

Collision-Free Velocity Scheduling for Multi-Agent Systems on Predefined Routes via Inexact-Projection ADMM

Seungyeop Lee[✉] and Jong-Han Kim[✉], *Member, IEEE*

Abstract—In fixed-route multi-agent systems, such as urban air mobility and warehouse logistics, collision avoidance relies on temporal scheduling rather than spatial rerouting. Existing fixed-route coordination methods often rely on priority assignment, explicit sequencing, or mixed-integer order selection. However, selecting an effective passing-order can be difficult, especially when additional waypoint-level timing requirements such as dwell times or fixed-terminal times are present. This paper presents a priority-free temporal scheduling framework for fixed-route multi-agent coordination. The framework optimizes waypoint passage times to minimize the sum of terminal arrival times while satisfying speed limits, timing specifications, and collision avoidance constraints. To handle collision avoidance, a smooth flyby surrogate maps discrete passage times into continuous spatial trajectories. The resulting nonlinear and nonconvex scheduling problem is solved using an inexact-projection ADMM scheme without introducing explicit passing-order variables. Numerical comparisons with order-based and reactive baselines, together with additional stress tests, are presented to evaluate the performance and limitations of the proposed approach.

Index Terms—Collision avoidance, multi-robot systems, planning, scheduling and coordination, optimization and optimal control

I. INTRODUCTION

COLLISION avoidance is a fundamental requirement in multi-agent systems and has been widely studied across various motion planning and control frameworks [1]–[8]. However, existing methods primarily resolve conflicts by modifying spatial trajectories. In domains such as urban air mobility and warehouse logistics, agents typically operate on fixed-routes where lateral maneuvers are restricted. Consequently, spatial avoidance strategies are inherently limited in these systems, motivating approaches that resolve conflicts by adjusting timing of agents along their predefined paths.

Fixed-route conflicts are conventionally addressed through priority assignment, conflict-based search, or discrete sequencing variables. Priority and safe-interval planning algorithms

sequence agents incrementally, treating previously planned trajectories as dynamic obstacles [9], [10]. Mixed-integer and sequential convex programming approaches formulate passing interactions via binary or logical constraints before optimizing timing or speed profiles [11]–[14]. Recent fixed-route timing and cooperative speed planning strategies also adjust speed profiles along prescribed paths, frequently relying on priority queues or explicit ordering heuristics [15], [16]. While these frameworks perform well when an effective passing-order can be identified efficiently, finding such an order generally requires combinatorial search, and the resulting performance can depend strongly on the selected order. Furthermore, integrating additional temporal constraints—such as dwell times or fixed-terminal times—substantially expands the discrete decision space [17]–[19].

Motivated by these limitations, we propose a priority-free temporal optimization framework that directly resolves fixed-route conflicts without relying on discrete sequencing variables. By formulating multi-agent coordination strictly as a waypoint passage time optimization problem, the spatial route geometry is preserved. Under this formulation, speed limits map to inter-waypoint duration bounds, and specific temporal constraints are explicitly incorporated. To enforce pairwise collision avoidance constraints over the mission horizon, the framework employs a differentiable continuous-time trajectory surrogate. The surrogate maps waypoint passage times to continuous position trajectories, thereby enabling first-order optimization of the sampled safety constraints. This surrogate maps waypoint passage times to continuous spatial trajectories, which enables the integration of spatial collision avoidance constraints into the optimization problem and facilitates the application of gradient-based optimization techniques.

The primary contributions of this work are summarized as follows:

- 1) We formulate multi-agent coordination on fixed-routes as a waypoint passage time optimization problem. The formulation uses only timing variables and directly incorporates speed limits, dwell times, and optional terminal time constraints.
- 2) We introduce a differentiable flyby surrogate for fixed-route motion. The surrogate maps waypoint passage times to continuous position trajectories, enabling first-order, distance-based collision avoidance corrections in the timing variables.
- 3) We develop a residual-controlled inexact alternating direction method of multipliers (ADMM) solver with an interval closest-approach safety penalty and evaluate it against order-based and reactive baselines.

Received 21 March 2026; revised 14 June 2026; accepted 13 July 2026. Date of publication 00 July 2026; date of current version 00 July 2026. This article was recommended for publication by Editor Chao-Bo Yan upon evaluation of the associate editor and reviewers' comments. This work was supported in part by the Space Challenge Program funded by the Korea Aerospace Administration (KASA) under Grant No. RS-2025-16063807 and in part by the Theater Defense Research Center funded by the Defense Acquisition Program Administration (DAPA) under Grant No. UD240002SD. (*Corresponding author: Jong-Han Kim*).

The authors are with the Department of Aerospace Engineering and the Program in Aerospace Systems Convergence, Inha University, Incheon, South Korea seungyeoplee@inha.edu; jhank@inha.ac.kr

Jong-Han Kim is also with the Aerospace Systems Research Institute, Inha University, Incheon, South Korea

Source code and interactive video demonstrations provided by the authors are available at <https://github.com/sy010119/MACSP0>

Digital Object Identifier 10.1109/LRA.2026.0000000

We evaluate the proposed framework against representative fixed-route baselines on standard multi-agent benchmarks, timing-constrained cases, and random-crossing stress tests.

II. PROBLEM STATEMENT

A. Notation

Throughout this paper, scalars are denoted by nonbold symbols, vectors by lowercase bold symbols, and matrices by uppercase bold symbols. The set of integers $\{1, 2, \dots, N\}$ is denoted by $[N]$. For $\mathbf{x} \in \mathbb{R}^N$, the difference operator Δ is defined as $\Delta \mathbf{x} := \mathbf{x}_{2:N} - \mathbf{x}_{1:N-1}$, where $\mathbf{x}_{i:j}$ denotes the subvector from the i -th to the j -th element. Vertical concatenation is denoted by $[\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_N] := [\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_N^T]^T$. For a scalar a , let $[a]_+ := \max(a, 0)$. The indicator function of a set $\mathcal{C}$ is denoted by $I_{\mathcal{C}}(\mathbf{x})$, where $I_{\mathcal{C}}(\mathbf{x}) = 0$ if $\mathbf{x} \in \mathcal{C}$ and $I_{\mathcal{C}}(\mathbf{x}) = +\infty$ otherwise.

Table I summarizes the main variables and parameters.

B. Assumptions

We consider a system of K agents. Each agent i is assigned a predefined waypoint sequence $\{\mathbf{p}_n^{(i)}\}_{n=1}^{N_i}$ and follows the corresponding piecewise-linear path using flyby maneuvers.

Each agent is assumed to move at a constant speed along each segment. Under this assumption, the velocity on segment n is parameterized by the waypoint passage times as follows:

$$\mathbf{v}_n^{(i)}(\mathbf{t}^{(i)}) = \frac{\mathbf{p}_{n+1}^{(i)} - \mathbf{p}_n^{(i)}}{\Delta t_n^{(i)}}, \quad n = 1, \dots, N_i - 1.$$

Fig. 1 illustrates the resulting waypoint path representation.

C. Problem Formulation

We formulate a fixed-route velocity-scheduling problem in which the waypoint passage times $\{\mathbf{t}^{(i)}\}_{i=1}^K$ are the decision

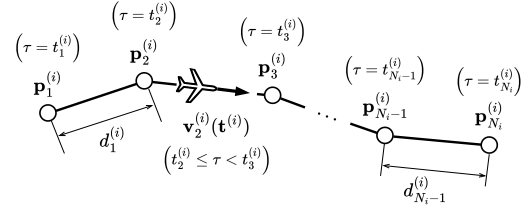


Fig. 1. Illustration of a piecewise-linear waypoint path and its corresponding segment notation.

variables. The objective is to minimize the sum of terminal arrival times of agents with free terminal times, subject to boundary-time constraints, speed bounds, and pairwise safety constraints.

Each agent departs at a prescribed time $t_s^{(i)}$. Terminal constraints apply only to agents in $\mathcal{K}_f$, whereas agents in $\mathcal{K}_u$ contribute to the arrival time objective. Under the constant speed assumption, speed limits are expressed as box constraints on the segment durations. Finally, collision avoidance is imposed on the continuous spatial trajectories for every agent pair in $\mathcal{P}$ at every temporal-grid point in $\mathcal{T}_d$.

These conditions give

$$\begin{aligned} & \text{minimize} && \sum_{i \in \mathcal{K}_u} t_{N_i}^{(i)} \\ & \text{subject to} && t_1^{(i)} = t_s^{(i)}, \quad \forall i \in [K], \\ & && t_{N_i}^{(i)} = t_f^{(i)}, \quad \forall i \in \mathcal{K}_f, \\ & && \frac{\mathbf{d}^{(i)}}{v_{\max}} \leq \Delta \mathbf{t}^{(i)} \leq \frac{\mathbf{d}^{(i)}}{v_{\min}}, \quad \forall i \in [K], \\ & && \|\mathbf{p}^{(i)}(\mathbf{t}^{(i)}, \tau) - \mathbf{p}^{(j)}(\mathbf{t}^{(j)}, \tau)\|_2 \geq d_{\text{safe}}, \\ & && \forall (i, j) \in \mathcal{P}, \quad \forall \tau \in \mathcal{T}_d. \end{aligned} \quad (1)$$

Here, $\Delta \mathbf{t}^{(i)} := [\Delta t_1^{(i)}, \dots, \Delta t_{N_i-1}^{(i)}]^T$ and $\mathbf{d}^{(i)} := [d_1^{(i)}, \dots, d_{N_i-1}^{(i)}]^T$ denote the segment duration and length vectors, respectively. The vector $\mathbf{p}^{(i)}(\mathbf{t}^{(i)}, \tau)$ denotes the position of agent i at time τ induced by its waypoint sequence and passage times $\mathbf{t}^{(i)}$.

III. SURROGATE TRAJECTORY MODEL

A piecewise-linear trajectory exhibits instantaneous velocity changes at waypoints, producing a nonsmooth inter-agent distance function. We therefore replace abrupt segment switching with a smooth surrogate.

Given the prescribed waypoints and passage times, the surrogate velocity of agent i is

$$\tilde{\mathbf{v}}^{(i)}(\mathbf{t}^{(i)}, \tau) = \sum_{n=1}^{N_i-1} \mathbf{v}_n^{(i)}(\mathbf{t}^{(i)}) \left[\sigma(\beta(\tau - \hat{t}_n^{(i)})) - \sigma(\beta(\tau - \hat{t}_{n+1}^{(i)})) \right], \quad (2)$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function, $\hat{t}_n^{(i)} := t_n^{(i)} + 2 \ln(2)/\beta$ is the shifted transition center, and β is the transition sharpness parameter. The constant offset centers the transition in time.

TABLE I
NOTATION

Variable	Description
K	Number of agents
N_i	Number of waypoints assigned to agent i
N	Number of timing variables, $N = \sum_{i=1}^K N_i$
$\mathbf{p}_n^{(i)}$	n -th spatial waypoint for agent i
$t_n^{(i)}$	Passage time of agent i at $\mathbf{p}_n^{(i)}$
$\mathbf{t}^{(i)}$	Passage time vector of agent i , $[t_1^{(i)}, \dots, t_{N_i}^{(i)}]^T$
$\Delta t_n^{(i)}$	Duration of segment n for agent i
$d_n^{(i)}$	Length of segment n for agent i
$\mathbf{v}_n^{(i)}$	Nominal velocity of agent i on segment n
$v_{\min}, v_{\max}$	Prescribed speed bounds
$t_s^{(i)}$	Prescribed departure time of agent i
$t_f^{(i)}$	Prescribed terminal time of agent i , if imposed
$\mathcal{K}_f$	Set of agents with fixed-terminal times
$\mathcal{K}_u$	Set of agents with free terminal times, $[K] \setminus \mathcal{K}_f$
$\mathcal{P}$	Set of unordered agent pairs
$T_{\max}$	Maximum terminal arrival time, $T_{\max} := \max_{i \in [K]} t_{N_i}^{(i)}$
$\mathcal{T}$	continuous-time domain, $\mathcal{T} := [0, T_{\max}]$
$\mathcal{T}_d$	Uniform discrete temporal grid of $\mathcal{T}$
M	Number of discrete time nodes, $M := \mathcal{T}_d $

Integrating (2) gives the surrogate position model

$$\tilde{\mathbf{p}}^{(i)}(\mathbf{t}^{(i)}, \tau) = \mathbf{p}_1^{(i)} + \sum_{n=1}^{N_i-1} \frac{\mathbf{v}_n^{(i)}(\mathbf{t}^{(i)})}{\beta} \left[\zeta(\beta(\tau - \hat{t}_n^{(i)})) - \zeta(\beta(\tau - \hat{t}_{n+1}^{(i)})) - b_n^{(i)} \right], \quad (3)$$

where $\zeta(x) := \ln(1 + e^x)$ is the softplus function, and $b_n^{(i)} := \zeta(\beta(t_1^{(i)} - \hat{t}_n^{(i)})) - \zeta(\beta(t_1^{(i)} - \hat{t}_{n+1}^{(i)}))$. The trajectory in (3) is differentiable with respect to τ and $\mathbf{t}^{(i)}$, so the collision constraint in (1) can be handled using first-order information.

In many operational scenarios, agents are required to remain stationary at specific waypoints for designated dwell times. These dwell-time requirements are incorporated by shifting the segment-transition intervals in the surrogate model, rather than by introducing additional ADMM constraints. This parameterization encodes the prescribed dwell durations while preserving the split structure of the optimization problem.

Let $t_{\text{wp},n}^{(i)} \geq 0$ denote the dwell time at waypoint n before departing along segment n , with $t_{\text{wp},1}^{(i)} = 0$, and define

$$\kappa_n^{(i)} := \sum_{\ell=1}^n t_{\text{wp},\ell}^{(i)}, \quad n = 1, \dots, N_i - 1.$$

In the surrogate evaluation, the transition interval for segment n is shifted from $[t_n^{(i)}, t_{n+1}^{(i)}]$ to

$$[t_n^{(i)} + \kappa_n^{(i)}, t_{n+1}^{(i)} + \kappa_n^{(i)}].$$

Equivalently, the two transition centers in the n -th term of (2) and (3) are replaced by

$$\hat{t}_{n,L}^{(i)} = t_n^{(i)} + \kappa_n^{(i)} + \frac{2 \ln 2}{\beta}, \quad \hat{t}_{n,R}^{(i)} = t_{n+1}^{(i)} + \kappa_n^{(i)} + \frac{2 \ln 2}{\beta}.$$

This convention inserts a prescribed dwell interval between consecutive moving segments while retaining the waypoint timing variables $\mathbf{t}^{(i)}$. The executed terminal time of agent i is $t_{N_i}^{(i)} + \kappa_{N_i-1}^{(i)}$. Therefore, a prescribed terminal time $t_f^{(i)}$ is imposed through

$$t_{N_i}^{(i)} + \kappa_{N_i-1}^{(i)} = t_f^{(i)}.$$

The surrogate model (3) generates a flyby path near the waypoints, and the miss distance is controlled by β . To keep the tracking error below a prescribed tolerance ϵ_{wp} , β must be chosen accordingly.

Let $\Delta \mathbf{v}_k^{(i)} := \mathbf{v}_k^{(i)} - \mathbf{v}_{k-1}^{(i)}$ denote the velocity jump at waypoint k , and let $\rho_\beta(s) := \zeta(-\beta|s|)/\beta$ denote the integral tail of the sigmoid transition at time offset s . The spatial deviation at the transition center $\hat{t}_k^{(i)}$ satisfies

$$\|\tilde{\mathbf{p}}^{(i)}(\mathbf{t}^{(i)}, \hat{t}_k^{(i)}) - \mathbf{p}_k^{(i)}\|_2 \leq \sum_{\ell=2}^{N_i-1} \|\Delta \mathbf{v}_\ell^{(i)}\|_2 \rho_\beta(\hat{t}_k^{(i)} - \hat{t}_\ell^{(i)}).$$

The term with $\ell = k$ gives the dominant local miss distance.

For a given timing profile, the following condition enforces the deviation bound ϵ_{wp} :

$$\max_{i \in [K]} \max_{k=2, \dots, N_i-1} \sum_{\ell=2}^{N_i-1} \|\Delta \mathbf{v}_\ell^{(i)}\|_2 \rho_\beta(\hat{t}_k^{(i)} - \hat{t}_\ell^{(i)}) \leq \epsilon_{\text{wp}}.$$

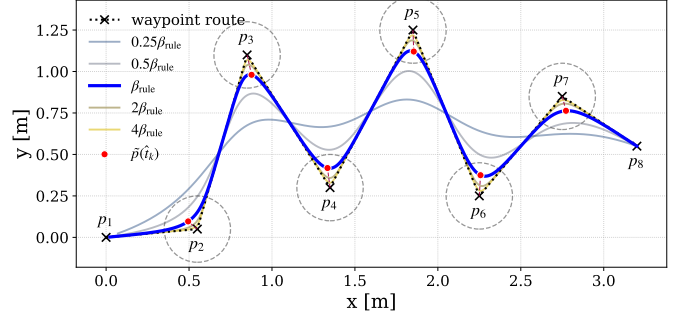


Fig. 2. Smooth flyby surrogate in (3). The blue curve corresponds to $\beta = \beta_{\text{rule}}$, where β_{rule} is defined in (4). Dashed circles indicate the waypoint deviation tolerance ϵ_{wp} . In this example, the surrogate trajectories with $\beta \geq \beta_{\text{rule}}$ remain within the prescribed tolerance.

The exact velocity jumps depend on the timing variables. We therefore select β using the following geometry-based local rule. Let $\theta_k^{(i)}$ denote the angle between consecutive route segments at waypoint k . The maximum possible velocity jump is bounded by

$$D(\theta) = \max \left\{ \sqrt{v_{\text{max}}^2 + v_{\text{min}}^2 - 2v_{\text{max}}v_{\text{min}} \cos \theta}, 2v_{\text{max}} \sin \frac{\theta}{2} \right\}.$$

To obtain a simple local rule, we neglect overlap between neighboring transitions and retain only the largest local velocity jump. This gives

$$\beta \geq \frac{\ln 2}{\epsilon_{\text{wp}}} \max_{i,k} D(\theta_k^{(i)}). \quad (4)$$

In the numerical examples, β is fixed before optimization using (4) for a given tolerance ϵ_{wp} . Fig. 2 shows an example.

IV. FIRST-ORDER OPTIMIZATION

All constraints in (1) except collision avoidance are affine or box constraints. We use an ADMM framework to separate the timing and speed-bound terms from the collision avoidance terms [20].

A. Split Form

Define the stacked timing vector

$$\mathbf{t} := [\mathbf{t}^{(1)}; \dots; \mathbf{t}^{(K)}] \in \mathbb{R}^N.$$

We introduce $\mathbf{x}$ as the stacked segment duration vector and $\mathbf{z}$ as a local copy of $\mathbf{t}$ used in the collision avoidance terms. Let $\mathbf{D}$ denote the stacked forward difference operator, so that

$$\mathbf{D}\mathbf{t} = \mathbf{x} = [\Delta \mathbf{t}^{(1)}; \dots; \Delta \mathbf{t}^{(K)}].$$

A selector matrix $\mathbf{E}$ imposes the prescribed departure time constraints and the fixed-terminal time constraints:

$$\mathbf{E}\mathbf{t} = \mathbf{e} = [t_s^{(1)}, \dots, t_s^{(K)}, \{t_f^{(i)}\}_{i \in \mathcal{K}_f}]^T.$$

The objective is represented by $\mathbf{q} \in \mathbb{R}^N$ satisfying

$$\mathbf{q}^T \mathbf{t} = \sum_{i \in \mathcal{K}_u} t_{N_i}^{(i)}.$$

The speed and safety sets are

$$\begin{aligned}\mathcal{V} &:= \{\mathbf{x} : \mathbf{d}/v_{\max} \leq \mathbf{x} \leq \mathbf{d}/v_{\min}\}, \\ \mathcal{C} &:= \{\mathbf{z} : \|\tilde{\mathbf{p}}^{(i)}(\mathbf{z}^{(i)}, \tau) - \tilde{\mathbf{p}}^{(j)}(\mathbf{z}^{(j)}, \tau)\|_2 \geq d_{\text{safe}}, \\ &\quad \forall (i, j) \in \mathcal{P}, \forall \tau \in \mathcal{T}_d\}.\end{aligned}$$

Using the indicator functions $I_{\mathcal{V}}$ and $I_{\mathcal{C}}$, the split problem is

$$\begin{aligned}\underset{\mathbf{t}, \mathbf{x}, \mathbf{z}}{\text{minimize}} \quad & \mathbf{q}^T \mathbf{t} + I_{\mathcal{V}}(\mathbf{x}) + I_{\mathcal{C}}(\mathbf{z}) \\ \text{subject to} \quad & \mathbf{E}\mathbf{t} = \mathbf{e}, \\ & \mathbf{D}\mathbf{t} - \mathbf{x} = \mathbf{0}, \\ & \mathbf{t} - \mathbf{z} = \mathbf{0}.\end{aligned}\quad (5)$$

B. ADMM Updates

Let $\rho > 0$ be the ADMM penalty parameter. We use scaled dual variables $\mathbf{u}_E$, $\mathbf{u}_x$, and $\mathbf{u}_z$ for $\mathbf{E}\mathbf{t} = \mathbf{e}$, $\mathbf{D}\mathbf{t} = \mathbf{x}$, and $\mathbf{t} = \mathbf{z}$, respectively.

t-update: Given $(\mathbf{x}^k, \mathbf{z}^k, \mathbf{u}_E^k, \mathbf{u}_x^k, \mathbf{u}_z^k)$, the $\mathbf{t}$ -subproblem is quadratic. From its optimality condition, $\mathbf{t}^{k+1}$ is obtained by solving the sparse linear system

$$\begin{aligned}(\mathbf{E}^T \mathbf{E} + \mathbf{D}^T \mathbf{D} + \mathbf{I}) \mathbf{t}^{k+1} &= \mathbf{E}^T (\mathbf{e} - \mathbf{u}_E^k) \\ &+ \mathbf{D}^T (\mathbf{x}^k - \mathbf{u}_x^k) + \mathbf{z}^k - \mathbf{u}_z^k - \frac{1}{\rho} \mathbf{q}.\end{aligned}\quad (6)$$

The coefficient matrix is constant and can be factorized once.

x-update: The $\mathbf{x}$ -update is the projection onto the speed bounds:

$$\mathbf{x}^{k+1} = \Pi_{\mathcal{V}}(\mathbf{D}\mathbf{t}^{k+1} + \mathbf{u}_x^k), \quad (7)$$

where the projection is applied elementwise as

$$\Pi_{\mathcal{V}}(\mathbf{y}) = \min\left(\frac{\mathbf{d}}{v_{\min}}, \max\left(\frac{\mathbf{d}}{v_{\max}}, \mathbf{y}\right)\right).$$

z-update: Since the collision avoidance feasible set is nonlinear and nonconvex, an exact projection would require solving a nonlinear subproblem at every ADMM iteration. We instead define a finite-grid violation penalty and construct a local first-order correction.

For each pair $(i, j) \in \mathcal{P}$ and temporal grid point $\tau_m \in \mathcal{T}_d$, define

$$\begin{aligned}\mathbf{r}_{ijm}(\mathbf{z}) &:= \tilde{\mathbf{p}}^{(i)}(\mathbf{z}^{(i)}, \tau_m) - \tilde{\mathbf{p}}^{(j)}(\mathbf{z}^{(j)}, \tau_m), \\ d_{ijm}(\mathbf{z}) &:= \|\mathbf{r}_{ijm}(\mathbf{z})\|_2.\end{aligned}$$

The nodal violation is

$$c_{ijm}^{\text{node}}(\mathbf{z}) := \left[1 - \frac{d_{ijm}(\mathbf{z})}{d_{\text{safe}}}\right]_+.$$

To reduce sensitivity to the temporal grid phase, we also evaluate the closest approach within each interval $[\tau_m, \tau_{m+1}]$. Let

$$\mathbf{e}_{ijm} := \mathbf{r}_{ij, m+1} - \mathbf{r}_{ijm}.$$

Using linear interpolation of the relative position, the closest point parameter is approximated as

$$s_{ijm}^* := \text{clip}\left(-\frac{\mathbf{r}_{ijm}^T \mathbf{e}_{ijm}}{\|\mathbf{e}_{ijm}\|_2^2}, 0, 1\right).$$

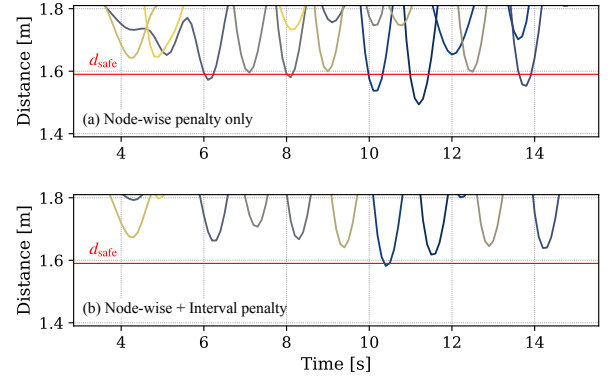


Fig. 3. Bottleneck example comparing the nodal penalty with the combined nodal and interval penalty. The curves show inter-agent distances for the pairs in $\mathcal{P}$, and the red horizontal line indicates d_{safe} . In this example, the interval term eliminates the between-node violation observed when only the nodal penalty is used.

The corresponding interval distance is

$$d_{ijm}^*(\mathbf{z}) := \|\mathbf{r}_{ijm} + s_{ijm}^* \mathbf{e}_{ijm}\|_2.$$

The interval term is used only when the closest point lies inside the interval and yields a distance smaller than those at both endpoints. This avoids double counting nodal violations. Define

$$d_{ijm}^{\text{end}} := \min\{d_{ijm}, d_{ij, m+1}\}.$$

The interval activation indicator is

$$\chi_{ijm} = \begin{cases} 1, & \epsilon_s < s_{ijm}^* < 1 - \epsilon_s \text{ and } d_{ijm}^* < d_{ijm}^{\text{end}}, \\ 0, & \text{otherwise,} \end{cases}$$

where $\epsilon_s \in (0, 1/2)$. The interval violation is

$$c_{ijm}^{\text{int}}(\mathbf{z}) := \chi_{ijm} \left[1 - \frac{d_{ijm}^*(\mathbf{z})}{d_{\text{safe}}}\right]_+.$$

The collision penalty used in the $\mathbf{z}$ -update is the finite-grid violation sum

$$f(\mathbf{z}) = \sum_{(i,j) \in \mathcal{P}} \sum_{m=1}^M c_{ijm}^{\text{node}}(\mathbf{z}) + \sum_{(i,j) \in \mathcal{P}} \sum_{m=1}^{M-1} c_{ijm}^{\text{int}}(\mathbf{z}). \quad (8)$$

Fig. 3 shows the effect of the interval closest approach term.

Replacing the collision constraints with (8) gives the following penalized proximal subproblem:

$$\mathbf{z}^{k+1} = \arg \min_{\mathbf{z}} f(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{z} - (\mathbf{t}^{k+1} + \mathbf{u}_z^k)\|_2^2. \quad (9)$$

The penalty f is nonlinear, so solving (9) exactly at each iteration is costly. We instead compute a local correction at the proximal point $\hat{\mathbf{z}}^k := \mathbf{t}^{k+1} + \mathbf{u}_z^k$. Let $f_k := f(\hat{\mathbf{z}}^k)$ and $\mathbf{g}_k \in \partial f(\hat{\mathbf{z}}^k)$, where ∂f denotes the subdifferential of the finite-grid hinge penalty. We linearize f at $\hat{\mathbf{z}}^k$ as $f(\hat{\mathbf{z}}^k + \Delta \mathbf{z}) \approx f_k + \mathbf{g}_k^T \Delta \mathbf{z}$ and define the regularized local correction by

$$\Delta \mathbf{z}^k = \arg \min_{\Delta \mathbf{z}} \frac{1}{2} (f_k + \mathbf{g}_k^T \Delta \mathbf{z})^2 + \frac{\delta_z}{2} \|\Delta \mathbf{z}\|_2^2,$$

where $\delta_z > 0$ regularizes the correction. The closed-form correction is

$$\Delta \mathbf{z}^k = -\frac{f_k}{\|\mathbf{g}_k\|_2^2 + \delta_z} \mathbf{g}_k, \quad (10)$$

and the non-inertial update is

$$\mathbf{z}^{k+1} = \hat{\mathbf{z}}^k + \Delta \mathbf{z}^k.$$

The correction in (10) reduces the linearized violation model:

$$f_k + \mathbf{g}_k^T \Delta \mathbf{z}^k = f_k \frac{\delta_z}{\|\mathbf{g}_k\|_2^2 + \delta_z}.$$

Thus, for $f_k > 0$ and $\mathbf{g}_k \neq \mathbf{0}$, the relaxed linearized penalty strictly decreases. However, due to the nonconvexity of f , this decrease property is local and does not imply monotone decrease of the original penalty. In congested scheduling scenarios, the iterates may enter a local passing-order basin in which the finite-grid violation stops decreasing sufficiently.

To reduce this stagnation, the implementation uses a gated inertial term. Let $\mathbf{m}^k$ denote the correction memory. The implemented update is

$$\begin{aligned} \mathbf{m}^{k+1} &= \Delta \mathbf{z}^k + \alpha^k \mathbf{m}^k, \\ \mathbf{z}^{k+1} &= \hat{\mathbf{z}}^k + \mathbf{m}^{k+1}. \end{aligned}$$

The gain α^k is selected from a sliding window progress test. Let $\tilde{f}^k$ be the monitored penalty value, and define the lower envelope over the recent window

$$B^k := \min_{\ell \in \{k-m_{\text{stag}}+1, \dots, k\}} \tilde{f}^\ell.$$

The relative decrease of the envelope is

$$r_f^k := \frac{B^{k-m_{\text{stag}}} - B^k}{B^{k-m_{\text{stag}}}}.$$

The inertial term is activated only when the envelope decrease is small:

$$\alpha^k = \begin{cases} \alpha_{\text{active}}, & r_f^k < \varepsilon_{\text{stag}} \text{ and } \tilde{f}^k > 0, \\ 0, & \text{otherwise.} \end{cases}$$

During the inertial phase, the gain is reset to zero if the lower envelope improves or if the current penalty exceeds the envelope by a prescribed ratio:

$$\alpha^k = 0, \quad \text{if } B^k < B^{k-m_{\text{stag}}} \text{ or } \tilde{f}^k > (1 + \varepsilon_{\text{inc}})B^k.$$

Dual update: After the three primal blocks are updated, the scaled dual variables are updated as

$$\begin{aligned} \mathbf{u}_E^{k+1} &= \mathbf{u}_E^k + \mathbf{E} \mathbf{t}^{k+1} - \mathbf{e}, \\ \mathbf{u}_x^{k+1} &= \mathbf{u}_x^k + \mathbf{D} \mathbf{t}^{k+1} - \mathbf{x}^{k+1}, \\ \mathbf{u}_z^{k+1} &= \mathbf{u}_z^k + \mathbf{t}^{k+1} - \mathbf{z}^{k+1}. \end{aligned}$$

Here, $\mathbf{u}_E$, $\mathbf{u}_x$, and $\mathbf{u}_z$ correspond to the equality constraints, the speed bound, and the collision avoidance constraints, respectively.

C. Convergence Remarks and Residual Floors

The scheme is applied to a finite dimensional discretization of the continuous-time scheduling problem. The collision free set is nonconvex, and the $\mathbf{z}$ -update uses an inexact gradient based correction rather than an exact projection. We therefore do not claim global optimality or convergence to a stationary point of the original continuous-time problem. The algorithm is used as a residual controlled inexact ADMM scheme.

At each iteration, we monitor the scalar primal residual

$$r_{\text{pri}}^k := \max \{ \|\mathbf{E} \mathbf{t}^k - \mathbf{e}\|_\infty, \|\mathbf{D} \mathbf{t}^k - \mathbf{x}^k\|_\infty, \|\mathbf{t}^k - \mathbf{z}^k\|_\infty \},$$

and the maximum safety violation

$$r_{\text{safe}}^k := \max_{(i,j) \in \mathcal{P}, \tau_m \in \mathcal{T}_d} \left[1 - \frac{\|\mathbf{r}_{ijm}(\mathbf{z}^k)\|_2}{d_{\text{safe}}} \right]_+.$$

The iteration is terminated when

$$r_{\text{pri}}^k \leq \varepsilon_{\text{pri}}, \quad r_{\text{safe}}^k \leq \varepsilon_{\text{safe}},$$

where ε_{pri} and $\varepsilon_{\text{safe}}$ are prescribed residual floors. The returned iterate is accepted as an $(\varepsilon_{\text{pri}}, \varepsilon_{\text{safe}})$ feasible solution of the finite-grid surrogate problem. The primal residual condition enforces consistency between the timing variable and the auxiliary variable, while r_{safe}^k checks the minimum pairwise distance of the trajectories.

All reported trajectories are additionally evaluated on a finer temporal grid after optimization. This process checks for constraint violations within the intervals defined by the optimization grid. In the limiting case where the residual floors are driven to zero and a convergent subsequence exists, the corresponding limit point is feasible for the finite-grid surrogate. Stationarity or local convergence of the implemented inexact $\mathbf{z}$ -correction, especially with the gated inertial safeguard, would require additional regularity and subproblem accuracy assumptions.

V. NUMERICAL EXAMPLES

The experiments evaluate three aspects: performance relative to fixed-route baselines in structured environments, applicability to additional temporal constraints, and capability in random crossing stress tests, which are challenging for priority-free timing updates, as coupled passing-order conflicts dominate the problem.

A. Experimental Settings and Benchmark Scenarios

We consider four benchmark classes:

- *Bottleneck:* a dense queueing scenario through a narrow shared corridor.
- *Aligned crossing:* closely aligned fixed-routes used to evaluate arrival time coordination in structured traffic.
- *Graph trajectory:* agents track paths on a directed geometric graph generated from randomly sampled nodes and Delaunay triangulation. Each agent is assigned the shortest path between a start and a goal node. Directed edges are used to avoid opposing direction conflicts along the same segment.
- *Random crossing:* randomly perturbed waypoint paths through a shared central region, used as a coupled stress test.

Fig. 4 shows representative route geometries.

All benchmark scenarios use a workspace with spatial scale approximately $[-10, 10] \text{ m} \times [-10, 10] \text{ m}$. The speed bounds $[v_{\min}, v_{\max}] = [0.02, 2.0] \text{ m/s}$ are used in all cases. Table II lists the default parameters for the proposed method.

The safety distance d_{safe} is selected by scenario. For the bottleneck and random crossing cases, it is determined by

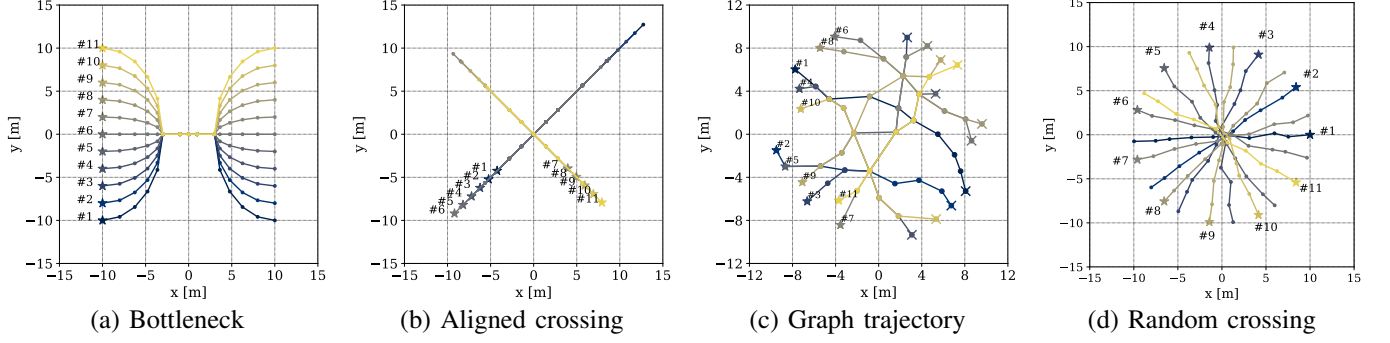


Fig. 4. Representative fixed-route benchmark geometries. The bottleneck and aligned crossing cases are deterministic. The graph trajectory and random crossing cases show representative Monte Carlo instances. Solid lines with markers denote prescribed routes.

TABLE II
DEFAULT PARAMETER SETTINGS

Variable	Value	Description
ρ	10^2	ADMM penalty parameter
N_{iter}	1000	Maximum number of ADMM iterations
M	200	Number of temporal grid nodes
ϵ_{pri}	10^{-8}	Primal residual tolerance
ϵ_{safe}	10^{-3}	Safety violation tolerance
ϵ_{wp}	0.1 m	Flyby waypoint tracking tolerance
m_{stag}	10	Stagnation window size
ϵ_{stag}	10^{-2}	Stagnation threshold
α_{active}	0.7	Inertial gain
ϵ_{inc}	1.0	Penalty increase tolerance

the occupancy density $\phi = K\pi(d_{\text{safe}}/2)^2/A$, which gives $d_{\text{safe}} = 2\sqrt{A\phi/(K\pi)}$, where A is the workspace area. For the aligned-crossing case, we set $\phi = 5\%$. For the graph route experiments, $d_{\text{safe}} = (\sqrt{3}/4)d_{\text{node}}$, where d_{node} is the average distance between network nodes. This value is tied to the geometric scale of the Delaunay network. Fig. 5 reports the corresponding safety distance sweep.

We compare the proposed method with three fixed-route baselines: *mixed-integer programming-second-order cone programming (MIP-SOCP)*, a hierarchical scheduling pipeline that uses mixed-integer programming for passing-order selection [14]; *Priority planning*, a prioritized planner with waiting insertion inspired by continuous-time prioritized safe-interval planning [21]; and *Fixed-route control barrier function quadratic program (CBF-QP)*, a route-projected CBF-QP that adjusts scalar speeds along the prescribed route tangents [22].

B. Benchmark Results

Table III compares the proposed method with the three baselines on standard fixed-route benchmarks. Runtime is not reported because the implementations use different prototype solver backends and levels of code optimization. We report success rate, mean arrival time, and maximum arrival time.

In the bottleneck benchmark, priority planning gives a slightly lower mean arrival time at the lowest density, whereas the proposed method gives comparable or lower mean arrival times as the density increases. In the aligned crossing and graph trajectory benchmarks, the proposed method gives the

lowest mean arrival time in Table III and successfully solves all tested instances. The MIP-SOCP baseline succeeds in most cases but gives larger arrival times in these structured scenarios. The fixed-route CBF baseline, which uses equal priority reactive speed control, fails in most scenarios due to deadlock. We therefore compare the proposed method with the most competitive baseline, the priority planning in graph trajectory scenarios with additional timing constraints.

Table IV reports graph trajectory results under additional timing constraints. The comparison is restricted to the proposed method and Priority planning. The MIP-SOCP baseline is omitted because the formulation in [14] does not include these timing constraints.

In the dwell-time case, each agent dwells for 1.0 s at every intermediate waypoint. Both methods successfully solve all tested instances. Priority planning gives a slightly lower mean

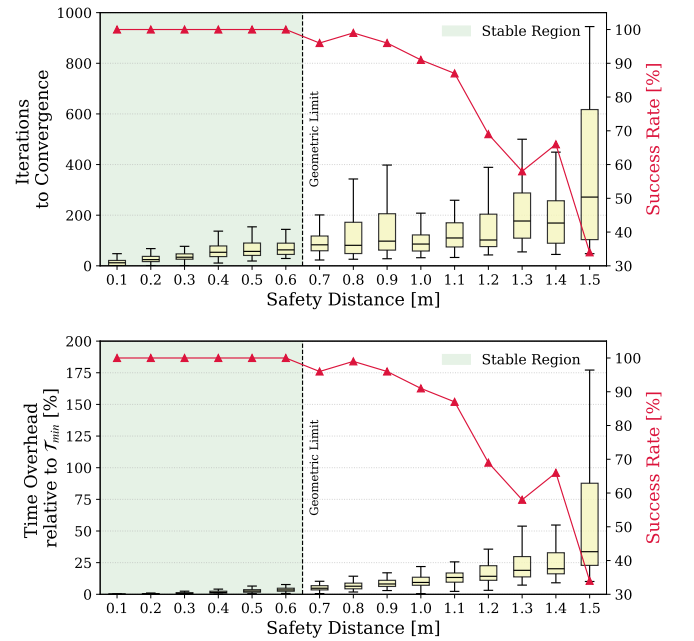


Fig. 5. Graph route safety distance sweep for the proposed method. The box plots show the number of iterations and the time overhead, and the red curve shows the success rate. The shaded region indicates safety distances satisfying $d_{\text{safe}} < (\sqrt{3}/4)d_{\text{node}}$.

TABLE III

BASELINE COMPARISON ON STANDARD FIXED-ROUTE BENCHMARKS. EACH ENTRY REPORTS THE SUCCESS RATE, MEAN ARRIVAL TIME $\bar{T}$, AND MAXIMUM ARRIVAL TIME $T_{\max}$

Scenario	Proposed			MIP-SOCP			Priority planning			fixed-route CBF		
	SR	$\bar{T}$ [s]	$T_{\max}$ [s]	SR	$\bar{T}$ [s]	$T_{\max}$ [s]	SR	$\bar{T}$ [s]	$T_{\max}$ [s]	SR	$\bar{T}$ [s]	$T_{\max}$ [s]
Bottleneck, $\phi = 1\%$	100%	13.32	17.40	100%	15.05	20.35	100%	13.19	16.78	0%	–	–
Bottleneck, $\phi = 5\%$	100%	15.33	21.53	100%	16.97	22.79	100%	15.65	21.48	0%	–	–
Bottleneck, $\phi = 10\%$	100%	16.96	24.74	100%	18.82	27.27	100%	17.07	24.58	0%	–	–
Aligned crossing	100%	11.33	12.72	100%	14.90	18.88	100%	11.34	13.03	0%	–	–
Graph trajectory	100%	9.16	11.11	97%	9.54	12.16	99%	9.25	11.97	47%	9.35	11.44

TABLE IV

COMPARISON ON GRAPH ROUTE BENCHMARKS WITH TIMING CONSTRAINTS. THE DWELL-TIME CASE IMPOSES A UNIFORM DWELL DURATION AT EVERY INTERMEDIATE WAYPOINT. IN THE FIXED-TERMINAL TIME CASE, ONE AGENT IS ASSIGNED A TARGET TERMINAL TIME COMPUTED USING THE NOMINAL SPEED DEFINED IN SECTION V-A. EACH ENTRY REPORTS THE SUCCESS RATE, MEAN ARRIVAL TIME $\bar{T}$, AND MAXIMUM ARRIVAL TIME $T_{\max}$. ARRIVAL TIME STATISTICS ARE COMPUTED OVER SUCCESSFUL RUNS

Scenario	Proposed			Priority planning		
	SR	$\bar{T}$ [s]	$T_{\max}$ [s]	SR	$\bar{T}$ [s]	$T_{\max}$ [s]
Graph with waypoint dwell time	100%	17.22	22.80	100%	17.12	24.40
Graph with fixed-terminal time	100%	9.93	20.08	98%	10.01	20.11

TABLE V

RANDOM CROSSING STRESS TEST. SR IS COMPUTED OVER ALL GENERATED INSTANCES. ITERATION AND COST STATISTICS ARE COMPUTED OVER SUCCESSFUL RUNS AND REPORTED AS MEDIAN $[Q_{25}, Q_{75}]$

K	ϕ	SR	Iter.	Cost
3	1%	100%	57 [53, 64]	1.21 [1.17, 1.24]
	5%	100%	86 [78, 91]	1.56 [1.47, 1.86]
	10%	100%	90 [87, 94]	1.69 [1.64, 1.99]
7	1%	99%	211 [152, 264]	1.31 [1.26, 1.41]
	5%	100%	206 [174, 251]	1.88 [1.70, 2.07]
	10%	99%	273 [234, 346]	2.49 [2.20, 2.78]
11	1%	87%	546 [434, 678]	1.49 [1.43, 1.56]
	5%	64%	626 [531, 746]	2.59 [2.27, 2.97]
	10%	19%	730 [656, 830]	3.72 [3.28, 4.23]

arrival time, while the proposed method gives a lower maximum arrival time.

In the fixed-terminal time case, we impose a target arrival time on a specific agent. The target time is defined as the duration required to traverse the given route at the mean of the prescribed speed bounds. The proposed method successfully solves all tested instances and yields slightly lower mean and maximum arrival times than the baseline. In contrast, priority planning fails to find a feasible solution in 2% of the tested instances.

These results indicate that the proposed priority-free timing formulation is competitive with the priority planning baseline on the structured fixed-route benchmarks.

C. Stress Test and Parameter Sensitivity Analysis

The random crossing benchmark is designed as a stress test for the proposed method. In this scenario, multiple routes overlap in a confined central region, and agents approaching from opposite sides share long path portions. These interactions require coupled passing-order decisions, making the case challenging for a formulation without prescribed priorities. We use this benchmark to assess the limits of the local timing update.

TABLE VI

PARAMETER SENSITIVITY IN THE RANDOM-CROSSING STRESS TEST. SR IS COMPUTED OVER ALL GENERATED INSTANCES, WHEREAS ITERATION COUNT AND COST ARE REPORTED OVER SUCCESSFUL RUNS AS THE MEDIAN AND INTERQUARTILE INTERVAL $[Q_{25}, Q_{75}]$. BOLD ENTRIES INDICATE THE DEFAULT PARAMETER SETTING AND ITS CORRESPONDING RESULTS.

Param.	Value	SR	Iter.	Cost
α_{active}	0	7%	847 [728, 873]	1.94 [1.90, 1.98]
	0.3	22%	780 [590, 884]	2.03 [1.96, 2.19]
	0.7	64%	626 [531, 746]	2.59 [2.27, 2.97]
	1.0	96%	585 [460, 775]	4.54 [3.61, 6.34]
ε_{inc}	0.5	54%	652 [569, 761]	2.29 [2.11, 2.52]
	1.0	64%	626 [531, 746]	2.59 [2.27, 2.97]
	2.0	86%	576 [464, 714]	2.87 [2.56, 3.17]
ρ	10	70%	594 [496, 701]	2.63 [2.38, 3.02]
	100	64%	626 [531, 746]	2.59 [2.27, 2.97]
	1000	71%	534 [448, 697]	2.63 [2.26, 2.99]

Table V reports the random crossing stress test. The success rate is computed over all tested instances, while the cost and iteration statistics are conditioned on successful runs. The results show that problem difficulty increases with both the agent count K and the occupancy density ϕ . For $K = 3$, all instances at the tested densities are solved within the iteration budget. For $K = 7$, the success rate remains high, with larger iteration counts and costs. For $K = 11$, the success rate decreases from 87% at $\phi = 1\%$ to 19% at $\phi = 10\%$. A failed run means that the residual and safety criteria were not satisfied within the iteration budget. These trends indicate that, as congestion increases, the proposed priority-free approach has more difficulty entering a feasible passing-order basin within the prescribed iteration budget.

The collision avoidance auxiliary variable update includes an inertial safeguard to mitigate such local stagnation. Table VI evaluates its effect on the $K = 11$, $\phi = 5\%$ random crossing set. The results show a tradeoff between success rate and cost. With $\alpha_{\text{active}} = 0$, the success rate is 7%, and the low cost reflects the easier instances that were solved. Increasing α_{active} improves the success rate, up to 96% at $\alpha_{\text{active}} = 1.0$, but also increases the returned cost. A similar trend appears for ε_{inc} :

a looser reset condition improves the success rate but yields higher cost schedules. The ADMM penalty ρ has a smaller effect over the tested range.

The random crossing results indicate that highly symmetric, densely coupled crossing patterns are challenging for the proposed priority-free local timing update. In these cases, many agents interact near the same region, and the iteration can become trapped in a local passing-order basin. The non-inertial update alone is often insufficient to escape such stagnation. The inertial safeguard improves the success rate by continuing a useful correction direction when progress stalls, as shown in Table VI. However, this benefit comes with a cost trade-off, and some dense cases still remain unsolved within the iteration budget. Addressing these passing-order bottlenecks without explicit priority assignment is left for future work.

VI. CONCLUSION

This paper presents a priority-free temporal optimization framework for fixed-route multi-agent systems. The proposed framework optimizes waypoint passage times to generate velocity profiles along given routes, subject to speed limits, pairwise safety constraints, and temporal requirements such as waypoint dwell times and fixed terminal times.

In the standard benchmarks, the proposed method yielded feasible schedules across all tested instances, whereas order-based baselines occasionally failed to compute feasible schedules in the graph trajectory scenarios. Furthermore, the proposed method demonstrated slightly better performance in most cases and achieved comparable results in environments with dwell time constraints and fixed-terminal times. These results demonstrate that the proposed priority-free framework is highly competitive with conventional order-based approaches.

The results also clarify the intended limitations of our approach. Dense random crossing is inherently order-dominated because of its structural symmetry, making it difficult to establish feasible yielding patterns without explicit sequence selection. In this stress test, the proposed local timing update solved all tested $K = 3$ cases, but for $K = 11$ the success rate decreased from 87% at $\phi = 1\%$ to 19% at $\phi = 10\%$. The inertial safeguard mitigated some of this stagnation: increasing α_{active} from 0.7 to 1.0 increased the success rate from 64% to 96%, but the median cost increased from 2.59 to 4.54. These observations clarify a limitation of the proposed priority-free formulation. While the method is effective as a temporal optimizer for fixed-routes with waypoint-level timing constraints, dense order-dominated cases may require explicit mechanisms for resolving local passing-order stagnation. Future work will address these bottlenecks associated with resolving passing orders without explicit priority variables while retaining the waypoint time formulation and its support for dwell time and terminal time constraints.

ACKNOWLEDGMENT

During the preparation of this work, the authors used ChatGPT-5.5 to improve grammar, spelling, and phrasing, without changing technical contents. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

REFERENCES

- [1] D. Bareiss and J. van den Berg, "Generalized reciprocal collision avoidance," *Int. J. Robot. Res.*, vol. 34, no. 12, pp. 1501–1514, Oct. 2015.
- [2] —, "Reciprocal collision avoidance for robots with linear dynamics using LQR-obstacles," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2013, pp. 3847–3853.
- [3] A. Best, S. Narang, and D. Manocha, "Real-time reciprocal collision avoidance with elliptical agents," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2016, pp. 298–305.
- [4] U. Borrmann, L. Wang, A. D. Ames, and M. Egerstedt, "Control barrier certificates for safe swarm behavior," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 68–73, 2015.
- [5] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.
- [6] Z.-H. Pang, C.-B. Zheng, J. Sun, Q.-L. Han, and G.-P. Liu, "Distance- and velocity-based collision avoidance for time-varying formation control of second-order multi-agent systems," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 4, pp. 1253–1257, Apr. 2021.
- [7] T. Baca, D. Hert, G. Loianno, M. Saska, and V. Kumar, "Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2018, pp. 6753–6760.
- [8] J. Hu, H. Zhang, L. Liu, X. Zhu, C. Zhao, and Q. Pan, "Convergent multiagent formation control with collision avoidance," *IEEE Trans. Robot.*, vol. 36, no. 6, pp. 1805–1818, Dec. 2020.
- [9] K. Okumura, M. Machida, X. Défago, and Y. Tamura, "Priority inheritance with backtracking for iterative multi-agent path finding," *Artif. Intell.*, vol. 310, Sep. 2022, Art. no. 103752.
- [10] Z. A. Ali and K. Yakovlev, "Safe interval path planning with kinodynamic constraints," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 12 330–12 337.
- [11] F. Augugliaro, A. P. Schoellig, and R. D'Andrea, "Generation of collision-free trajectories for a quadcopter fleet: A sequential convex programming approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2012, pp. 1917–1922.
- [12] A. Kushleyev, D. Mellinger, and V. Kumar, "Towards a swarm of agile micro quadrotors," in *Proc. Robot.: Sci. Syst. (RSS)*, 2012, doi: 10.15607/RSS.2012.VIII.028.
- [13] Y. Huang, J. Tang, and S. Lao, "Cooperative multi-UAV collision avoidance based on a complex network," *Appl. Sci.*, vol. 9, no. 19, Oct. 2019, Art. no. 3943.
- [14] W. Wu, F. Gao, L. Wang, B. Zhou, and S. Shen, "Temporal scheduling and optimization for multi-MAV planning," in *Robotics Research*, ser. Springer Proc. Adv. Robot. Cham, Switzerland: Springer, 2022, vol. 20, pp. 813–831.
- [15] K. Mao, I. Spasojevic, M. Hopkins, M. A. Hsieh, and V. Kumar, "Collision-free time-optimal path parameterization for multi-robot teams," 2024, arXiv:2409.17079.
- [16] F. Bu, B. Li, P. Chen, G. Yu, and Y. Wang, "CosineOpt: Optimization-based centralized cooperative speed planning for multiple CAVs along intersected fixed paths," *IEEE Trans. Intell. Transp. Syst.*, vol. 27, no. 6, pp. 7211–7225, Jun. 2026.
- [17] S. Zhang, J. Li, T. Huang, S. Koenig, and B. Dilkina, "Learning a priority ordering for prioritized planning in multi-agent path finding," in *Proc. Int. Symp. Comb. Search (SoCS)*, 2022, pp. 208–216.
- [18] H. Ma, G. Wagner, A. Felner, J. Li, T. K. S. Kumar, and S. Koenig, "Multi-agent path finding with deadlines," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2018, pp. 417–423.
- [19] J. Gao, Q. Liu, S. Chen, K. Yan, X. Li, and Y. Li, "Multi-agent path finding with time windows: Preliminary results," in *Proc. Int. Conf. Auton. Agents Multiagent Syst. (AAMAS)*, 2023, pp. 2586–2588.
- [20] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
- [21] K. Kasaura, M. Nishimura, and R. Yonetani, "Prioritized safe interval path planning for multi-agent pathfinding with continuous time on 2D roadmaps," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 10 494–10 501, Oct. 2022.
- [22] P. Mestres, C. Nieto-Granda, and J. Cortés, "Distributed safe navigation of multi-agent systems using control barrier function-based controllers," *IEEE Robot. Autom. Lett.*, vol. 9, no. 7, pp. 6760–6767, Jul. 2024.