes its linear part, and affine offsets cancel in differences. The $z_i$-step is the exact minimizer on $\mathcal{Z}_i$. The y-step is the joint proximal map on all copies, with the multiplier inner product and the frozen halfspaces, as in Algorithm 5.1. The dual step is $\lambda_i^{k+1} = \lambda_i^k + \rho_i(P_i z_i^{k+1} - y_i^{k+1})$. Edge-wise placement* (5.5) *followed by averaging is not this iteration. Let $(z^\star, y^\star, \lambda^\star)$ be a saddle of the frozen-halfspace problem, and set*

$$D^k := \sum_i \left( \frac{\gamma_i}{2} \|z_i^k - z_i^\star\|^2 + \frac{\rho_i}{2} \|y_i^k - y_i^\star\|^2 + \frac{1}{2\rho_i} \|\lambda_i^k - \lambda_i^\star\|^2 \right). \tag{6.2}$$

*Then*

$$\begin{aligned} D^k - D^{k+1} \ge & \sum_i \mu_i \|z_i^{k+1} - z_i^\star\|^2 + \sum_i \frac{\gamma_i}{2} \|z_i^{k+1} - z_i^k\|^2 \\ & + \sum_i \frac{\rho_i}{2} \|P_i z_i^{k+1} - y_i^k\|^2 + \langle \lambda^{k+1} - \lambda^\star,\, y^{k+1} - y^\star \rangle \end{aligned} \tag{6.3}$$

*and $\langle \lambda^{k+1} - \lambda^\star, y^{k+1} - y^\star \rangle \ge 0$. The coefficient of the proximal term is $\gamma_i/2$ for every $\gamma_i > 0$. It is not taken from a delay-absorption constant. Consequently, $D^k$ is nonincreasing and nonnegative, $z^k \to z^\star$, $y^k \to y^\star$, and $\lambda^k$ converges to a multiplier of that saddle. In particular,*

$$\sum_i \frac{1}{2\rho_i} \|\lambda_i^k - \lambda_i^\star\|^2 \le D^k \le D^0,$$

*so the multipliers are bounded by the initial distance to the saddle. This bound does not treat a square-summable residual as a hypothesis. The augmented Lagrangian can increase when the dual step adds $\rho_i \|r_i\|^2$.*

*Remark 6.7* (Scope of Theorem 6.6). Theorem 6.6 assumes $\tau = 0$, $\eta = 0$, $\rho_i > 0$, $\gamma_i > 0$, and an update of every agent. The synchronous proximal run in Section 7 is that iteration. The edge-wise rows use (5.5)

and are not covered by the theorem. Theorem 6.17 separately treats feasible normal refresh under uniform strict contact points. Its exact-minimization statement is distinct from the finite inner tolerances disclosed in Section 7.

Theorem 6.6 uses the weighted saddle distance $D^k$ as its Lyapunov function.

**Lemma 6.8** (When multipliers are bounded)**.** *On an awake agent, $\lambda_i^{k+1} - \lambda_i^k = \rho_i r_i(z_i^{k+1}, y_i^{k+1})$. A sleeping agent keeps $\lambda_i$. Square-summability of the residuals does not imply summability of $\|\lambda^{k+1} - \lambda^k\|$. The bound in Theorem 6.6 is $D^k \le D^0$, which starts from a saddle rather than from that implication. If $\sum_k \sum_{i\in\mathcal{A}^k} \|r_i(z_i^{k+1}, y_i^{k+1})\| < \infty$, then $\{\lambda^k\}$ converges.*

**Lemma 6.9** (Limit of the frozen halfspaces)**.** *Suppose $y^{k_j} \to y^\star$, $\hat{y}^{k_j} \to y^\star$, and $\|y_i^{\star,t} - y_j^{\star,t}\| \ge \delta > 0$ on every edge and time used to form a normal. Then $n_e^{t,k_j}$ converges to the aligned unit vector at the limit separation. For each i, suppose in addition that there is $w_i \in \mathcal{Z}_i$ satisfying every limit contact halfspace in* (5.4) *strictly. Then $\hat{\mathcal{Z}}_i^{k_j}$ converges in the Painlevé–Kuratowski sense to $\mathcal{Z}_i$ cut by the limit contact halfspaces. If $z_i^{k_j} \to z_i$ and $v_i^{k_j} \in N_{\hat{\mathcal{Z}}_i^{k_j}}(z_i^{k_j})$ with $v_i^{k_j} \to v_i$, then $v_i \in N_{\hat{\mathcal{Z}}_i^\star}(z_i)$. Without the strict-point hypothesis, only outer set convergence is asserted; boundedness of individual constraint multipliers is not asserted.*

**Theorem 6.10** (Halfspace stationarity and disk multiplier conversion)**.** *Under Theorem 6.6, the limit satisfies the KKT system of the fixed halfspace lifting, with $\psi = \iota_{\mathcal{Y}}$:*

$$\begin{aligned} 0 &\in \nabla f_i(z_i) + P_i^\top \lambda_i + N_{\mathcal{Z}_i}(z_i), \\ 0 &\in N_{\mathcal{Y}}(y) - \lambda, \\ 0 &= P_i z_i - y_i. \end{aligned} \tag{6.4}$$

*Here the local step is on $\mathcal{Z}_i$; all contact halfspaces are in $\mathcal{Y}$. If a different algorithm minimizes over the cut set $\hat{\mathcal{Z}}_i^k$, its local optimality condition contains $N_{\hat{\mathcal{Z}}_i^k}$ instead, and that extra normal cannot be dropped. Such an algorithm is not covered by Theorem 6.6.*

*If every active halfspace normal is aligned with the corresponding limit separation, this KKT point is also KKT for* (4.5)*. In fact, an active halfspace then has $\|p_i^t - p_j^t\| = d_{ij}$, and a nonnegative halfspace multiplier $\nu_{e,t}$ gives the disk multiplier $\mu_{e,t} = \nu_{e,t}/(2d_{ij})$. The multiplier of each inactive halfspace is zero. Active local bounds remain in $N_{\mathcal{Z}_i}$. This conversion uses the already existing halfspace multipliers and needs no additional MFCQ assumption. Lemma 4.2 gives a separate sufficient constraint qualification for the disk program when local inequalities are inactive. No constraint qualification is claimed for arbitrary active local constraints.*

*The theorem proves no convergence result for the nonconvex disk-indicator or penalty liftings, or for the delayed edge-wise maps. Their stationarity conditions are different; numerical alignment to a finite tolerance is not an exact disk KKT certificate.*

**Corollary 6.11** (Synchronous heterogeneous limit)**.** *Under all hypotheses of Theorem 6.6, including $\tau = 0$ and an update of every agent, the iteration converges to a saddle of the frozen-halfspace problem, with multipliers bounded by $D^0$. Theorem 6.10 applies to that limit. Disk stationarity of* (4.5) *holds when the frozen normals are aligned with the separations; active input bounds remain in the normal cone of $\mathcal{Z}_i$.*

**Corollary 6.12** (Square-summable proximal increments)**.** *Under Theorem 6.6,*

$$\sum_k \left( \sum_i \|z_i^{k+1} - z_i^k\|^2 + \sum_i \|P_i z_i^{k+1} - y_i^k\|^2 \right) < \infty$$

*and* $Pz^k - y^k \to 0$*. A square-summable residual is not the hypothesis that bounds* $\lambda$ *(Lemma 6.8). The bound is* $D^k \le D^0$*.*

The decrease (6.3) is proved in Appendix B from the saddle inequalities, strong convexity, and the three-point identity. The delay-absorption weight of [14] is not used.

*Remark 6.13* (Complexity outlook). Theorem 6.6 proves a decrease of $D^k$ and convergence for one convex frozen-halfspace iteration. It does not give an oracle bound for the delayed nonconvex disk program. Complexity bounds for other ADMM splittings require their own update, smoothness, penalty, and residual assumptions; no rate is transferred here. A KKT strict saddle is already first-order stationary. Reaching it does not by itself preclude a complexity bound for a first-order residual; it precludes a guarantee of local minimality from that residual alone (Definition 6.18 and Theorem 6.21). The physical feasibility column of Table 2 is distinct from its stationarity and complementarity columns.

**Proposition 6.14** (Uniform versus agent-wise penalties)**.** *The uniform choice* $\rho \equiv \max_i \rho_i$*,* $\gamma \equiv \max_i \gamma_i$*, and the same* $\eta$ *satisfies Assumption 6.4 whenever the agent-wise choice does. If* $\gamma_i$ *is chosen proportional to* $L_i$*, then* $(\max_i \gamma_i)/(\min_i \gamma_i) = \max_i L_i / \min_i L_i$*. The uniform proximal weight* $\max_i \gamma_i$ *is admissible and increases regularization on smaller blocks; no rate or objective ordering follows.*

Proposition 6.14 concerns parameter scaling only. Positive weights suffice for the synchronous convex theorem. It does not imply a smaller objective.

### *6.1. A neighbor solver for the joint copy program*

For fixed normals, the joint copy step has the form

$$\min_y \ \frac{1}{2}\|y - c\|_R^2 \qquad \text{s.t.} \quad By \ge d, \tag{6.5}$$

where $R = \operatorname{diag}(w_i I_{2T}) \succ 0$, and the row $a = (e, t)$ of $B$ has $n_e^t$ at endpoint $i$, $-n_e^t$ at endpoint $j$, and zeros elsewhere. Every $n_e^t$ is a unit vector and $d_a = d_{ij} > 0$. For the copy step with no extra proximal term, $w_i = \rho_i$ and $c_i = P_i z_i^{k+1} + \lambda_i^k/\rho_i$. If a term $\eta\|y - y^k\|^2/2$ is included, the same formulation holds with $w_i = \rho_i + \eta$ and $c_i = (\rho_i P_i z_i^{k+1} + \lambda_i^k + \eta y_i^k)/(\rho_i + \eta)$. This algebraic observation does not supply a delayed outer-iteration convergence theorem.

The dual of (6.5) is

$$\max_{\nu \ge 0} \ q(\nu) := \nu^\top (d - Bc) - \frac{1}{2}\nu^\top Q \nu, \qquad Q = BR^{-1}B^\top, \tag{6.6}$$

with primal recovery $y = c + R^{-1}B^\top \nu$. Hildreth's coordinate procedure [32], an equivalent corrected halfspace projection method [33, 34], maintains this recovery identity after every edge update. For $a = (i, j, t)$, set $s_a = B_a y - d_a$ and update

$$\begin{aligned} &\delta_a = \max\left\{0, \nu_a - \frac{s_a}{w_i^{-1} + w_j^{-1}}\right\} - \nu_a, \\ &\nu_a \leftarrow \nu_a + \delta_a, \qquad y_i^t \leftarrow y_i^t + \frac{\delta_a}{w_i} n_e^t, \qquad y_j^t \leftarrow y_j^t - \frac{\delta_a}{w_j} n_e^t. \end{aligned} \tag{6.7}$$

Only the two endpoints and the scalar multiplier of that edge are involved. Begin with $\nu = 0$ and $y = c$, or recover $y$ from a nonnegative warm-start multiplier before the first update. Visit every edge and time cyclically. Disjoint edge updates may run concurrently because they touch disjoint variables. Overlapping updates must be serialized and use current endpoint values; stale unsynchronized writes are outside this algorithm. Unlike placement followed by averaging, this procedure retains an edge correction multiplier across visits.

**Proposition 6.15** (Convergence of the neighbor copy solver)**.** *Suppose $\mathcal{Y} = \{y : By \geq d\}$ is nonempty, every $d_a > 0$, and every $w_i > 0$. Then the cyclic updates* (6.7) *converge in the primal variable to the unique minimizer of* (6.5)*. The dual iterates are bounded and every dual cluster point solves* (6.6)*. Full row rank of B and uniqueness of the dual solution are not required.*

A proof is given in Appendix B.7.

For a frozen static-obstacle row, only the moving endpoint is updated and the diagonal denominator is $w_i^{-1}$. Its affine right-hand side need not be positive after the obstacle center is absorbed into the row. In that extension, a strict feasible point must be assumed explicitly rather than inferred by scaling. The proof then uses its positive minimum slack. The automatic strict-point argument above concerns the moving-agent pair constraints of the stated program.

**Stopping and communication.** At any iterate, primal stationarity is the recovery invariant and dual feasibility is $\nu \geq 0$. Each edge computes $(d_a - B_a y)_+$ and $|\nu_a(B_a y - d_a)|$. Their maxima can be propagated by neighbor max-consensus in at most the graph diameter synchronous rounds, or $N - 1$ rounds on a known connected $N$-node graph. No distinguished fusion node is required. The communication rounds and serialized edge visits are part of the implementation cost. If $y$ is feasible, the primal-dual gap is

$$\tfrac{1}{2}\|y - c\|_R^2 - q(\nu) = \sum_a \nu_a(B_a y - d_a).$$

A distributed sum, or the number of constraints times the maximum absolute complementarity product, bounds this gap. Finite stopping tolerances give an approximate QP solution. An exact outer theorem uses the exact limit or separately justified vanishing inner errors; a fixed numerical tolerance is not silently substituted into an exact-minimization proof.

### *6.2. Feasible normal refresh and original-disk stationarity*

A feasible normal-refresh procedure can be analyzed separately from the old delayed edge-wise template. Its structure is an inner convex approximation [35, 36]; the proof below states the conditions needed for the present disk program. Use equality-eliminated independent coordinates $u$ and write the local feasible set as a nonempty compact convex polyhedron $U = \prod_i U_i$. Let $F(u)$ be the original convex differentiable quadratic tracking objective. For each edge and time, write the affine separation as $\Delta_a(u) = A_a u + b_a$ and the original disk inequality as

$$g_a(u) = d_a^2 - \|\Delta_a(u)\|^2 \leq 0, \qquad d_a > 0.$$

Starting from a disk-feasible $u^0 \in U$, define

$$\begin{aligned} n_a^r &= \frac{\Delta_a(u^r)}{\|\Delta_a(u^r)\|}, \qquad C_r = \{u \in U : (n_a^r)^\top \Delta_a(u) \geq d_a \text{ for all } a\}, \\ u^{r+1} &= \operatorname*{argmin}_{u \in C_r} \left\{ F(u) + \frac{\kappa}{2}\|u - u^r\|^2 \right\}, \qquad \kappa > 0. \end{aligned} \tag{6.8}$$

All contact normals are refreshed at an outer iterate and then held fixed while that convex subproblem is solved. The subproblem may be solved by the synchronous local/copy splitting with the neighbor copy solver above. This is an exact-limit decomposition of one convex subproblem, not permission to mix delayed outer normals with an incomplete inner solve.

**Assumption 6.16** (Uniform strict points for the outer contact cuts)**.** *There is a number $\sigma > 0$ such that for every outer index r, there exists $w^r \in U$ with $(n_a^r)^\top \Delta_a(w^r) \geq d_a + \sigma$ for every contact index a. The points $w^r$ may depend on r. Active local inequalities at $u^{r+1}$ are allowed.*

This is an explicit regularity condition on the convex contact subproblems. A strict point in an individual local set does not by itself establish it. For a recorded outer trajectory, it can be checked numerically by maximizing the common halfspace slack over $U$ at each iterate; positive finite-run margins are diagnostics of the condition, not a proof about all subsequent iterates. The condition can fail, and the following theorem does not omit it.

**Theorem 6.17** (Feasibility and disk KKT under normal refresh)**.** *Suppose* (6.8) *is solved exactly with $\kappa > 0$, $u^0$ is feasible for all disks, and Assumption 6.16 holds. Then every outer iterate is feasible for the original disk program, and*

$$F(u^r) - F(u^{r+1}) \geq \frac{\kappa}{2}\|u^{r+1} - u^r\|^2. \tag{6.9}$$

*The outer increments are square-summable. The contact multipliers of the convex subproblems are uniformly bounded. Every cluster point is a KKT point of the original disk program with its actual local constraints. The assertion includes active local input bounds and corridors; it does not assert whole-sequence convergence, a complexity rate, or convergence to a local minimum.*

A proof is given in Appendix B.7.

**Relation to delayed updates.** The two added results close distinct gaps: the neighbor procedure solves the specified fixed-halfspace copy QP, and the feasible outer procedure gives conditional original-disk KKT accumulation. They do not prove the original event-wise augmented-Lagrangian claim for arbitrary stale normals, sleeping multipliers, or edge correction followed by averaging. Such a claim requires a separate, matched asynchronous algorithm and proof.

### *6.3. Contact flex under delay*

The next statements are geometric. They do not use Theorem 6.6. At a jammed regular packing, the contact graph is a cycle. The rotational field $v_i = Jp_i^\star$ is a nontrivial infinitesimal flex of the nonlinear disks. It is tangent to every frozen contact plane because $n_e^\top(v_i - v_j) = 0$. The planes exclude the curved rigid rotation, which leaves the halfspaces at second order. They do not exclude the first-order field.

**Definition 6.18** (KKT strict saddle)**.** Let $x$ be feasible for a twice differentiable inequality program with objective $f$ and constraints $g_e \le 0$, and let $\mu \ge 0$ be such that $(x, \mu)$ is KKT. Write $L(x, \mu) = f(x) + \sum_e \mu_e g_e(x)$ for the Lagrangian. The critical cone is

$$C(x, \mu) = \{v : \ \langle \nabla g_e(x), v\rangle = 0 \text{ if } \mu_e > 0, \ \langle \nabla g_e(x), v\rangle \le 0 \text{ if } e \text{ is active}\}. \tag{6.10}$$

The pair $(x, \mu)$ is a *KKT strict saddle* if there exists $0 \ne v \in C(x, \mu)$ with $\langle \nabla^2_{xx} L(x, \mu)v, v\rangle < 0$. Under the linear independence constraint qualification (LICQ), this is the failure of the second-order necessary condition [21, Thm. 12.5]; see Figure 4 for the jammed-polygon illustration.

This is not the unconstrained object of [22, 23] (a point with $\nabla f = 0$ and $\lambda_{\min}(\nabla^2 f) < 0$).

Let $M \ge 3$ agents of equal radius $r$ occupy the vertices of a regular $M$-gon of radius $R_\star = d/(2\sin(\pi/M))$, $d = 2r$, so adjacent disks are in contact, and let the tracking targets be the antipodal points $t_i = -\alpha p_i^\star$ with $\alpha > 0$. Write $p^\star$ for the stacked planar positions and $\mathcal{E}_\circ$ for the Hamiltonian cycle of adjacent contacts.

**Theorem 6.19** (Jammed antipodal packing is a KKT strict saddle)**.** *Let $f(p) = \sum_i \frac{1}{2}\|p_i - t_i\|^2$ on $\mathbb{R}^{2M}$ with $t_i = -\alpha p_i^\star$, and let $g_e(p) = d^2 - \|p_i - p_j\|^2$ on $\mathcal{E}_\circ$. Then $(p^\star, \mu)$ is KKT with ring multipliers*

$$\mu_\circ = \frac{1 + \alpha}{4(1 - \cos(2\pi/M))} > 0.$$

*The LICQ holds at $p^\star$. The rotational field $v_i = Jp_i^\star$, with $J$ a $90^\circ$ rotation, lies in the critical cone, and*

$$\langle \nabla^2_{pp} L(p^\star, \mu)v, v\rangle = -M\alpha R_\star^2 < 0.$$

*Every frozen-normal halfspace $H_e^t(n_e^{t,\star})$ with $n_e^{t,\star} = (p_i^\star - p_j^\star)/d$ contains $p^\star$ on its boundary and satisfies $(n_e^{t,\star})^\top (v_i - v_j) = 0$.*

Theorem 6.19 is a statement about this position model. It is not a theorem about the heterogeneous unrolled program (4.5).

**Lemma 6.20** (Rest input in the unrolled coordinates)**.** *Suppose $u_i = 0$ is admissible and the initial state is a zero-input equilibrium with planar position $p_i^\star$. For an integrator chain, every initial derivative block, including acceleration in the triple-integrator case, is zero. Then $z_i = z_i^0 + S_i u_i$ and $P_i z_i = p_i^\star \mathbf{1}_T + Q_i u_i$ are affine maps. If all local inequalities are inactive at $u_i = 0$ and $Q_i$ is surjective on the dynamics affine hull, the tangent cone contains an input direction whose extracted positions equal $Jp_i^\star$ at every predicted time. For position tracking $\frac{1}{2}\sum_t \|p_i(t) + \alpha p_i^\star\|^2$ and a quadratic input cost with a zero gradient at $u_i = 0$, the position gradient at rest is $(1 + \alpha)p_i^\star$ at each time. The ring contact multipliers from Theorem 6.19, repeated at each time, cancel this gradient before pullback through $Q_i^\top$, so the rest trajectory is KKT for this specified disk-constrained tracking model. This does not prove a negative-curvature direction or trap persistence for general heterogeneous costs and constraints.*

**Theorem 6.21** (Trap at coincident copies)**.** *Consider only the reduced position model of Theorem 6.19, with $P_i = I$, $\mathcal{Z}_i = \mathbb{R}^2$, and $f_i(p_i) = \frac{1}{2}\|p_i + \alpha p_i^\star\|^2$. Initialize all current and delayed copies at $p^\star$ and the local proximal centers at $p^\star$. At an awake endpoint i, the feasible-point local set is the intersection of the adjacent halfspaces with its neighbors fixed at their copied positions. Then:*

1. *With consensus multipliers $\lambda_i^\star = -(1+\alpha)p_i^\star$, the exact local minimizer remains $p_i^\star$. The edge placement* (5.5) *and subsequent averaging preserve $y = p^\star$, and the zero residual preserves the multipliers. This fixed point persists for any bounded ages because every historical copy is the same point. It is not a statement for unequal stale copies or arbitrary heterogeneous local dynamics.*

2. *With $\lambda_i = 0$, a single awake endpoint, and the exact local step without contact cuts, the position is*

$$p_i^+ = \kappa_i p_i^\star, \qquad \kappa_i = \frac{\rho_i + \gamma_i - \alpha}{1 + \rho_i + \gamma_i}.$$

   *For a neighbor fixed at $p_j^\star$, the adjacent disk is violated if and only if*

$$2\cos(2\pi/M) - 1 < \kappa_i < 1,$$

   *that is, $0 < (1 - \kappa_i)R_\star < 2(1 - \cos(2\pi/M))R_\star$. For $M = 8$, the upper bound is $(2 - \sqrt{2})R_\star$. Sufficiently large $\rho_i + \gamma_i$ produces this short step; a small weight can pass the interval and separate the disks again. The copy map* (5.6) *changes $y$ and does not create this violation of the primal positions. For a pair at distance $s \in (0, d)$, its updated copy distance is $s + 2\beta(d - s)$, which remains below $d$ exactly when $\beta < 1/2$. At $s = 0$, the direction is undefined. No descent-based threshold $\beta_*$ enters this geometry.*

3. *On $\{\|p_i - p_j\| \geq d/2\}$, the normalization map has a Lipschitz constant of at most $2\sqrt{2}/d$ in the stacked Euclidean norm. If $\|p^{k-d} - p^\star\| \leq d/(4\sqrt{2})$, then*

$$\|n^{k-d} - n_e^\star\| \leq \tfrac{1}{2}, \qquad (n^{k-d})^\top n_e^\star \geq \tfrac{7}{8}.$$

Item (i) is a fixed point of the halfspace-restricted map when copies coincide and the multipliers are consistent with Theorem 6.19. Item (ii) is a statement about the primal positions, not about $y$. The interval on $\kappa_i$ is strict: not every exact local step violates the disk. Item (iii) is a local sign statement for a stale normal. It is not a robustness theorem on the whole delay ball of radius $\tau$.

*Remark 6.22* (Reading item (ii) of Theorem 6.21). The disk violation in item (ii) requires $\lambda_i = 0$ and an inward displacement inside the interval above. If $\lambda_i$ already equals the Karush–Kuhn–Tucker multiplier of the contact, the first-order term may cancel and the infeasible-point step need not move. Item (ii) does not say that every infeasible copy step leaves the polygon.

**Proposition 6.23** (Trap geometry and Lyapunov descent)**.** *Theorems 6.19 and 6.21, along with Lemma 6.20, use planar geometry and* (5.5), (6.6) *only; they do not invoke $V^k$,* (6.1)*, or Theorem 6.6. Convergence to a saddle of the convex halfspace surrogate is compatible with a strict saddle of the original disk problem. Disk violations in the edge-wise infeasible map are outside the iteration of Theorem 6.6.*

The unconstrained saddle-avoidance results [22, 23] concern stationary points of an unconstrained objective. Here $\nabla f(p^\star) \neq 0$ and the point is stationary only after contact multipliers are included. The frozen polyhedron excludes the finite curved rotation while retaining its first-order tangent field. Theorem 6.21(iii) adds a neighborhood of radius $d/(4\sqrt{2})$ in which a stale normal keeps the same sign. Outside that neighborhood, the sign is not controlled.

## 7. Numerical tests

The experiments address three aspects of the analysis: convergence of a fixed-halfspace inner iteration, stationarity for the original nonlinear disks after normal refresh, and the reduced contact geometry. The original edge-placement and averaging runs are retained in the reproducibility archive, but their tables and plots are removed from the revised evaluation because their local implementation did not solve the stated QPs. All comparisons below use the corrected affine position maps and original local constraints.

### *7.1. A common original-disk KKT screen*

The crossing instance has eight agents, horizon $T = 10$, and sampling period 0.3: three double integrators, three single integrators with corridors, and two triple integrators. Equality elimination gives 160 independent input variables. There are 320 input inequalities, 60 corridor inequalities, and 280 pair/time disk inequalities. Problem-data seed 0 is fixed for all methods. The five starts are five initializations of this one instance, not five different problem instances. Start 0 has zero inputs; starts 1–4 use deterministic small input perturbations, backtracked to strict feasibility. Every algorithm receives the identical input vector for a given start. No solution from another algorithm or from the earlier normal-refresh experiment initializes these runs.

Write all 660 original inequalities as $c_j(u) \geq 0$, including $c_a(u) = \|\Delta_a(u)\|^2 - d_a^2$ for the disks, and let $J_c$ be their Jacobian. For a nonnegative multiplier certificate $\xi$, the stationarity and complementarity residuals are

$$S(u,\xi) = \|\nabla F(u) - J_c(u)^\top \xi\|_\infty, \qquad C(u,\xi) = \|\xi \odot c(u)\|_\infty. \tag{7.1}$$

At each recorded iterate, the same independent evaluator solves

$$\min_{\xi \geq 0,\, t \geq 0} t \quad \text{s.t.} \quad S(u,\xi) \leq 10^{-3} t, \qquad C(u,\xi) \leq 10^{-4} t. \tag{7.2}$$

Fixing $u$ makes the certificate problem linear. It uses every inequality, without a method-specific active-set cutoff. Fixed row and gradient scales are used to condition the linear program (LP); the multipliers are then transformed back and both residuals in (7.1) are recomputed from the unscaled model.

Let $F_{\text{viol}}(u)$ be the maximum of the input-bound violation, corridor violation, and Euclidean disk penetration, in the model's stated units. The common success screen requires, simultaneously,

$$F_{\text{viol}} \leq 10^{-4}, \qquad S \leq 10^{-3}, \qquad C \leq 10^{-4}, \qquad \xi \geq 0. \tag{7.3}$$

The LP supplies the same best joint certificate criterion for every method; the solver's native success flag is recorded separately. Position-copy agreement is not substituted for stationarity. The screen is a first-order certificate and does not establish local minimality or global optimality.

### *7.2. Methods and reproducibility*

The proposed run is the feasible normal-refresh procedure (6.8), with $\kappa = 0.25$. Each convex subproblem is actually solved by the local/copy multiplier iteration, using fixed $\rho_i = 100$, inner proximal weights $\gamma_i = 0.25$, and $\eta = 0$. Normals remain fixed during each inner solve. The requested inner primal

and local-stationarity tolerance is $10^{-4}/(r+1)^2$. We use the Operator Splitting Quadratic Program (OSQP) solver for the local and joint copy QPs, with absolute and relative tolerances of $10^{-9}$. The outer budget is 160, the per-subproblem inner budget is 6000, and the optimizer-time cap is 300 seconds. Every reported successful run met each requested inner tolerance. These are finite-accuracy numerical solves; the exact-minimization hypotheses of Theorems 6.6 and 6.17 are not claimed as exact arithmetic properties of floating-point output.

The external baselines are the released SciPy implementations of SLSQP and `trust-constr` [37–39]. Both solve the original nonlinear disk program. We supply the exact objective gradient and disk Jacobian, and also the objective and weighted constraint Hessians to `trust-constr`. SLSQP uses native `ftol` $= 10^{-11}$; `trust-constr` uses `gtol` $= 10^{-10}$, `xtol` $= 10^{-12}$, and barrier tolerance $10^{-11}$. Both have a 500-major-iteration cap. Their callbacks terminate at the first recorded iterate satisfying (7.3). Thus a native termination flag alone cannot declare a successful comparison.

An additional independent Python reference implements the full-step Algorithm 2 of ALADIN [26], with its exact source mapping documented alongside the code. It has eight local input blocks and 280 four-dimensional pair/time contact blocks. The local contact nonlinear programs (NLPs) retain the nonlinear disk inequality and are solved by an analytic exterior-ball projection. A centralized soft-consensus QP coordinates regularized Lagrangian Hessians and active-constraint Jacobian equalities. The fixed settings are $\rho = 50$, initial coordination penalty 1000, Hessian floor 10, a 200-coordination-step cap, and a 300-second time cap. Every primal and dual step length is one. This explicitly permitted full-step variant has no global convergence guarantee from arbitrary starts. Its core signs and updates were checked against a closed-form convex first step and two known two-agent solutions. The independent code is not the official MATLAB ALADIN toolbox [40]. A separate numerical globalization attempt failed its sanity checks and is excluded from the comparison evidence. No result is attributed to that attempt.

The computations use Python 3.12, NumPy, SciPy, OSQP, and HiGHS on the same machine, with the Basic Linear Algebra Subprograms (BLAS) thread count set to one. Solver time and the time spent verifying the common KKT certificate are reported separately. Native outer iterations, trust-region callbacks, local QPs, and coordination QPs have different costs, so their counts are not converted into an iteration-based speedup. These are serial software measurements, not network or onboard-runtime experiments. Concurrent machine load was not controlled, so observed times are descriptive work logs rather than a controlled speed ranking.

*7.3. Results at the shared stopping criterion*

**Table 2.** Original-disk KKT comparison on five identical shared starts. $J$ is the mean ± sample standard deviation at the first common pass. $F_{\max}$, $S_{\max}$, and $C_{\max}$ are worst final residuals over the five runs. † marks failure of the common screen: the ALADIN objective is left blank because its infeasible endpoint costs are not pooled with accepted solutions.

| Method | Pass | $J$ (mean ± SD) | $F_{\max}$ | $S_{\max}$ | $C_{\max}$ |
|---|---|---|---|---|---|
| Proximal normal refresh | 5/5 | 364.54 ± 36.58 | $5.1 \times 10^{-10}$ | $9.1 \times 10^{-4}$ | $9.1 \times 10^{-5}$ |
| SLSQP | 5/5 | 332.65 ± 23.98 | $5.0 \times 10^{-6}$ | $9.4 \times 10^{-4}$ | $9.4 \times 10^{-5}$ |
| trust-constr | 5/5 | 354.19 ± 26.88 | 0 | $9.4 \times 10^{-4}$ | $9.4 \times 10^{-5}$ |
| ALADIN (custom) | 0/5 | –† | $4.6 \times 10^{0}$ | $5.5 \times 10^{1}$ | $5.5 \times 10^{0}$ |

**Table 3.** Observed work and serial time for the same runs. Steps are method-native and are not equal units of work. Solve time excludes the independently measured common KKT evaluation time. Work counts and per-start results are supplied in the reproducibility files.

| Method | Steps | Solve (s) | KKT (s) | Local QPs | Copy/coord. QPs |
|---|---|---|---|---|---|
| Proximal normal refresh | 29.8 ± 6.1 | 5.489 | 1.420 | 56726.4 | 7090.8 |
| SLSQP | 33.0 ± 4.3 | 0.392 | 1.462 | – | – |
| trust-constr | 102.6 ± 15.1 | 1.267 | 5.004 | – | – |
| ALADIN (custom) | 200.0 ± 0.0 | 1.627 | 8.067 | 1600.0 | 200.0 |

The proximal normal-refresh method, SLSQP, and `trust-constr` each satisfy all four conditions in (7.3) on all five starts. Their mean objectives are 364.54, 332.65, and 354.19, respectively. The comparison therefore does not establish an objective or runtime advantage for the proposed method. The different endpoints reflect the nonconvex contact problem and sensitivity to the path from a common initialization. In particular, the smallest objective is not declared a global optimum.

The proposed method takes 24, 23, 34, 31, and 37 normal-refresh updates. All 149 recorded outer objective changes are decreases. The finite-run strict-point checks for Assumption 6.16 maximize the common contact-halfspace slack over the actual local set at each saved anchor. All 149 LPs have a positive realized margin; the minimum is 0.3403457. Their witness vectors and local feasibility errors are saved. This verifies the condition along the recorded finite trajectories; it is not a proof of a uniform bound for an infinite extension.

The validated full-step ALADIN reference does not reach the common screen within its stated budget. Although its local contact NLPs are solved feasibly, its final coordinated input vectors violate original constraints and fail stationarity and complementarity. These infeasible endpoint costs are not compared with KKT-qualified objectives. All five trajectories, parameters, failures, and independent checks are retained. This outcome concerns the disclosed independent full-step implementation, not the capabilities of other ALADIN variants or the official toolbox.

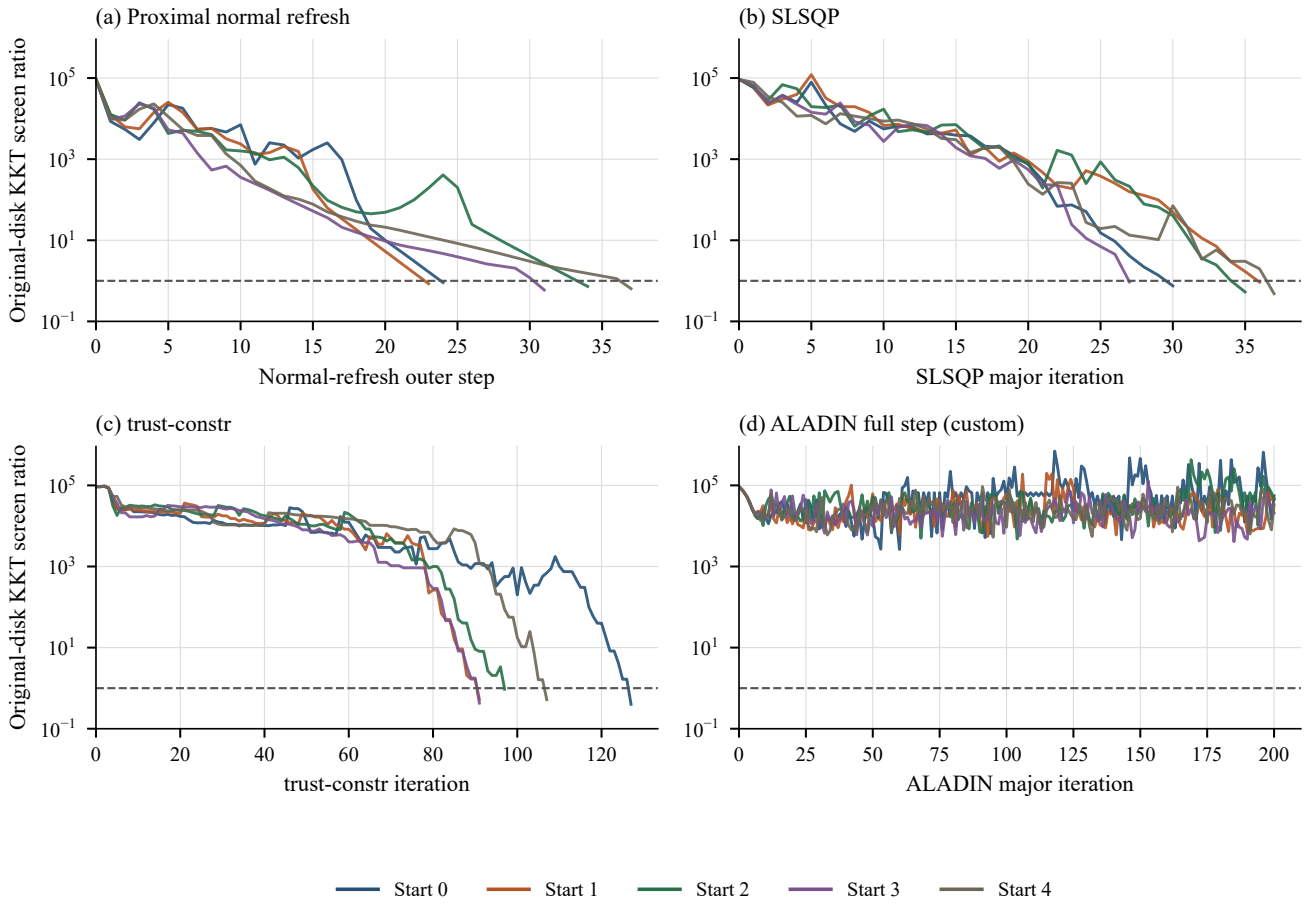


**Figure 2.** Common original-disk KKT screen along the saved trajectories. The displayed ratio combines physical feasibility, raw stationarity, and raw complementarity using the thresholds in (7.3); success requires the ratio at most one. Each panel uses that method's native iteration axis. The axes are not a common measure of computational work.

### *7.4. Joint-copy accuracy and fixed-halfspace distance*

The neighbor solver in Section 6.1 is compared with a high-accuracy OSQP solution on three weighted projection cases: zero centers, perturbed centers, and the final frozen geometry. The cyclic solver uses 39, 98, and 19 sweeps. The largest primal-solution difference from OSQP is $1.19 \times 10^{-10}$, the largest halfspace violation is $8.55 \times 10^{-11}$, and the largest complementarity product is $1.60 \times 10^{-10}$. The primal recovery invariant agrees to $1.78 \times 10^{-15}$. The code includes endpoint updates, retained edge multipliers, and replicated neighbor max-consensus for stopping. It does not average independently corrected edge copies. An end-to-end start-0 run using this neighbor copy solver also passes the original-disk KKT screen after 24 outer updates, at $J = 324.548277865$. Its objective differs from the joint-OSQP implementation by $2.80 \times 10^{-9}$. It uses 6,647 copy projections and 102,723 cyclic sweeps; the serial simulation takes 116.6 seconds excluding certificate evaluation, so this agreement is an implementation check rather than a speedup claim.

For a separate check of Theorem 6.6, we retain the fixed-halfspace synchronous experiment. Twelve normal refreshes are followed by another convex solve at the final normals, so the reference saddle and the entire measured distance use the same frozen problem. Its reference objective is 290.824066, with minimum normal cosine 0.999999989. With $\rho_i = 4\lambda_{\max}(H_i)$, $\gamma_i = 2\lambda_{\max}(H_i)$, and the input-coordinate proximal norm, the iteration reaches penetration $7.722 \times 10^{-4}$ and consensus residual $6.776 \times 10^{-3}$ at step 69, at objective 292.806. All 69 saddle-distance comparisons, including the initial state, decrease; the augmented Lagrangian increases on 16 steps. This figure tests the fixed-halfspace distance theorem. It has a different initializer and stopping screen from Table 2 and is not mixed into the original-disk performance comparison. Near alignment alone is not an exact disk KKT certificate.

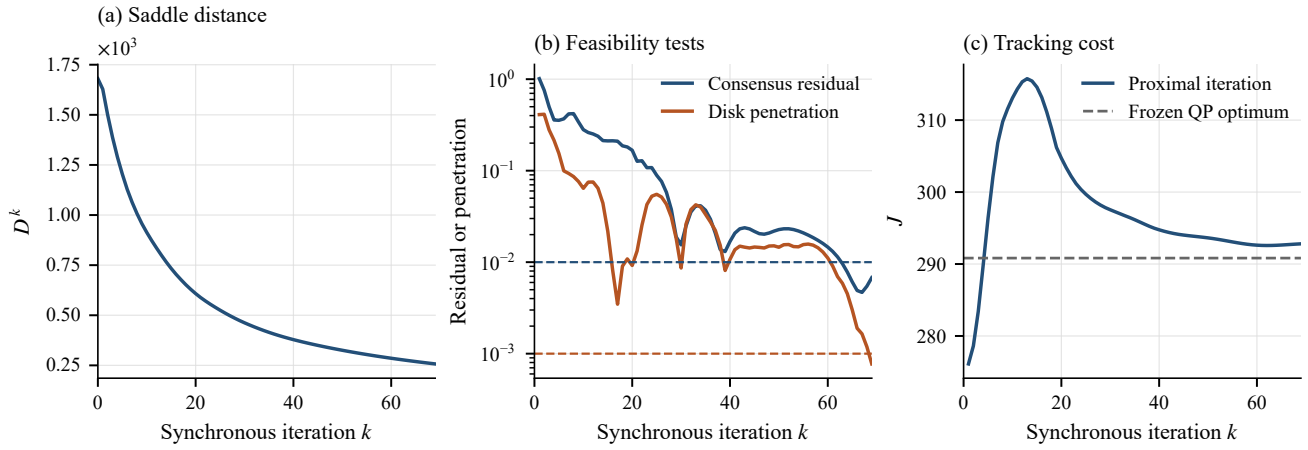


**Figure 3.** Fixed-halfspace synchronous proximal run, seed 0. (a) Distance to the saddle of that same frozen problem, including $D^0$. (b) Consensus residual and disk penetration. (c) Tracking cost and the frozen-QP reference value. This is a distance-inequality diagnostic, separate from the common original-disk KKT benchmark.

*7.5. Reduced contact example and data access*

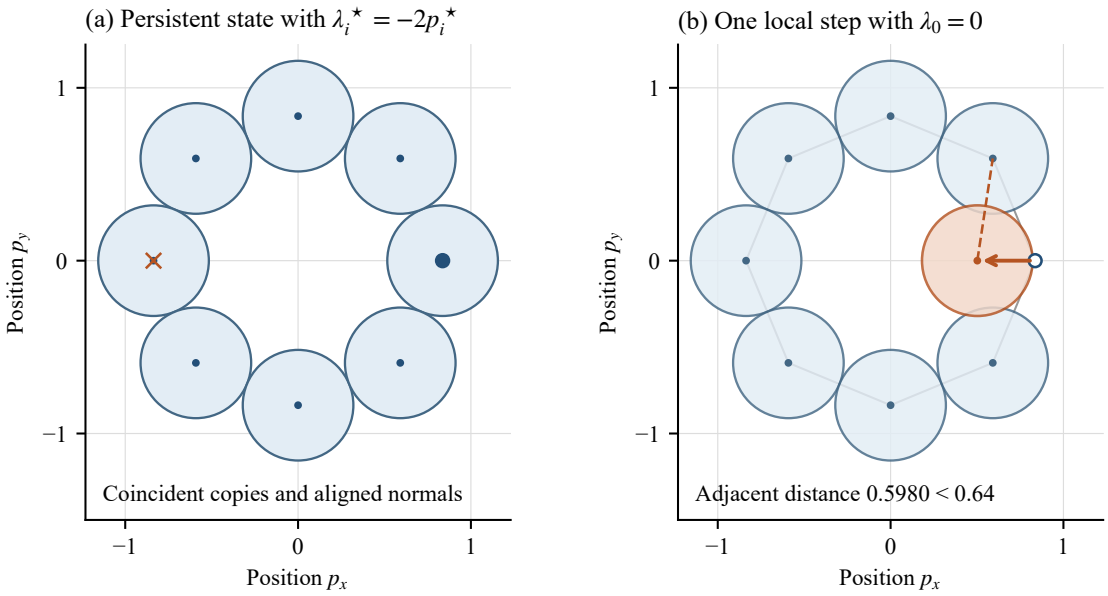


**Figure 4.** Exact reduced-position illustration of Theorem 6.21, with $M = 8$, $d = 0.64$, and $\alpha = 1$. (a) Coincident history and $\lambda_i^\star = -2p_i^\star$ give a fixed point. (b) One awake endpoint with $\lambda_i = 0$ and $\rho_i + \gamma_i = 4$ has a radial factor $\kappa_i = 0.6$ and an adjacent distance of $0.5980 < d$. The panels use distinct multiplier states.

Here $R_\star = 0.8362003$, the inward displacement is 0.3344801, and the adjacent penetration is 0.0419725. These follow from the exact local minimizer, without an approximate heterogeneous QP. Historical copies in panel (a) are identical, so a bounded age reads the same point. This is not a persistence result for arbitrary unequal stale copies.

The package `experiments/benchmark/` supplies the problem construction, analytic derivatives, common certificate LP, all solver settings, all five starts, and regeneration scripts. Saved states and nonnegative multiplier certificates allow independent recomputation of the table. Model checks compare the condensed objective with the unrolled implementation and check directional objective, Jacobian, and Hessian derivatives. The original logs are archived separately and are not used as evidence for the corrected algorithms.

## 8. Discussion

There are three distinct stationarity mechanisms. Exact lifting preserves the nonlinear disk program, a fixed-halfspace inner solve converges to a saddle of that convex restriction, and feasible normal refresh yields original-disk KKT accumulation under the explicit strict-point condition of Assumption 6.16. These statements cannot be interchanged. The contact multiplier bound for the outer method comes from strict contact slack and compact local sets, while the inner consensus-multiplier bound comes from its saddle distance.

The neighbor copy solver retains a dual variable for each contact. It therefore solves the weighted intersection problem rather than averaging independent pair placements. Its convergence proof

permits redundant contact rows. Simulated serialized edge visits and max-consensus implement the stated information pattern, but they do not establish process-time acceleration or performance on a physical network.

The shared original-disk KKT screen separates numerical feasibility from first-order stationarity. All five proximal normal-refresh runs and both external SciPy baselines pass it, but their objectives and observed work differ. No uniform objective or runtime advantage is claimed. The additional ALADIN reference is identified by implementation and variant, and unsuccessful endpoints are not treated as KKT-qualified solutions.

The reduced jam remains informative even with a genuine KKT screen: a strict saddle can satisfy that screen. In Theorem 6.21, coincident history and consistent multipliers give a fixed point, while a zero-multiplier single-endpoint step can enter a neighboring disk. This is a statement about specified contact states, not arbitrary heterogeneous constrained dynamics or every stale-copy trajectory.

## 9. Limitations

The exact inner theorem assumes current data, fixed halfspaces, exact minimization, and a saddle. The feasible normal-refresh theorem requires a disk-feasible start and uniform strict points for the convex contact restrictions. Positive slack on all saved finite anchors is a numerical check, not a proof that this assumption holds on every infinite trajectory. Finite QP tolerances and the original-disk KKT thresholds are disclosed; they are not exact-arithmetic optimality certificates.

The comparison uses five starts of one eight-agent, ten-step instance. It does not measure scaling to large fleets, communication delays, onboard computation, or closed-loop MPC. The independent ALADIN reference is not the official MATLAB toolbox and is not a substitute for a full implementation study across ALADIN variants. Native iteration counts are not common oracle-work units, and recorded serial times do not prove distributed speedups.

No convergence theorem is asserted for the original arbitrary stale edge-placement/averaging map. The paper also does not establish global optimality, strict-saddle avoidance, linear rates, nonconvex oracle-complexity bounds, unbounded-delay robustness, fully nonlinear local dynamics, or recursive feasibility in a closed loop.

## 10. Conclusion

The heterogeneous disk program admits an exact consensus lifting, while fixed contact halfspaces form an inner approximation. A specified synchronous local/copy multiplier iteration has a nonincreasing saddle distance and bounded multipliers. Hildreth's retained-edge procedure provides a neighbor implementation of its joint weighted copy projection. Under uniform strict points, feasible normal refresh has KKT cluster points for the original disk program, including active local bounds.

The reduced antipodal polygon is a KKT strict saddle, and the corrected trap statements specify its history, multipliers, and displacement interval. Five shared-start comparisons now use a single original-disk feasibility, stationarity, and complementarity criterion for the proposed implementation and external solvers. They establish a reproducible comparison at common accuracy, without implying a general performance advantage or convergence of the withdrawn stale-edge heuristic.

## Author contributions

Zhonghui Xue: Conceptualization, formal analysis, writing—original draft. Yu Chen: Software, investigation, writing—review and editing.

## Use of Generative-AI tools declaration

During the preparation and revision of this article, the authors used two OpenAI tools (ChatGPT and Codex) to assist with language editing, drafting and revising mathematical arguments, developing and debugging numerical code, computational experiments and checks, bibliographic checks, and manuscript formatting. This assistance concerns the text and proofs throughout the manuscript, the numerical study in Section 7 and its accompanying code, and the bibliography. The authors reviewed and edited the resulting material and take full responsibility for the content of the article.

## Acknowledgments

This work was supported by the Shanghai Educational Science Research Project (Grant No. C2025180).

## Conflict of interest

The authors declare that they have no conflicts of interest.

## Appendix

The two delayed-contact maps and the detailed proofs are provided in the accompanying appendix (Appendices A and B).

## References

1. Z. Pan, K. Wu, BC-ADMM: A parallel decoupled non-convex constrained optimizer for robot applications, *Int. J. Rob. Res.*, 2026, Advance online publication. https://doi.org/10.1177/02783649261458408
2. D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. M. Scokaert, Constrained model predictive control: Stability and optimality, *Automatica*, **36** (2000), 789–814. https://doi.org/10.1016/s0005-1098(99)00214-9
3. I. Karamouzas, N. Sohre, R. Narain, S. J. Guy, Implicit crowds: Optimization integrator for robust crowd simulation, *ACM Trans. Graph.*, **36** (2017), 136. https://doi.org/10.1145/3072959.3073705
4. B. Zhou, F. Gao, L. Wang, C. Liu, S. Shen, Robust and efficient quadrotor trajectory generation for fast autonomous flight, *IEEE Robot. Autom. Lett.*, **4** (2019), 3529–3536. https://doi.org/10.1109/lra.2019.2927938
5. S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends Mach. Learn.*, **3** (2011), 1–122. https://doi.org/10.1561/2200000016

6. R. T. Rockafellar, Augmented Lagrangians and applications of the proximal point algorithm in convex programming, *Math. Oper. Res.*, **1** (1976), 97–116. https://doi.org/10.1287/moor.1.2.97
7. Y. Yang, X. Guan, Q.-S. Jia, L. Yu, B. Xu, C. J. Spanos, A survey of ADMM variants for distributed optimization: Problems, algorithms and features, arXiv:2208.03700, 2022. https://doi.org/10.48550/arXiv.2208.03700
8. O. Shorinwa, T. Halsted, J. Yu, M. Schwager, Distributed optimization methods for multi-robot systems: Part 2—a survey, *IEEE Robot. Autom. Mag.*, **31** (2024), 154–169. https://doi.org/10.1109/mra.2024.3352852
9. M. Hong, Z.-Q. Luo, M. Razaviyayn, Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems, *SIAM J. Optim.*, **26** (2016), 337–364. https://doi.org/10.1137/140990309
10. Y. Wang, W. Yin, J. Zeng, Global convergence of ADMM in nonconvex nonsmooth optimization, *J. Sci. Comput.*, **78** (2019), 29–63. https://doi.org/10.1007/s10915-018-0757-z
11. B. Jiang, T. Lin, S. Ma, S. Zhang, Structured nonconvex and nonsmooth optimization: Algorithms and iteration complexity analysis, *Comput. Optim. Appl.*, **72** (2019), 115–157. https://doi.org/10.1007/s10589-018-0034-y
12. A. Themelis, P. Patrinos, Douglas–Rachford splitting and ADMM for nonconvex optimization: Tight convergence results, *SIAM J. Optim.*, **30** (2020), 149–181. https://doi.org/10.1137/18m1163993
13. G. Li, T. K. Pong, Global convergence of splitting methods for nonconvex composite optimization, *SIAM J. Optim.*, **25** (2015), 2434–2460. https://doi.org/10.1137/140998135
14. T.-H. Chang, M. Hong, W.-C. Liao, X. Wang, Asynchronous distributed ADMM for large-scale optimization—part I: Algorithm and convergence analysis, *IEEE Trans. Signal Process.*, **64** (2016), 3118–3130. https://doi.org/10.1109/tsp.2016.2537271
15. M. Hong, A distributed, asynchronous and incremental algorithm for nonconvex optimization: An ADMM approach, *IEEE Trans. Control Netw. Syst.*, **5** (2018), 935–945. https://doi.org/10.1109/tcns.2017.2657460
16. S. Kumar, R. Jain, K. Rajawat, Asynchronous optimization over heterogeneous networks via consensus ADMM, *IEEE Trans. Signal Inf. Process. Netw.*, **3** (2017), 114–129. https://doi.org/10.1109/tsipn.2016.2593896
17. B. Mafakheri, J. H. Manton, I. Shames, An asynchronous decentralised optimisation algorithm for nonconvex problems, arXiv:2507.22311, 2025. https://doi.org/10.48550/arXiv.2507.22311
18. G. Carnevale, N. Bastianello, G. Notarstefano, R. Carli, ADMM-tracking gradient for distributed optimization over asynchronous and unreliable networks, *IEEE Trans. Autom. Control*, **70** (2025), 5160–5175. https://doi.org/10.1109/tac.2025.3539454
19. X. Zhang, J. Ma, Z. Cheng, S. Huang, C. W. de Silva, T. H. Lee, Improved hierarchical ADMM for nonconvex cooperative distributed model predictive control, arXiv:2011.00463, 2020. https://doi.org/10.48550/arXiv.2011.00463
20. K. Sun, X. A. Sun, A two-level distributed algorithm for nonconvex constrained optimization, *Comput. Optim. Appl.*, **84** (2023), 609–649. https://doi.org/10.1007/s10589-022-00433-4
21. J. Nocedal, S. J. Wright, *Numerical Optimization*, 2nd edition, Springer, 2006. Available from: https://doi.org/10.1007/978-0-387-40065-5.

22. J. D. Lee, M. Simchowitz, M. I. Jordan, B. Recht, Gradient descent only converges to minimizers, In: *29th Annual Conference on Learning Theory*, vol. 49 of Proceedings of Machine Learning Research. PMLR, 2016, 1246–1257. Available from: https://proceedings.mlr.press/v49/lee16.html.
23. J. D. Lee, I. Panageas, G. Piliouras, M. Simchowitz, M. I. Jordan, B. Recht, First-order methods almost always avoid strict saddle points, *Math. Program.*, **176** (2019), 311–337. https://doi.org/10.1007/s10107-019-01374-3
24. H. Attouch, J. Bolte, B. F. Svaiter, Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods, *Math. Program.*, **137** (2013), 91–129. https://doi.org/10.1007/s10107-011-0484-9
25. J. Bolte, S. Sabach, M. Teboulle, Proximal alternating linearized minimization for nonconvex and nonsmooth problems, *Math. Program.*, **146** (2014), 459–494. https://doi.org/10.1007/s10107-013-0701-9
26. B. Houska, J. Frasch, M. Diehl, An augmented Lagrangian based algorithm for distributed nonconvex optimization, *SIAM J. Optim.*, **26** (2016), 1101–1127. https://doi.org/10.1137/140975991
27. W. Deng, W. Yin, On the global and linear convergence of the generalized alternating direction method of multipliers, *J. Sci. Comput.*, **66** (2016), 889–916. https://doi.org/10.1007/s10915-015-0048-x
28. D. P. Bertsekas, J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, 2015. Available from: http://www.athenasc.com/pdcbook.html.
29. J. Guo, G. Hug, O. Tonguz, Asynchronous ADMM for distributed non-convex optimization in power systems, arXiv:1710.08938, 2017. https://doi.org/10.48550/arXiv.1710.08938
30. M. Mesbahi, M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*, Princeton University Press, 2010. https://doi.org/10.1515/9781400835355
31. A. Nedić, J. Liu, Distributed optimization for control, *Annu. Rev. Control Robot. Auton. Syst.*, **1** (2018), 77–103. https://doi.org/10.1146/annurev-control-060117-105131
32. C. Hildreth, A quadratic programming procedure, *Naval Res. Logist. Q.*, **4** (1957), 79–85. https://doi.org/10.1002/nav.3800040113
33. R. L. Dykstra, An algorithm for restricted least squares regression, *J. Am. Stat. Assoc.*, **78** (1983), 837–842. https://doi.org/10.1080/01621459.1983.10477029
34. R. J. Tibshirani, Dykstra's algorithm, ADMM, and coordinate descent: Connections, insights, and extensions, arXiv:1705.04768, 2017. https://doi.org/10.48550/arXiv.1705.04768
35. B. R. Marks, G. P. Wright, Technical note—a general inner approximation algorithm for nonconvex mathematical programs, *Oper. Res.*, **26** (1978), 681–683. https://doi.org/10.1287/opre.26.4.681
36. G. Scutari, F. Facchinei, L. Lampariello, Parallel and distributed methods for constrained nonconvex optimization—part I: Theory, *IEEE Trans. Signal Process.*, **65** (2017), 1929–1944. https://doi.org/10.1109/tsp.2016.2637317
37. D. Kraft, *A Software Package for Sequential Quadratic Programming*, Technical Report DFVLR-FB 88-28, DFVLR German Aerospace Center, 1988.
38. R. H. Byrd, M. E. Hribar, J. Nocedal, An interior point algorithm for large-scale nonlinear programming, *SIAM J. Optim.*, **9** (1999), 877–900. https://doi.org/10.1137/s1052623497325107

39. P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, et al., SciPy 1.0: Fundamental algorithms for scientific computing in Python, *Nat. Methods*, **17** (2020), 261–272. https://doi.org/10.1038/s41592-019-0686-2

40. A. Engelmann, Y. Jiang, H. Benner, R. Ou, B. Houska, T. Faulwasser, ALADIN-$\alpha$—an open-source MATLAB toolbox for distributed non-convex optimization, *Optim. Control Appl. Methods*, **43** (2022), 4–22. https://doi.org/10.1002/oca.2811




AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0)