

Multi-Agent Path Finding in Continuous Spaces with Projected Diffusion Models

Jinhao Liang¹, Jacob K. Christopher¹, Sven Koenig², Ferdinando Fioretto¹

¹ Department of Computer Science, University of Virginia, Charlottesville, VA 22903, USA

² Department of Computer Science, University of California, Irvine, CA 92697, USA

Abstract

Multi-Agent Path Finding (MAPF) is a fundamental problem in robotics, requiring the computation of collision-free paths for multiple agents moving from their respective start to goal positions. Coordinating multiple agents in a shared environment poses significant challenges, especially in continuous spaces where traditional optimization algorithms struggle with scalability. Moreover, these algorithms often depend on discretized representations of the environment, which can be impractical in image-based or high-dimensional settings. Recently, diffusion models have shown promise in single-agent path planning, capturing complex trajectory distributions and generating smooth paths that navigate continuous, high-dimensional spaces. However, directly extending diffusion models to MAPF introduces new challenges since these models struggle to ensure constraint feasibility, such as inter-agent collision avoidance. To overcome this limitation, this work proposes a novel approach that integrates constrained optimization with diffusion models for MAPF in continuous spaces. This unique combination directly produces *feasible* multi-agent trajectories that respect collision avoidance and kinematic constraints. The effectiveness of our approach is demonstrated across various challenging simulated scenarios of varying dimensionality.

Introduction

Multi-agent path finding (MAPF) is a critical problem in robotics and autonomous systems, where the goal is to compute collision-free paths for multiple agents navigating from their respective start positions to designated goals in a shared environment (Stern et al. 2019). This problem finds formulation in numerous domains, such as gaming, automated warehouses, and aircraft taxiing (Li et al. 2021). The problem is inherently challenging due to the high-dimensional joint configuration space and the need for coordination among multiple agents to avoid collision. The complexity increases exponentially with the number of agents, making scalability a significant issue for traditional MAPF algorithms. Additionally, existing studies typically consider discrete environments (Stern et al. 2019; Hopcroft, Schwartz, and Sharir 1984), thus further limiting their applicability in scenarios in-the-wild.

The complexity of MAPF in continuous or high-dimensional environments calls for approaches that move beyond traditional discretized methods. Within this con-

text, trajectory optimization has recently been tackled using diffusion models, a powerful class of generative models originally developed for tasks in image and signal processing (Song and Ermon 2019; Ho, Jain, and Abbeel 2020). These models approximate high-dimensional probability distributions by iteratively denoising sampled trajectories, leveraging strong inductive biases that provide effective heuristics even for very complex distributions. Their adaptability has accelerated their adoption across diverse engineering domains, including single-agent robotic path planning (Carvalho et al. 2023; Christopher, Baek, and Fioretto 2024). By learning the underlying distribution of (feasible) trajectories, diffusion models can produce diverse solutions that may be missed by traditional planners due to inductive bias. Additionally, these models possess the ability to generate smooth trajectories that effectively navigate high-dimensional spaces with complex obstacles while directly processing real-world representations of the environment.

However, despite their potential, current diffusion models face significant challenges in generating *feasible* trajectories. Existing approaches often rely on costly rejection sampling methods, which attempt to identify a feasible subset from a larger set of initially generated trajectories, if such a subset exists at all (Carvalho et al. 2023; Christopher, Baek, and Fioretto 2024). Additionally, despite their adoption in single-agent scenarios, extending diffusion models to MAPF presents additional challenges. The introduction of multiple agents requires the consideration of collision avoidance among agents, as well as kinematic constraints.

To address this limitation, this paper proposes a novel integration of constrained optimization in diffusion processes tailored to MAPF in continuous spaces. The proposed method leverages the projection-based method for diffusion models (Christopher, Baek, and Fioretto 2024), which has been recently introduced to steer the learned data distribution to satisfy some constraints of interest. This approach reformulates the sampling process as a constrained optimization problem, projecting the outputs of DMs at each sampling step into the feasible region. However, the MAPF feasible region is defined by a set of nonconvex nonlinear constraints (NNCs), which massively complicates the application of these diffusion models in scenarios with a large number of agents or with moving objects. To address this limitation and enhance computational efficiency, we propose an

augmented Lagrangian method that relaxes the NNCs, making the proposed approach suitable for complex applications where classical MAPF algorithms fall short. This novel integration enables generative diffusion models to generate, for the first time, collision-free trajectories for scenarios involving dozens of agents and obstacles.

The paper’s contributions are summarized as follows:

1. We introduce a novel formulation of MAPF in continuous spaces using diffusion models, enabling the simultaneous generation of trajectories for all agents in a single framework.
2. To address the challenge of constraint satisfaction, we adapt the projection mechanism for MAPF by embedding constraints directly into the diffusion process, projecting the generated solutions into the feasible region.
3. Given the computational intractability of MAPF in continuous spaces, especially with a large number of agents, we develop an augmented Lagrangian approach to accelerate the projection process. This enhancement significantly reduces computational overhead, making the method scalable and practical for real-world applications.
4. We assess the ability of our approach to generate feasible MAPF trajectory empirically over several challenging scenarios, which include maps with narrow corridors, dense obstacles, and a large number of agents.

Related Work

Multi-Agent Path Finding. The classical MAPF problem assumes that time and the environment are discretized into time steps and grids, respectively (Stern et al. 2019). Under this assumption, numerous search algorithms have been developed to efficiently obtain near-optimal solutions for MAPF in discrete environments, even for scenarios involving hundreds of agents (Li et al. 2019; Li, Ruml, and Koenig 2021; Okumura et al. 2022a; Li, Ruml, and Koenig 2021). While this assumption significantly reduces the complexity of MAPF, it creates a gap between the problem’s formulation and real-world applications, posing challenges in many domains (Shaoul et al. 2024). Some studies attempt to generalize MAPF to continuous environments using probabilistic roadmaps (Kavraki et al. 1996) and rapidly exploring random trees (LaValle 1998). Another line of research formulates MAPF as a constrained optimization problem with continuous variables, employing methods such as sequential convex programming (Augugliaro, Schoellig, and D’Andrea 2012; Chen, Cutler, and How 2015) and the alternating direction method of multipliers (Chen et al. 2023). However, these methods often fail to find any solution if there are a large number of agents and obstacles, even if one exists.

Path Finding with Generative Models. There has been a growing interest in leveraging generative models for path finding problems. Existing studies primarily focus on using diffusion models to solve single-agent path finding problems (Janner et al. 2022; Carvalho et al. 2023). Besides these approaches, Okumura et al. utilizes a conditional variational autoencoder to predict cooperative timed roadmaps to aid in solving MAPF in continuous spaces. Shaoul et al.

uses diffusion models to generate a trajectory for a single agent and employs classical searching algorithms to determine the final solutions. However, these methods do not ensure the feasibility of the diffusion model outputs and cannot directly generate collision-free paths. In contrast, our approach integrates optimization techniques into diffusion models to directly generate feasible MAPF solutions in continuous spaces, even in scenarios with a significant number of obstacles, while ensuring feasibility.

Preliminaries

Diffusion Models. Diffusion Models (DMs) are a class of probabilistic generative models designed to transform simple noise distributions into complex target data distributions. They operate through two Markov chains: (1) a forward diffusion process that progressively adds noise to data samples, and (2) a reverse denoising process that iteratively removes noise to recover data samples (Yang et al. 2023). In the forward process, Gaussian noise is incrementally added to the data $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ over T timesteps, producing a sequence of noisy samples $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T$:

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}), \quad (1)$$

where $\beta_t \in (0, 1)$ is a predefined variance schedule controlling the amount of noise added at each step, ensuring that the final distribution approximates an isotropic Gaussian. The reverse process begins with a sample from the noise distribution $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and aims to reconstruct data samples by sequentially removing noise:

$$p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)), \quad (2)$$

where θ represents the learned parameters of neural networks, and $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$ are functions parameterizing the mean and covariance, respectively. Through this process, DMs iteratively transform random noise samples into data resembling the target distribution $q(\mathbf{x}_0)$.

Score-based DMs employ a neural network s_θ to approximate the score function $\nabla \mathbf{x}_t \log q(\mathbf{x}_t)$, which points in the direction of the steepest ascent of the data density at each noise level (Song et al. 2020). The training objective is to minimize the difference between the true score and the network’s approximation (Yang et al. 2023):

$$\begin{aligned} & \mathbb{E}_{\substack{t \sim 1, \dots, T, \mathbf{x}_0 \sim q(\mathbf{x}_0), \\ \mathbf{x}_t \sim q(\mathbf{x}_t | \mathbf{x}_0)}} |\nabla \mathbf{x}_t \log q(\mathbf{x}_t) - s_\theta(\mathbf{x}_t, t)|^2 \eta(t) \beta_t \\ & = \mathbb{E}_{\substack{t \sim 1, \dots, T, \mathbf{x}_0 \sim q(\mathbf{x}_0), \\ \mathbf{x}_t \sim q(\mathbf{x}_t | \mathbf{x}_0)}} |\nabla \mathbf{x}_t \log q(\mathbf{x}_t | \mathbf{x}_0) - s_\theta(\mathbf{x}_t, t)|^2 \eta(t) \beta_t + \text{const}, \end{aligned} \quad (3a)$$

where $q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_0, \beta_t \mathbf{I})$ and $\eta(t)$ is a positive weighting function.

As shown by Yang et al. (2023), classical DMs are a special case of score-based DMs. In the subsequent sections, our focus will be on score-based DMs due to their flexibility and effectiveness.

Multi-Agent Path Finding in Continuous Space

Multi-Agent Path Finding (MAPF) involves computing collision-free trajectories for multiple agents moving from

their respective start locations to designated goals within a shared environment. Consider a set of N_a agents $\mathcal{A} = \{a_1, a_2, \dots, a_{N_a}\}$ operating on a two-dimensional plane, in a continuous space. Each agent a_i is modeled as a sphere with radius r_i and has a trajectory over H time steps denoted by $\pi_i = [\pi_i^1, \pi_i^2, \dots, \pi_i^H]$, where $\pi_i^h = (x_i^h, y_i^h)$ represents the position of agent a_i at time h . The agents have start positions $\mathbf{B} = [b_1, b_2, \dots, b_{N_a}]$ and goal positions $\mathbf{E} = [e_1, e_2, \dots, e_{N_a}]$. In addition, their movement must adhere to kinematic constraints, such as maximum velocities. The environment contains N_o obstacles $\mathcal{O} = \{o_1, \dots, o_{N_o}\}$. The objective is to find a set of trajectories $\mathbf{\Pi} = \{\pi_1, \pi_2, \dots, \pi_{N_a}\}$, each associated with agent a_i , that minimizes a cost function while ensuring feasibility with respect to environmental constraints and inter-agent collision avoidance.

The MAPF problem can be formulated as the following constrained optimization:

$$\min_{\mathbf{\Pi}} \mathcal{J}(\mathbf{\Pi}) \quad (4a)$$

$$\text{s.t. } \mathbf{\Pi} \subseteq \Omega_{\text{obs}}, \quad (4b)$$

$$\pi_i^1 = b_i, \quad \forall i \in [N_a], \quad (4c)$$

$$\pi_i^H = e_i, \quad \forall i \in [N_a], \quad (4d)$$

$$\text{Kinematic constraints on } \mathbf{\Pi}, \quad (4e)$$

$$\text{Collision avoidance between agents in } \mathbf{\Pi}, \quad (4f)$$

where $\mathcal{J} : \mathbb{R}^{N_a \times H \times 2} \rightarrow \mathbb{R}$ is the cost function (e.g., total travel time or energy consumption), and Ω_{obs} denotes the feasible region of the environment considering obstacles. Constraints (4b) ensure that agents avoid obstacles, (4c) and (4d) ensure that each agent starts at and reaches its designated start and end positions, respectively, (4e) enforce kinematic limits, and (4f) prevent inter-agent collisions. In the following, we denote the constraint set (4c)–(4f), with Ω .

The MAPF problem is challenging due to the high dimensionality of the joint configuration space and the need to coordinate multiple agents simultaneously (Stern et al. 2019; Shaoul et al. 2024). Traditional methods often struggle with scalability and may not efficiently handle the continuous nature of real-world environments, as shown in (Augugliaro, Schoellig, and D’Andrea 2012; Shaoul et al. 2024). We seek to address this issue using constrained DMs.

Constrained Diffusion Models

In this section, we first revisit the sampling process for DMs and then investigate the integration of DMs and optimization to constrain the output of DMs satisfying constraints.

Recall The Sampling Process in DMs

Since the sampling process in DMs is a Markov process, we generate \mathbf{x}_0 by iterative sampling from the conditional distribution $\mathbf{x}_t \sim q(\mathbf{x}_t | \mathbf{x}_0)$ as $t \rightarrow 0$, where $q(\mathbf{x}_t | \mathbf{x}_0)$ shifts from Gaussian noise to the training data distribution as t decreases. The sample is optimized with respect to each interim data distribution by M iterations of Stochastic Gradient Langevin Dynamics (SGLD):

$$\mathbf{x}_t^{i+1} = \mathbf{x}_t^i + \gamma_t \nabla_{\mathbf{x}_t^i} \log q(\mathbf{x}_t^i | \mathbf{x}_0) + \sqrt{2\gamma_t} \mathbf{z}, \quad (5)$$

Algorithm 1: PDM

```

1  $\mathbf{x}_T^0 \sim \mathcal{N}(\mathbf{0}, \sqrt{\beta_T} \mathbf{I})$ 
2 for  $t = T$  to 1 do
3    $\gamma_t \leftarrow \beta_t / 2\beta_T$ 
4   for  $i = 1$  to  $M$  do
5      $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}); \mathbf{g} \leftarrow \mathbf{s}_{\theta^*}(\mathbf{x}_t^{i-1}, t)$ 
6      $\mathbf{x}_t^i = \mathcal{P}_{\Omega}(\mathbf{x}_t^{i-1} + \gamma_t \mathbf{g} + \sqrt{2\gamma_t} \mathbf{z})$ 
7    $\mathbf{x}_{t-1}^0 \leftarrow \mathbf{x}_t^M$ 
8 return  $\mathbf{x}_0^0$ 

```

where \mathbf{z} is standard normal, $\gamma_t > 0$ is the step size, and $\nabla_{\mathbf{x}_t^i} \log q(\mathbf{x}_t^i | \mathbf{x}_0)$ is approximated by the learned score function $\mathbf{s}_{\theta}(\mathbf{x}_t, t)$.

Christopher, Baek, and Fioretto (2024) derive theory connecting the application of SGLD for sampling to iterative, gradient-based optimization algorithms. The described process ensures that, under appropriate conditions, samples are distributed according to the target distribution $q(\mathbf{x}_t)$. As shown by Christopher, Baek, and Fioretto (2024), SGLD converges toward a stationary distribution under mild assumptions, transitioning toward deterministic gradient ascent as the stochastic component diminishes. This connects the reverse diffusion process to an optimization problem, minimizing the negative log-likelihood of the data distribution and forming the foundation for constrained sampling via iterative projections.

Projected Diffusion Models

In this subsection, we introduce Projected Diffusion Models (PDM) to ensure that generated outputs satisfy predefined constraints. While the objective remains consistent with traditional score-based DMs, the solution is restricted to lie within a feasible region Ω . This transforms the optimization problem into a constrained formulation (Christopher, Baek, and Fioretto 2024):

$$\min_{\mathbf{x}_T, \dots, \mathbf{x}_1} \sum_{t=T, \dots, 1} -\log q(\mathbf{x}_t | \mathbf{x}_0) \quad (6a)$$

$$\text{s.t. } \mathbf{x}_T, \dots, \mathbf{x}_0 \in \Omega. \quad (6b)$$

The reverse sampling process in PDM aligns closely with that of traditional score-based DMs. Specifically, the score network $\mathbf{s}_{\theta}(\mathbf{x}_t, t)$ estimates the gradient of the objective in Equation (6a), enabling iterative updates as defined in Equation (5). However, the presence of constraints (6b) necessitates a modification to the update rule to maintain feasibility. To address this, PDM employs a projected guidance approach, incorporating constraints into the optimization process.

The projection operator, \mathcal{P}_{Ω} , is defined as solving a constrained optimization problem:

$$\mathcal{P}_{\Omega}(\mathbf{x}) = \arg \min_{\mathbf{y} \in \Omega} d(\mathbf{x}, \mathbf{y}), \quad (7)$$

where $d(\mathbf{x}, \mathbf{y})$ is a distance function, and, unless otherwise $d(\mathbf{x}, \mathbf{y})$ denotes the euclidean distance: $\|\mathbf{y} - \mathbf{x}\|_2^2$, which

identifies the closest feasible point \mathbf{y} within Ω to the input \mathbf{x} .

To ensure feasibility at each step, the projected diffusion model applies the projection operator after updating \mathbf{x}_t , leading to the *projected diffusion model sampling* step:

$$\mathbf{x}_t^{i+1} = \mathcal{P}_\Omega \left(\mathbf{x}_t^i + \gamma_t \nabla_{\mathbf{x}_t^i} \log q(\mathbf{x}_t | \mathbf{x}_0) + \sqrt{2\gamma_t} \mathbf{z} \right), \quad (8)$$

where Ω is the set of constraints and \mathcal{P}_Ω is a projection onto Ω . Throughout the reverse Markov chain, each iteration performs a gradient step to minimize the objective in Equation (6a), while ensuring feasibility through projection. As is the case in this paper, the complete sampling process is outlined in Algorithm 1.

PDM directly minimizes the negative log-likelihood as its core objective, similar to standard unconstrained sampling methods. This approach provides a crucial benefit: *it directly optimizes the probability of generating samples that align with the data distribution*, while simultaneously imposing explicit, verifiable constraints. In the next section, we develop a projection mechanism tailored for MAPF in continuous spaces.

Efficient Projections for MAPFs

While PDM provides a useful method to steer samples generated by the generative model to satisfy relevant constraints, projecting onto nonconvex sets can be a computationally expensive operation, especially when it is required to be computed at each step of the sampling process. To address this shortcoming, we develop a projection mechanism to generate feasible trajectories for all agents. To accelerate the projection process, we adopt the augmented Lagrangian method (ALM) (Boyd et al. 2011) to the projection process.

Collision-free Trajectories Projection Mechanism

In the following, we define the mathematical formulation of the feasible region Ω for the MAPF problem, distinguishing between convex and nonconvex constraints.

Convex Constraints. First, each agent's trajectory must start and end its specified start and goal points, as specified in Constraints (4c) and (4d).

Additionally, agents must adhere to maximum velocity limits between consecutive time steps:

$$(\pi_i^h - \pi_i^{h-1})^2 \leq (v_i^{\max} \Delta t)^2, \quad \forall i \in [N_a], h \in \{2, \dots, H\}, \quad (9)$$

where v_i^{\max} denotes the maximum allowable velocity for agent a_i , and Δt is the time interval between steps.

Together, these constraints define a convex set:

$$\Omega_c = \left\{ \mathbf{\Pi} \in \mathbb{R}^{N_a \times H \times 2} \mid \text{Constr. (4c), (4d), and (9) hold} \right\}. \quad (10)$$

Nonconvex Constraints. To ensure collision avoidance between agents, we impose the following nonconvex constraints:

$$(\pi_i^h - \pi_j^h)^2 \geq (R^a)^2, \quad \forall i, j, i \neq j \in [N_a], h \in [H], \quad (11)$$

where R^a denotes the minimum distance between agents at each time.

Similarly, to avoid collisions between agents and static obstacles, we have:

$$(\pi_i^h - o_j)^2 \geq (R^o)^2, \quad \forall i, j \in [N_a], h \in [H], \quad (12)$$

where R^o denotes the minimum distance between agents and obstacles to guarantee noncollision. Similarly, these two constraints define

$$\Omega_n = \left\{ \mathbf{\Pi} \in \mathbb{R}^{N_a \times H \times 2} \mid \text{Constr. (11), (12), hold} \right\}. \quad (13)$$

The complete feasible set is given by: $\Omega = \Omega_c \cap \Omega_n$. Although the projector \mathcal{P}_Ω can generate feasible MAPF trajectories, the nonconvex constraints result in high computational costs.

ALM for Efficient Projection

To address this issue, we seek to relax the nonconvex constraints in MAPF to transform the original nonconvex quadratically constrained quadratic problem (QCQP) into a convex QCQP. To facilitate analysis, we rewrite the inequality constraints as equalities:

$$\mathcal{H}_a : (R^a)^2 - (\pi_i^h - \pi_j^h)^2 + d_{i,j,h}^a = 0, \quad \forall i, j, i \neq j, \forall h, \quad (14a)$$

$$\mathcal{H}_o : (R^o)^2 - (\pi_i^h - o_j)^2 + d_{i,j,h}^o = 0, \quad \forall i, j, \forall h, \quad (14b)$$

where $d_{i,j,h}^a$ and $d_{i,j,h}^o$ (with vector form \mathbf{d}^a and \mathbf{d}^o , respectively) are positive dummy variables. The Lagrangian function is defined as:

$$\mathcal{L}_c(\mathbf{\Pi}, \boldsymbol{\nu}_a, \boldsymbol{\nu}_o) = f(x) + \boldsymbol{\nu}_a^\top \mathcal{H}_a(\mathbf{\Pi}) + \boldsymbol{\nu}_o^\top \mathcal{H}_o(\mathbf{\Pi}), \quad (15)$$

where $\boldsymbol{\nu}_a$ and $\boldsymbol{\nu}_o$ are Lagrangian multipliers, \mathcal{H}_a and \mathcal{H}_o represent the equality constraints defined by (14a) and (14b), respectively. Specifically, \mathcal{H}_a corresponds to the agent collision avoidance constraints $(R^a)^2 - (\pi_i^h - \pi_j^h)^2 + d_{i,j,h}^a = 0, \forall i, j, i \neq j, \forall h$, and \mathcal{H}_o corresponds to the obstacle collision avoidance constraints $(R^o)^2 - (\pi_i^h - o_j)^2 + d_{i,j,h}^o = 0, \forall i, j, \forall h$. To improve the poor convergence of the classical lagrangian function, we can augment the Lagrangian function with a penalty on the constraint residuals (Boyd et al. 2011; Kotary, Fioretto, and Van Hentenryck 2022):

$$\begin{aligned} \mathcal{L}(\mathbf{\Pi}, \boldsymbol{\nu}_a, \boldsymbol{\nu}_o) = & f(x) \\ & + \boldsymbol{\nu}_a^\top \mathcal{H}_a(\mathbf{\Pi}) + \boldsymbol{\nu}_o^\top \mathcal{H}_o(\mathbf{\Pi}) \\ & + \rho_a \|\mathcal{H}_a(\mathbf{\Pi})\|^2 + \rho_o \|\mathcal{H}_o(\mathbf{\Pi})\|^2, \end{aligned} \quad (16)$$

where ρ_a and ρ_o are chosen penalty weights on the equality residuals. The corresponding Lagrangian Dual function can be defined:

$$\mathbf{d}(\boldsymbol{\nu}_a, \boldsymbol{\nu}_o) = \min_{\mathbf{\Pi}} \mathcal{L}(\mathbf{\Pi}, \boldsymbol{\nu}_a, \boldsymbol{\nu}_o). \quad (17)$$

The Lagrangian Dual Problem is to maximize the dual function:

$$\arg \max_{\boldsymbol{\nu}_a, \boldsymbol{\nu}_o} \mathbf{d}(\boldsymbol{\nu}_a, \boldsymbol{\nu}_o) \quad (18a)$$

$$\text{s.t. } \mathbf{\Pi} \in \Omega_c. \quad (18b)$$

Algorithm 2: ALM for Projection

Input: Tolerance δ , Weight ρ , Initial Trajectory $\hat{\Pi}$

- 1 $\mathbf{x}_T^0 \sim \mathcal{N}(\mathbf{0}, \sigma_T \mathbf{I})$
- 2 **while** $\nabla_{\nu_a} < \delta \wedge \nabla_{\nu_o} < \delta$ **do**
- 3 $\hat{\nu}_a \leftarrow \mathcal{H}_a(\hat{\Pi}), \hat{\nu}_o \leftarrow \mathcal{H}_o(\hat{\Pi});$
- 4 $\hat{\Pi} \leftarrow \arg \min_{\Pi \in \Omega_c} \mathcal{L}(\hat{\Pi}, \nu_a^*, \nu_o^*);$
- 5 $\nabla_{\nu_a} \leftarrow \mathcal{H}_a(\hat{\Pi}), \nabla_{\nu_o} \leftarrow \mathcal{H}_o(\hat{\Pi});$
- 6 $\rho \leftarrow \text{Update}(\rho)$
- 7 **return** $\hat{\Pi};$

Through weak duality, solving the dual problem (18) can provide a lower bound for the original problem’s optimal solution. Specifically, a feasible solution $\hat{\Pi}$ to the Primal problem can be derived from the dual solution (ν_a^*, ν_o^*) via the stationarity condition:

$$\hat{\Pi} = \arg \min_{\Pi \in \Omega_c} \mathcal{L}(\Pi, \nu_a^*, \nu_o^*). \quad (19a)$$

The dual problem (18) can be solved iteratively, named the Dual Ascent method (DAM):

$$\Pi^k = \arg \min_{\Pi \in \Omega_c} \mathcal{L}(\Pi, \nu_a^k, \nu_o^k), \quad (20a)$$

$$\nu_a^{k+1} = \nu_a^k + \rho_a^k \mathcal{H}_a(\Pi^k), \quad (20b)$$

$$\nu_o^{k+1} = \nu_o^k + \rho_o^k \mathcal{H}_o(\Pi^k). \quad (20c)$$

Using ALM significantly accelerates the projection process, especially in complex scenarios. The augmented sampling process is described in Algorithm 2.

Experiments

We evaluate the performance of PDM in generating feasible trajectories for MAPF in continuous spaces. We compare PDM against standard Diffusion Models (SDM) and Guided Diffusion Models (GDM) across three challenging scenarios: Narrow Corridors, Obstacle-Dense Environments, and Agent-Dense Environments.

Experimental Setup

We conduct experiments in the following scenarios:

- **Narrow Corridors:** Scenarios where agents must exchange positions in confined spaces, requiring precise coordination to avoid collisions.
- **Obstacle-Dense Environments:** Scenarios with a high density of obstacles, where agents must navigate complex paths to reach their goals without collisions.
- **Agent-Dense Environments:** Scenarios with a large number of agents, increasing the complexity of collision avoidance and coordination.

For each scenario, we generate environments where the positions of obstacles and agents are randomly assigned and do not appear in the training data, ensuring that the models are tested on unseen configurations. The training data is collected following the routine described in (Okumura et al. 2022b).

We evaluate the methods based on two metrics:

		PDM	DM	GDM
Violation	NC 1	0	34.62	0.96
	NC 2	0	15.79	5.26
Path	NC 1	0.7867	2.6766	0.8235
	NC 2	0.7521	2.1293	0.8398

Table 1: Performance Evaluation on Narrow Corridors.

- **Violation Rate:** The percentage of constraints violated, indicating the feasibility of the generated trajectories.
- **Total Path Length:** The sum of the lengths of the paths taken by all agents, reflecting the efficiency of the trajectories.

We compare our proposed PDM against the following baseline methods:

- **Standard Diffusion Models (SDM):** Standard diffusion models used to generate trajectories without any constraint handling.
- **Guided Diffusion Models (GDM):** Diffusion models guided by penalty terms added during the sampling process to encourage feasibility, similar to the method used in (Carvalho et al. 2023).

Evaluation on Narrow Corridors

The Narrow Corridor scenarios are designed to test the ability of the methods to generate feasible trajectories in tight spaces where agents must exchange positions. Figures 1(a) and 1(b) illustrate the trajectories generated by PDM in two different narrow corridor scenarios. Agents (solid circles) successfully reach their respective goals (empty circles) by coordinating their movements to avoid collisions in the confined space. Notice how PDM can identify a set of feasible paths for each agent in the narrow corridor by generating complex maneuvers and adjusting speed and position to allow an agent to overtake the other.

Table 1 presents the performance of all methods in terms of violation rate and total path length for the two narrow corridor scenarios (NC1 and NC2). Lower values indicate better performance. Notice how PDM outperforms both DM and GDM, *achieving zero violation rates and the shortest total path lengths in both scenarios*. In contrast, standard DM exhibits high violation rates and longer paths, indicating significant limitations in handling constraints. GDM reduces violation rates compared to DM but still falls short of PDM’s performance.

Evaluation on Obstacle-dense Scenarios

In the Obstacle-Dense scenarios, we test the methods in environments with twenty randomly placed obstacles and four agents. Figures 2(a) and 2(b) show the trajectories generated by PDM, demonstrating its ability to navigate complex environments while avoiding collisions even when agents need to navigate scenarios presenting a large number of obstacles.

Table 2 summarizes the performance of the methods in the obstacle-dense scenarios (OD1 and OD2). Notice how PDM

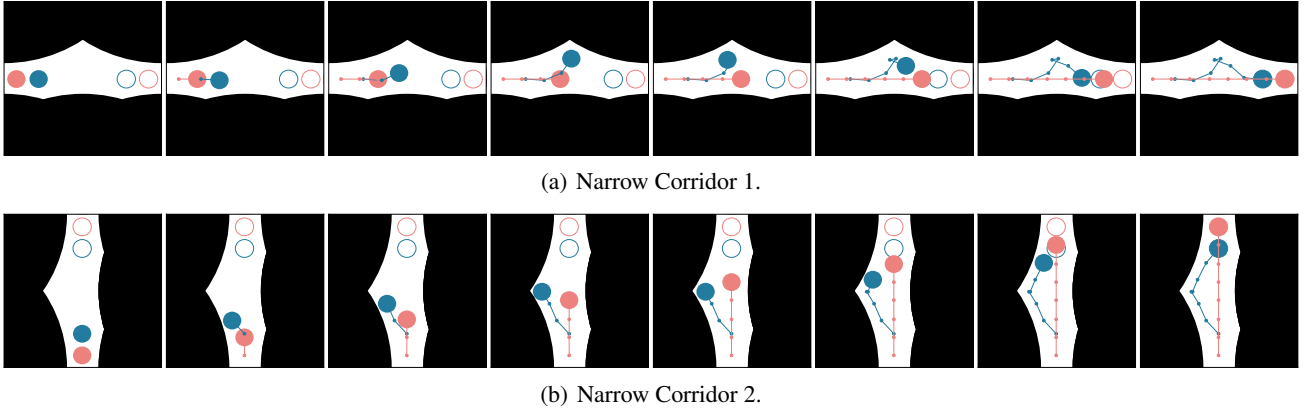


Figure 1: Collision-free trajectories generated by PDM in Narrow Corridor scenarios. Agents (solid circles) navigate to their goals (empty circles) by exchanging positions in confined spaces without collisions.

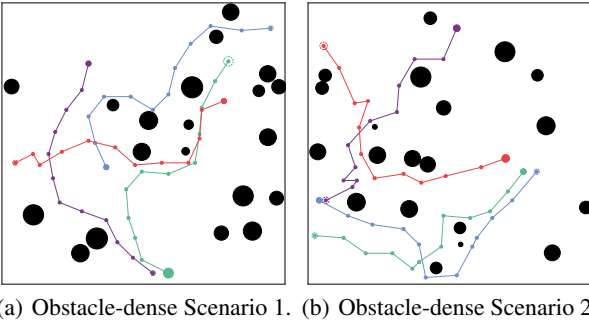


Figure 2: Collision-free trajectories generated by PDM in Obstacle-Dense scenarios. Agents successfully navigate through environments with numerous obstacles to reach their goals. The empty dashed circles denote starting points, and the solid circles represent the goals.

achieves the best performance, maintaining zero violation rates and the shortest paths, indicating strong adaptability to dense obstacles. In contrast, DM exhibits higher violation rates and the longest paths. GDM outperforms DM but is worse than PDM. These results emphasize PDM’s robustness and efficiency.

		PDM	DM	GDM
Violation	OS 1	0	0.58	0.48
Rate	OS 2	0	1.02	0.58
Path	OS 1	2.0087	6.0228	5.6979
Length	OS 2	2.0457	5.8585	5.2771

Table 2: Performance Evaluation on Obstacle-dense Scenarios.

		PDM	DM	GDM
Violation	AS 1	0.31	3.78	0.54
Rate	AS 2	0.17	2.99	0.37
Path	AS 1	5.2021	11.1599	5.4932
Length	AS 2	5.1631	11.4114	5.4081

Table 3: Performance Evaluation on Agent-dense Scenarios.

Evaluation on Agent-dense Scenarios

Finally, we test the ability of our proposed method to handle a large collection of agents. An increasing number of agents significantly introduces computational costs during projection, which makes standard projection methods challenging to handle. To address this, we use the ALM method to efficiently address Agent-dense Scenarios. Table 3 evaluates PDM, DM, and GDM in agent-dense scenarios (AS 1 and AS 2). PDM achieves the lowest violation rates (0.31 and 0.17) and shortest path lengths (5.2021 and 5.1631), highlighting its efficiency in handling high agent density. GDM also shows moderate performance, with higher violation rates and longer path lengths compared to PDM. DM performs the worst, with significantly higher violation rates (3.78 and 2.99) and longest paths (11.1599 and 11.4114), indicating limited suitability for agent-dense conditions.

These results are significant as they demonstrate the power of combining diffusion models with constrained optimization techniques to address problems that would be otherwise challenging to be tackled by these two areas independently.

Conclusion

In this paper, we have presented a novel approach that combines constrained optimization techniques with DMs to generate collision-free trajectories for MAPF in continuous spaces. By integrating constraints directly into the diffusion process, our method enables the direct generation of feasible solutions for MAPF without the need for expensive re-

jection sampling or post-processing steps. This integration ensures that the generated trajectories satisfy all necessary constraints, including collision avoidance between agents, adherence to kinematic limits, and compliance with start and goal positions.

To address the computational challenges inherent in handling complex constraints, especially in scenarios with a large number of agents or obstacles, we designed an ALM to efficiently manage the projection process within the diffusion framework. The ALM significantly accelerates the computation by transforming the constrained optimization problem into a series of unconstrained problems augmented with penalty terms and Lagrange multipliers. This enhancement makes our approach scalable and practical for real-world applications where computational resources and time are critical factors.

Our preliminary experiments across various challenging scenarios—including narrow corridors, obstacle-dense environments, and agent-dense environments—demonstrate the effectiveness and robustness of our proposed method. In narrow corridor scenarios, where precise coordination is crucial, our PDM successfully generated feasible trajectories that allowed agents to exchange positions without collisions. In obstacle-dense environments, PDM consistently navigated agents through complex paths while maintaining zero violation rates and optimizing path lengths. In agent-dense scenarios, despite the increased complexity due to the higher number of agents, PDM maintained superior performance with the lowest violation rates and shortest total path lengths.

Crucially, the integration of constrained optimization into the diffusion process not only ensures constraint satisfaction but also improves the overall quality of the generated trajectories. By embedding constraint handling directly within the generative model, the aim is to eliminate the reliance on heuristic adjustments, leading to a cohesive and effective solution. We hope that applications like this one would help to bridge the gap between probabilistic generative models and constrained optimization, opening new avenues for applying diffusion models to complex multi-agent robotic systems.

Acknowledgment

This research is partially supported by NSF grants 2334936, 2334448, and NSF CAREER Award 2401285. The authors acknowledge Research Computing at the University of Virginia for providing computational resources that have contributed to the results reported within this paper. The views and conclusions of this work are those of the authors only.

References

- Augugliaro, F.; Schoellig, A. P.; and D’Andrea, R. 2012. Generation of collision-free trajectories for a quadcopter fleet: A sequential convex programming approach. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1917–1922.
- Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; Eckstein, J.; et al. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1): 1–122.
- Carvalho, J.; Le, A. T.; Baierl, M.; Koert, D.; and Peters, J. 2023. Motion planning diffusion: Learning and planning of robot motions with diffusion models. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1916–1923. IEEE.
- Chen, R.; Liang, Z.; Cheng, J.; You, P.; and Yang, Z. 2023. Multi-Agent Cooperative Motion Planning Based on Alternating Direction Method of Multipliers. *IEEE Control Systems Letters*, 7: 3307–3312.
- Chen, Y.; Cutler, M.; and How, J. P. 2015. Decoupled multiagent path planning via incremental sequential convex programming. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 5954–5961.
- Christopher, J. K.; Baek, S.; and Fioretto, F. 2024. Constrained Synthesis with Projected Diffusion Models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.
- Hopcroft, J. E.; Schwartz, J. T.; and Sharir, M. 1984. On the Complexity of Motion Planning for Multiple Independent Objects; PSPACE-Hardness of the “Warehouseman’s Problem”. *The international journal of robotics research*, 3(4): 76–88.
- Janner, M.; Du, Y.; Tenenbaum, J. B.; and Levine, S. 2022. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*.
- Kavraki, L. E.; Svestka, P.; Latombe, J.-C.; and Overmars, M. H. 1996. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4): 566–580.
- Kotary, J.; Fioretto, F.; and Van Hentenryck, P. 2022. Fast approximations for job shop scheduling: A lagrangian dual deep learning method. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 7239–7246.
- LaValle, S. 1998. Rapidly-exploring random trees: A new tool for path planning. *Research Report 9811*.
- Li, J.; Felner, A.; Boyarski, E.; Ma, H.; and Koenig, S. 2019. Improved Heuristics for Multi-Agent Path Finding with Conflict-Based Search. In *IJCAI*, volume 2019, 442–449.
- Li, J.; Ruml, W.; and Koenig, S. 2021. Eecbs: A bounded-suboptimal search for multi-agent path finding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 12353–12362.
- Li, J.; Tinka, A.; Kiesel, S.; Durham, J. W.; Kumar, T. S.; and Koenig, S. 2021. Lifelong multi-agent path finding in large-scale warehouses. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 11272–11281.
- Okumura, K.; Machida, M.; Défago, X.; and Tamura, Y. 2022a. Priority inheritance with backtracking for iterative multi-agent path finding. *Artificial Intelligence*, 310: 103752.

Okumura, K.; Yonetani, R.; Nishimura, M.; and Kanezaki, A. 2022b. Ctrms: Learning to construct cooperative timed roadmaps for multi-agent path planning in continuous spaces. *arXiv preprint arXiv:2201.09467*.

Shaoul, Y.; Mishani, I.; Vats, S.; Li, J.; and Likhachev, M. 2024. Multi-Robot Motion Planning with Diffusion Models. *arXiv preprint arXiv:2410.03072*.

Song, Y.; and Ermon, S. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Stern, R.; Sturtevant, N.; Felner, A.; Koenig, S.; Ma, H.; Walker, T.; Li, J.; Atzmon, D.; Cohen, L.; Kumar, T.; et al. 2019. Multi-agent pathfinding: Definitions, variants, and benchmarks. In *Proceedings of the International Symposium on Combinatorial Search*, volume 10, 151–158.

Yang, L.; Zhang, Z.; Song, Y.; Hong, S.; Xu, R.; Zhao, Y.; Zhang, W.; Cui, B.; and Yang, M.-H. 2023. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4): 1–39.