2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Intermittent diffusion-based path planning for heterogeneous groups of mobile sensors in cluttered environments

Christina Frederick[1], Haomin Zhou[2], and Frank Crosby[3]

Abstract— **This paper presents a method for task-oriented path planning and collision avoidance for a group of heterogeneous holonomic mobile sensors. It is a generalization of the authors' prior work on diffusion-based path planning. The proposed variant allows one to plan paths in environments cluttered with obstacles. The agents follow flow dynamics, i.e., the negative gradient of a function that is the sum of two functions: the first minimizes the distance from desired target regions and the second captures distance from other agents within a field of view. When it becomes necessary to steer around an obstacle, this function is augmented by a projection term that is carefully designed in terms of obstacle boundaries. More importantly, a diffusion term is added intermittently so that agents can exit local minima. In addition, the new approach skips the offline planning phase in the prior approach to improve computational performance and handle collision avoidance with a completely decentralized method. This approach also provably finds collision-free paths under certain conditions. Numerical simulations of three deployment missions further support the performance of ID-based diffusion.**

I. Introduction

In the past decades, mobile sensor networks have emerged as a promising technology in a wide range of applications including automation in manufacturing [1], [2], environmental monitoring [3], [4], [5], target surveillance [6], [7], and threat detection [8]. They are considered ideal for many scenarios in which large groups of mobile sensors with limited computing and sensing abilities must accomplish complex tasks under harsh or hostile conditions, such as underwater surveillance or rescue missions in time-varying landscapes with cluttered regions and moving obstacles.

The success of mobile sensor networks depends on the design of autonomous deployment strategies that find feasible or optimal paths to mission-specific targets. The planning methods must take into consideration the performance features of individual sensors, such as sensing, communication, and onboard computation capabilities. Accounting for these factors in mobile sensor deployment, especially for large groups of sensors, is a highly advanced, large-scale system engineering problem.

In this paper, we consider the path planning problem in a cluttered environment for a large, heterogeneous group of low-functioning miniature mobile sensors. Heterogeneity, including different sizes, sensing radii, and functionalities, can significantly complicate the problem when the rules for a single sensor cannot be applied to another. While current methods have difficulties dealing with these issues level-set methods, such as those used in this work, simplify the problem. We assume that the sensors have severe performance limitations, such as short sensing and communication range, low battery supply, no path tracking ability, and weak onboard computing capability. With these limitations, the sensors must use local information to make decisions without the benefit of a centralized controller or coordination.

The method is based on intermittent diffusion (ID) based path planning [9], in which random perturbations are added intermittently to gradient flow dynamics so that sensors can move out of the local traps or break the deadlocks. We demonstrate that ID is an effective path planning algorithm for a large group of heterogeneous mobile sensors with low functionalities. It is also shown theoretically that the method can attain collision avoidance with provable convergence (accomplishing planning tasks) in obstacle-free environments.

The present study introduces increased difficulties in navigating cluttered environments while further restricting the capabilities of the mobile sensors. The main contributions of this paper include: 1) a new projection strategy that smoothly steers the sensors around obstacles; 2) incorporating the heterogeneity of mobile sensors in the model so that the paths for sensors with different features can be planned simultaneously; and 3) various new deployment missions that demonstrate effectiveness, collision avoidance, and target achievement.

A. Relevant work

Methods based on artificial potential field (APF) [10] and Model Predictive Control (MPC) [11] are capable path planning methods for groups of sensors. APF has a long history in path planning [12], [13], [14], [15], [16], [17], [18], [19], [20]. While offering desirable traits such as only using local information and scalability, APF suffers the limitation of the sensors getting trapped at local minima or forming deadlocks.

In decentralized MPC [21] and its variants, each sensor computes a path by solving sequential subproblems while cooperating with neighbors via a global objective based on other sensors' decisions in a short time horizon. This is different from the present setting, in which each sensor is not pre-assigned a target position and the paths for all sensors are computed simultaneously to achieve a collective objective using only local information.

*This work was supported in part by ONR N00014-21-1-2856 and N00014-21-1-2891.

[1]Christina Frederick is with Department of Mathematical Sciences, NJIT, Newark, NJ 07102, USA christin@njit.edu

[2]Haomin Zhou is with the Department of Mathematics, Georgia Tech, Atlanta, GA USA hmzhou@math.gatech.edu

[3]Frank Crosby is with NSWC PC, Florida USA frank.j.crosby.civ@us.navy.mil

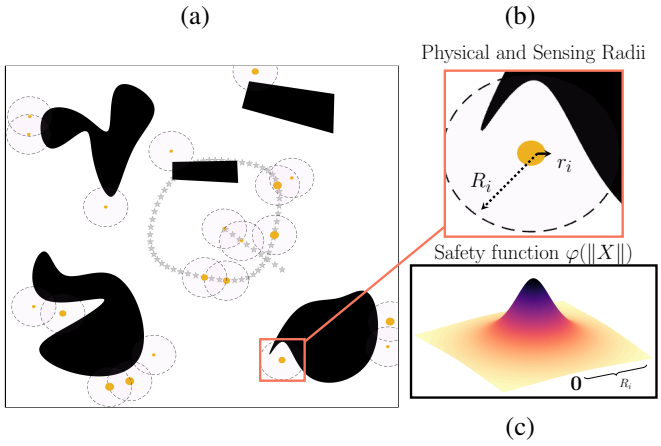


Fig. 1: (a) A heterogeneous group of mobile sensors must navigate an environment containing obstacles (black) and form a target distribution (diamonds). (b) The sensing regions are shaded in pink around each agent. (c) A safety function $\varphi(\|X\|)$ is used to ensure collision avoidance in (5).

In addition to APF and MPC, there is a rich literature for path planning algorithms, which include the family of Bugs algorithms [22], [23], the graph-based algorithms such as A, A*, D, and D* [24], [25], [26], [27], [28], the randomness involved strategies including Probability Roadmap (PRM) [29], [30], [31], and Rapidly-exploring Random Tree (RRT) [32], [33], [34], just to name a few. These were initially designed for single-agent path planning, and many have been extended to multi-agent systems with remarkable success [35], [36], [37], [38], [39], [40], [41], [42], [43], [44]. Applying these methods to large groups of mobile sensors can still be challenging, both theoretically and computationally.

Density control is another relevant tool used in planning tasks for groups of mobile sensors. Methods based on optimal transport [45], [46] or the Schrödinger bridge problem [47], [48] have been proposed in recent years. Density control formulations have been used in applications, e.g. the new architecture for space telescopes proposed in [49]. In the existing literature on this line of research, collision avoidance, which is critical in mobile sensor deployments, is not considered.

B. Outline

The paper is organized as follows: The problem setup is presented in Section II. The proposed method and numerical simulations are given in Sections III and IV. The paper is concluded with a discussion in Section V.

II. Problem Formulation

We consider a heterogeneous fleet of mobile sensors (called agents) navigating in an unbounded, cluttered, planar environment. Let $X^i(t)$, $1 \leq i \leq N$ denote the (x, y) coordinates of the i^{th} agent at a global time $t \geq 0$. Assume that the collision space of the i^{th} agent is $X^i(t) + B(r_i)$, where $B(r_i)$ is the circle of radius r_i, i.e., $B(r_i) = \{z \in \mathbb{R}^2, \|z\| \leq r_i\}$. Each agent is equipped with an onboard, omnidirectional, range-limited sensing device used to detect neighboring agents and obstacles. The sensing region of the i^{th} agent is given by $X^i(t) + B(R_i)$ for $R_i > 2r_i$.

The agents are assigned a collective task of covering desired regions in the environment, denoted by $\Gamma \subset \mathbb{R}^2$. These regions can be modeled by filled shapes (2D), curves (1D), or separated points (0D). Obstacles are also modeled as a union of disjoint regions, denoted by $\Gamma_O \subset \mathbb{R}^2$.

We assume that the path planning problem is well-posed, that is, there exist N collision-free trajectories $X^i(t)$, $0 \leq t \leq T$ that have a feasible initial configuration, avoid obstacles, and reach the target regions in finite time, i.e., $\{X^i(T)\}_{i=1}^N \subset \Gamma$. A feasible configuration implies that the agents are sufficiently far from obstacles and each other, $(X^i)_{i=1}^N \in \mathcal{X}$, where

$$\mathcal{X} = \{(X^i)_{i=1}^N \mid X^i \notin \Gamma_O, \\ \|X^i - X^j\| > 2\max\{r_i, r_j\}\, \forall j \neq i\}. \quad (1)$$

We aim to develop a path planning algorithm with the following properties: (O1) collisions do not occur; (O2) the majority of the agents are well-distributed in the target region in finite time, even if there are not enough or too many sensors; (O3) the agents navigate around obstacles, remaining close to the obstacle boundaries (instead of being repelled with the mechanism used for collision avoidance) (O4) the algorithm scales to large groups of sensors and is stable with respect to uncertainty in precise locations.

III. Method

The centralized approach we developed in [9] fulfills (O1)-(O2) and (O4) using ID-based path planning. In obstacle-free environments, this strategy guarantees asymptotic convergence and collision avoidance. Here, we take a decentralized approach in a more complicated, cluttered environment.

A. Assumptions

The following minimal assumptions are made:

- A1) Each agent can detect the positions of other agents and obstacles within its sensing region.
- A2) No transfer of knowledge occurs between the agents e.g., obstacles that are outside of an agent's sensing region are unknown even when they are detected by neighbors.
- A3) Agents can move in any direction (holonomic).
- A4) Agents are not pre-assigned target locations. The collective goal is accomplished when agents are well-separated and their distribution covers the target region.

B. Intermittent diffusion-based path planning

The path planning problem is formulated as a gradient flow (GF) with ID that minimizes a potential $\Psi(X)$, which will be defined precisely later, i.e.

$$\frac{dX(t)}{dt} = -(\nabla\Psi(X(t))). \quad (2)$$

To avoid local minima and speed up convergence, we intermittently add random perturbations to (2) as in [50], [9], leading to the following stochastic differential equation,

$$dY(t) = -(\nabla\Psi(Y(t)))dt + \sigma(t)dW(t), \qquad t > 0, \quad (3)$$

where $W(t)$ is a realization of a standard Brownian motion. More precisely we use a piecewise constant diffusion term $\sigma(t)$ that alternates between zero and a positive value, i.e.

$$\sigma(t) = \begin{cases} 0 & \text{if } t \in [S_k, T_k) \\ \sigma_k & \text{if } t \in [T_{k-1}, S_k]. \end{cases} \quad (4)$$

Here we partition the time interval $[0, T]$ as $\cup_{k=1}^{K}[T_{k-1}, T_k]$ with $T_0 = 0$, $T_K = T$ and $S_k \in [T_{k-1}, T_k]$. There are four positive ID parameters $\alpha_0 < \alpha$, and $\beta_0 < \beta$ that determine the strength and duration of the diffusion segments. The values σ_k are selected from a uniform distribution on $[\alpha_0, \alpha]$, and the intervals are chosen to have random lengths, that is $S_k - T_{k-1}$ is determined from a uniform distribution on $[\beta_0, \beta]$, where β_0 is a small, positive number and $\beta_0 \ll \beta$. The choice of ID parameters $\alpha_0, \alpha, \beta_0$, and β must be calibrated to ensure that agents take sufficiently bounded steps with high probability.

Figures 3−5 demonstrate the two stages of the algorithm: (b) the gradient flow stage ($\sigma = 0$) and (c) the diffusion stage ($\sigma > 0$). In both cases, the dominant driving mechanism is the potential function Ψ.

C. Path planner

Each of the paths $\{X^i(t), 0 \le t \le T, 1 \le i \le N\}$, is computed independently by path planners that are private to each agent. Each planner has three components:

1) A target objective function $F(X)$ that is minimized for $X \in \Gamma$, e.g.,

$$F(X) = d(X, \Gamma) := \min_{\tilde{X} \subset \Gamma} \|X - \tilde{X}\|.$$

2) A safety objective $G(X; R)$ that limits undesirable behavior (e.g., collisions with other agents), e.g.,

$$G(X; R) = G_0 \sum_{\substack{1 \le j \le N \\ 0 < \|X - X^j\| < R}} \varphi(\|X - X^j\|/2). \quad (5)$$

Here, $G(X; R)$ applies a smooth penalty φ within a sensing radius of R, and ϕ can be a user-specified monotonically decrease function vanishing at the sensing radius R. Fig. 1 shows a plot of an example φ.

3) To steer around obstacles, the greedy direction $-\nabla\Psi(X) := -\nabla(F(X) + G(X; R))$ is replaced by the modified direction $\Phi(X)$:

$$\Phi = \lambda(-\nabla\Psi) + (1 - \lambda)\widehat{\Pi}_O. \quad (6)$$

Details are given in Section III-E.

D. Algorithm

The ID-based path planning algorithm sketched in Algorithm 1 realizes each agent's high-level control loop to compute its path based on the planner described in the previous section. At each iteration, each agent updates its state $X^i(t)$, locations of neighbors, $\mathcal{N}(X^i) = \{X^j \mid d(X^i, X^j) < R_i, 1 \le j \le N, j \ne i\}$, and information about obstacle locations within the sensing region $\Gamma_O \cap (X^i + B(R_i))$ and computes the following quantities: $-\nabla\Psi(X^i) = -\nabla(F(X^i) + G(X^i; R_i))$, and when necessary, the obstacle steering direction using $\Phi(X^i)$ and $F(X^i)$.

Algorithm 1 Each agent's planning algorithm

Input: Agent location X, neighbors $\mathcal{N}(X)$, local obstacle $\Gamma_O \cap X + B(R)$

Init: $n = 0$, feasible initial configuration X_0, $K = 1$

1: **(GF) Gradient flow toward global target Γ:**
Calculate Φ by (6)

$$X_{n+1} = X_n - \Phi\Delta t$$
$$X_{opt} = X_n \text{ if } \Psi(X_n) < \Psi(X_{opt})$$

$n_K := n, K = K + 1$.

2: **(D) Diffusion:** Generate ID quantities ξ and M.

$$X_{n+1} = X_n - \nabla\Psi(X_n)\Delta t + \sigma\xi\sqrt{\Delta t}$$

$n = n_K, n_K + 1, \ldots, n_K + M$.

3: Repeat (GF) and (D) until $\Psi(X_{opt}) < tol$.

4: Perform the iterations in (GF) until $n > n_K + n_T$.

The iterations in (GF) terminate when stopping criteria are satisfied at $n = n_K$. Then, in (D), M additional iterations are performed. After a global tolerance is met, a fixed number of iterations, denoted by n_T, are performed in the final stage to allow the agents to "settle down" into terminal locations.

E. Obstacle steering

In cluttered environments, the ID-based path planning strategy guides each agent to follow the trajectory

$$dX(t) = \Phi(X(t))dt + \sigma(X(t), t)dW(t) \qquad t > 0, \quad (7)$$

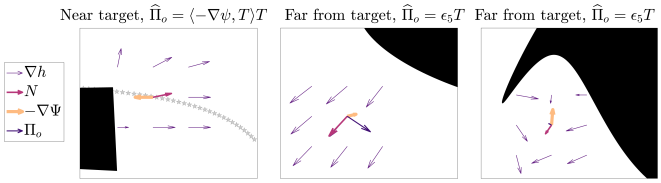
where Φ in (7) depends on the shape of the obstacle boundary and the proximity of the agent to both target regions (indicated by $F(X)$) and obstacle regions (indicated by $h(X) := d(X, \Gamma_O)$).

There are three main features of the obstacle avoidance strategy:

1) Smooth, continuous steering around obstacles;
2) Singularity avoidance near corners; and
3) Projections onto obstacle boundaries.

We elaborate on our rationale for these features below.

1) Smooth steering: To smoothly steer around obstacles, we define a weighting (6) that balances the original direction $-\nabla\Psi$ (ignoring the obstacle) with a modified direction $\widehat{\Pi}_O$ that steers the agents away from the obstacle. The value of



(a) (b) (c)

Fig. 2: Local gradients are used to obtain the vector η in the obstacle steering strategy (9).

the weight λ depends on the proximity of the agent to the obstacle, that is, $\lambda = \lambda(h(X))$ where

$$\lambda(H) = \begin{cases} 0 & H < \epsilon_1 \\ \frac{1}{2}(\tanh(\gamma)+1) & \epsilon_1 \le H \le \epsilon_2 \\ 1 & H > \epsilon_2 \end{cases} \tag{8}$$

where $\gamma = \frac{\epsilon_2-\epsilon_1}{2}(\frac{1}{\epsilon_2-H} - \frac{1}{H-\epsilon_1})$.

The constants $\epsilon_1 < \epsilon_2$ create a buffer region around the obstacle. Closer to the obstacle, when h is smaller, the steering is more aggressively geared in the direction tangent to the obstacle boundary. Farther from the obstacle, the steering is milder. Since λ is a smooth function, the steering is continuous. This means that the weighting in (6) favors the original direction when the agents are further from the obstacle and favor the projected direction $\widehat{\Pi}_O$ closer to the obstacle, ensuring a smooth transition.

2) Singularity avoidance: One issue with steering along obstacle boundaries occurs when the boundary is irregular, causing singularities in the gradient. To address this, we use an averaging technique for calculating the vector normal to the obstacle boundary. We define the (counter-clockwise) tangent and the averaged normal vectors in terms of the gradient of the level-set function $h(X)$:

$$\eta = \frac{1}{m}(\sum_{k=1}^{m} \nabla h(X + \epsilon_4 e_j)), \qquad \tau = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \eta.$$

The average is taken over a subset of the 3×3 grid of nearby locations centered at X with spacing $0 < \epsilon_4 < \min_{1\le i\le N} R_i$. The vectors used for averaging are $e_0 = 0$ and the $m \le 9$ unit vectors $\{e_j\}_{j=1}^m$ satisfying $h(X + e_j) < \epsilon_2$. The well-defined tangent vector τ is defined via a matrix transformation of the averaged normal vector in the counterclockwise direction. The averaging smooths out singularities in the normal derivatives (see Figure 2). We emphasize that we do not re-scale the averaged normal or tangent vectors in these computations. The two vectors η and τ are used in the boundary projections described next.

3) Obstacle boundary projections: The design of the 'steered' direction $\widehat{\Pi}_O$ accounts for the proximity of the agent location X to the target shape Γ indicated by $F(X)$ and the angles between the gradient flow direction $-\nabla\Psi$ and the averaged normal η and tangent τ vectors.

The main idea is that when the agent is close to the obstacle, it should be 'steered' along a counter-clockwise direction tangent to the obstacle boundary. The angular direction must be fixed (clockwise or counter-clockwise) to avoid back-and-forth motions near concavities.

The steering mechanism is defined close to the obstacle ($h < \epsilon_2$), given $\mu_\eta = \langle \eta, -\nabla\Psi\rangle$ and $\mu_\tau = \langle \tau, -\nabla\Psi\rangle$, as

$$\widehat{\Pi}_O = \begin{cases} \mu_\tau \tau & F < \epsilon_3, \quad \mu_\eta < 0, \\ \epsilon_5 \tau & F \ge \epsilon_3, \quad \mu_\eta \le \epsilon_4 \|\eta\| \|\nabla\Psi\| \end{cases} \tag{9}$$

Figure 2 depicts this steering strategy in three cases. In Figure 2(a) the agent is located near the target region ($F < \epsilon_3$) and the descent direction (orange) is pointed toward the obstacle ($\mu_\eta < 0$) and almost orthogonal to τ. Setting $\widehat{\Pi}_O = \mu_\tau \tau$ prevents the agent from moving since $\mu_\tau \simeq 0$. Two situations occur when the agent is far from the target ($F \ge \epsilon_3$) in which the steering vector has a fixed length $\widehat{\Pi}_O = \epsilon_5 \tau$. In 2(b) ($\mu_\eta < 0$) and 2(c) ($0 \le \mu_\eta \le \epsilon_4 \|\eta\| \|\nabla\Psi\|$), the descent direction is again pointed at the obstacle, but in 2(c) the averaging within the concavity used to calculate η affects the length of τ, and therefore $\widehat{\Pi}_O$ is small. This prevents the agent from moving further into the concavity. The added diffusion in the planning algorithm (3) avoids stagnation.

F. Complexity

ID-based path planning algorithms are well-suited for problems involving a large number of agents. Algorithm 1 has a local complexity that is linear in the number of agents. There is a bounded number of computations performed by each path planner. The most costly task in each iteration is computing the sum φ in (5) of a bounded number of terms depending on the maximum number of agents that can be sensed within a fixed sensing radius. The steering (6) involves the computation of the vector η that is a sum of up to nine values, the calculation of the quantities $\|\eta\|$, $\|\nabla\Psi\|$ and the inner products μ_τ and μ_η.

The global complexity $O(n_T N)$ is linear in the number of iterations n_T, meaning that the algorithm can easily be scaled.

G. Safety

Sufficiently far from obstacles, the agents are provably collision-free in the GF stage of Algorithm 1.

Theorem 1: Suppose $d(X^i, \Gamma_O) > \epsilon_2 + r_i$ for $1 \le i \le N$. The trajectories generated by (2) with feasible initial configuration $(X^i(0))_i \in \mathcal{X}$, the agents do not collide for all $t > 0$.

Proof: The function $Y(t) = \sum_{i=1}^N \Psi(X^i(t))$ is non-increasing since it satisfies

$$\frac{dY}{dt} = \sum_{i=1}^{N} (\nabla\Psi(X^i)) \cdot \frac{dX^i}{dt} = -\sum_{i=1}^{N} \|(\nabla\Psi(X^i(t)))\|^2 \le 0.$$

Assume there is a time $t^* > 0$ such that $\|X^i(t^*) - X^j(t^*)\|^2 \leq \max_{r\in\{r_i,r_j\}} r^2$ for some $i, j \neq i$, then

$$\begin{aligned} Y(t^*) &= \sum_{i=1}^{N} F(X^i(t^*)) + G(X^i(t^*)) \\ &\geq \max_{r\in\{r_i,r_j\}} G_0\varphi(r) \\ &> E_0 := Y(0). \end{aligned}$$

Here we used that F is non-negative by construction. This is a contradiction, because $\Psi(X(t))$ is non-increasing, so we must have $\Psi(X(t^*)) \leq E_0$. ∎

In the diffusion stage, the agents have a positive but low chance of collision. In the numerical experiments, we did not observe collisions for appropriately calibrated parameters α and Δt. Specifically, the movement in each time step is bounded by $\|\sigma\xi_n\sqrt{\Delta t}\|$, where $\sigma < \alpha_2$. The only unbounded term is ξ_n, which has a low probability of taking large values. Therefore, in regions sufficiently far from obstacles, the chance of collisions is low. This is discussed further in Section V.

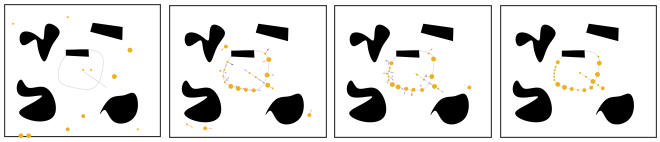
H. Stability

IV. Numerical Experiments

We demonstrate Algorithm 1 in three experiments in a computational domain $D \subset \mathbb{R}^2$ containing a combination of smooth and polygonal obstacles. The agents in the group have physical radii in the set $\mathcal{P}$ and fixed sensing radius R.

A. Continuous curved target region

Here, $N = 20$, $D = [-6, 6] \times [-6, 6]$, $\mathcal{P} = \{0.04, 0.08, 0.1\}$ and $R = .5$. Figure 3 demonstrates the method in an environment consisting of a curved target region and obstacles that have sharp corners or concavities that typically hinder gradient-based methods near singularities.

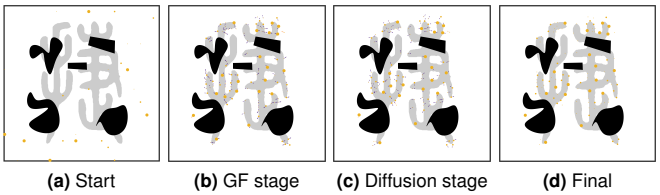


(a) Start **(b)** GF stage **(c)** Diffusion stage **(d)** Final

Fig. 3: Stages of ID-based path planning in an environment containing smooth and polygonal obstacles and targets forming a 'Q' shape.

B. Filled, disjoint target regions

Here $N = 100$, $D = [-6, 6] \times [-6, 6]$, $\mathcal{P} = \{0.02, 0.05, 0.1\}$ and $R = 1$. In this region the obstacles are the same as in the previous experience and the target region is the Chinese character '捷' consisting of two disjoint subregions forming the characters '扌' and '走'. Major difficulties arise from the separation between the two subregions. If a disparate sized group of agents is initialized on one side of the domain and one of the two subregions becomes overpopulated, path planning methods are faced with the possibility of stagnation and deadlocks. In these numerical experiments, highlighted in Figure 4, we found that the diffusion introduced by the ID algorithm enabled some of the agents to leave one subregion and move to a less populated subregion. In the final iteration, shown in Figure 4(d), one agent is located far from the target because it was hindered by the concavity of a the smooth obstacle. If the number of iterations is increased, the steering introduced will eventually overcome this with high probability.

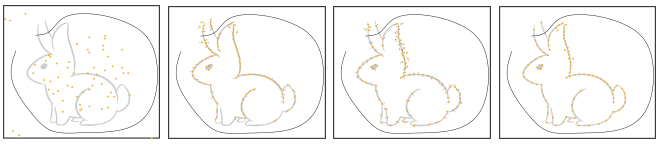


(a) Start **(b)** GF stage **(c)** Diffusion stage **(d)** Final

Fig. 4: Stages of ID-based path planning in an environment with targets forming the Chinese character '捷'.

C. Continuous curved obstacle region

Here, $N = 90$, $D = [-10, 10] \times [-10, 10]$, $\mathcal{P} = \{.05\}$, and $R = 1$. This experiment tested how well the obstacle steering strategy described in III-E performed in the case of a smooth, curved obstacle that almost completely encases the target region, posing the challenge of overcoming local minima (Figure 5).

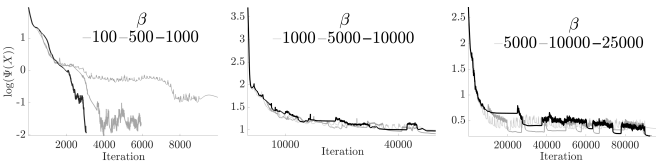


(a) Start **(b)** GF stage **(c)** Diffusion stage **(d)** Final

Fig. 5: Stages of ID-based path planning for a homogeneous group of agents in an environment containing a smooth, curved obstacle

D. Convergence and success

Figure 6 shows log plots of the energy functional Ψ in the three experiments described above. The plots indicate that in all three experiments, the successful convergence of



(a) Q **(b)** 捷 **(c)** Rabbit

Fig. 6: Energy vs. iteration for varying diffusion level β.

Shape	N	# classes	Success (β_1)	Success (β_2)	Success (β_3)
Q	20	3	95%	90%	100%
捷	100	3	86%	83%	85%
Rabbit	90	1	84%	87%	88%

TABLE I: The percent of agents reaching targets using ID-based path planning. There are three target shapes, N agents each belonging to a class corresponding to one of a finite number of possible physical radii. The duration of the diffusion segment, based on β_m, $1 \leq m \leq 3$ is varied.

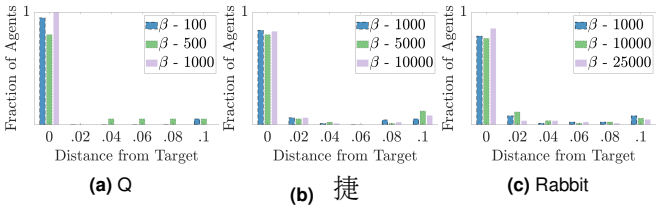


Fig. 7: Assessment of the terminal distribution.

the method seems to be independent of the choice of the ID parameter β. In Experiment 1 ('Q'), a smaller choice of β results in a longer simulation time. In Experiments 2 and 3 (' 捷 ' and 'Rabbit'), there is a fast reduction in energy. Also, the choice of β between 5000 and 10000 in these experiments does not seem to significantly affect the rate of convergence. This gives a sense of the calibration requirements of the ID parameters in the method. It is clear from the value of the energy functional that the agents never collide in any of the experiments.

Our results indicate that the minimization of Ψ accomplishes (O1)-(O4), i.e., the collision avoidance represented by G does not interfere with the mission. Another way to define agent success is the percentage of agents whose final location $X^i(T)$ satisfies $F(X^i(T)) = 0$ (Table I). In Experiment 1, the success rate attains 100% for large values of β. In Experiments 2 and 3, the influence of β is not as clear, possibly due to the increased sophistication of the geometries as well as the larger group size that produced more conflicts between neighboring agents.

A clearer picture of the success of the mission can be obtained by assessing the proximity of agents to target regions as measured by $F(X^i(T))$, shown in Figure 7. With more iterations, we expect a higher success rate.

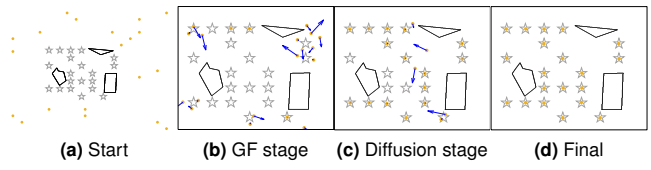


Fig. 8: Stages of ID-based path planning in an environment with discrete targets and modified stopping conditions.

E. Other stopping criteria

Algorithm 1 can be adapted with minimal modifications to accommodate sensor deployment tasks that require agents to "anchor" in specific locations. This can be accomplished by revising the stopping criteria in the algorithm, namely, enforcing that each sensor stops when reaching a target, and updating the target region for the remaining agents to omit the targets that are achieved. We tested these modifications in a simple setup, shown in Figure 8. The gray stars indicate the targets, and the agents are indicated in gold. In Figure 8(a), the sensors are randomly initialized, in Figure 8(b) they follow a gradient flow toward the targets, and in Figure 8(c) they follow a modified gradient flow with added diffusion with strength and duration given by the ID parameters (α, β). The targets are achieved in Figure 8(d).

V. Discussion

There are three major differences between Algorithm 1 and the original ID-based path planning algorithm introduced in [9]: 1) obstacles are introduced to the environment; 2) heterogeneity of the group; and most importantly 3) we decided to forgo the virtual diffusion stage in the original formulation in to eliminate the need for a centralized controller. In [9], it is assumed that the diffusion stage is performed virtually to produce intermediate locations, and then a gradient flow-type segment is used to move the agents to these locations. Even if there were collisions or blowups in the virtual stage, the actualized movements in this second gradient flow stage would be stable and collision free. However, this virtual diffusion demands communications, which is not assumed in the sensors considered here. It may also require coordinated calculation, which can be beyond the onboard computational power of each agent.

Convergence and collision avoidance can be guaranteed in obstacle-free environments [9], however, the proofs, based on optimal transport theory, cannot be easily extended to the present setup. The problem is indeed a difficult situation for any existing multi-agent path planning method. Our challenge in proving convergence arises from the obstacle avoidance strategy. Near an obstacle, optimal transport tools are no longer applicable, and a completely new theory is needed. On the other hand, when sufficiently far away from obstacles we can prove collision avoidance because the gradient flow dynamics behave similarly to the dynamics in the obstacle-free case. Near obstacles, the uniqueness of solutions to ordinary differential equations suggests the trajectories do not intersect, but there is no theoretical guarantee that they stay sufficiently far apart. Despite the incomplete supporting theory, our numerical experiments, including some "stress tests" designed to expose vulnerabilities that could hinder convergence or collision avoidance, indicate both convergence and collision avoidance for carefully calibrated simulation parameters. In future work, it is desirable to find practical strategies on how to translate the actual mobile sensor specifications and environmental characteristics, such as buffer distance and curvature of the obstacle boundaries, to the tuning parameters in the algorithm.

REFERENCES

[1] F. Franceschini, M. Galetto, D. Maisano, and L. Mastrogiacomo, "A review of localization algorithms for distributed wireless sensor networks in manufacturing," *International journal of computer integrated manufacturing*, vol. 22, no. 7, pp. 698–716, 2009.

[2] J. Mirabel, F. Lamiraux, T. L. Ha, A. Nicolin, O. Stasse, and S. Boria, "Performing manufacturing tasks with a mobile manipulator: from motion planning to sensor based motion control," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2021, pp. 159–164.

[3] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman, "Decentralized environmental modeling by mobile sensor networks," *IEEE Transactions on Robotics*, vol. 24, no. 3, pp. 710–724, 2008.

[4] G. Xu, W. Shen, and X. Wang, "Applications of wireless sensor networks in marine environment monitoring: A survey," *Sensors*, vol. 14, no. 9, pp. 16 932–16 954, 2014.

[5] S. L. Ullo and G. R. Sinha, "Advances in smart environment monitoring systems using iot and sensors," *Sensors*, vol. 20, no. 11, p. 3113, 2020.

[6] Z. Tang and U. Ozguner, "Motion planning for multitarget surveillance with mobile sensor agents," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 898–908, 2005.

[7] S. H. Semnani and O. A. Basir, "Semi-flocking algorithm for motion control of mobile sensors in large-scale surveillance systems," *IEEE Transactions on Cybernetics*, vol. 45, no. 1, pp. 129–137, 2014.

[8] A. H. Liu, J. J. Bunn, and K. M. Chandy, "Sensor networks for the detection and tracking of radiation and other threats in cities," in *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*. IEEE, 2011, pp. 1–12.

[9] C. Frederick, M. Egerstedt, and H. Zhou, "Collective motion planning for a group of robots using intermittent diffusion," *Journal of Scientific Computing*, vol. 90, no. 1, pp. 1–20, 2022.

[10] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots." *International Journal of Robotics Research*, vol. 5, no. 1, p. 90, 1986.

[11] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer Science & Business Media, 2013.

[12] B. Chanclou and A. Luciani, "Global and local path planning in natural environment by physical modeling," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS '96*, vol. 3. IEEE, pp. 1118–1125. [Online]. Available: http://ieeexplore.ieee.org/document/568960/

[13] C. Warren, "Global path planning using artificial potential fields," in *Proceedings, 1989 International Conference on Robotics and Automation*. IEEE Comput. Soc. Press, pp. 316–321. [Online]. Available: http://ieeexplore.ieee.org/document/100007/

[14] E. Rimon and D. E. Koditschek, "Exact Robot Navigation using Artificial Potential Functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992. [Online]. Available: http://ieeexplore.ieee.org/document/163777/

[15] S. S. Ge and Y. J. Cui, "Dynamic motion planning for mobile robots using potential field method," *Autonomous Robots*, vol. 13, no. 3, pp. 207–222, 2002.

[16] T. Zhao, H. Li, and S. Dian, "Multi-robot path planning based on improved artificial potential field and fuzzy inference system," *Journal of Intelligent & Fuzzy Systems*, vol. 39, no. 5, pp. 7621–7637, 2020.

[17] M. G. Park, J. H. Jeon, and M. C. Lee, "Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing," *Proceedings. ISIE 2001. IEEE International Symposium on Industrial Electronics, 2001.*, vol. 3, pp. 1530–1535, 2001. [Online]. Available: http://ieeexplore.ieee.org/document/931933/

[18] C. Warren, "Multiple robot path coordination using artificial potential fields," in *Proceedings., IEEE International Conference on Robotics and Automation*. IEEE Comput. Soc. Press, 1990, pp. 500–505. [Online]. Available: http://ieeexplore.ieee.org/document/126028/

[19] Min Cheol Lee and Min Gyu Park, "Artificial potential field based path planning for mobile robots using a virtual obstacle concept," *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*, vol. 2, no. Aim, pp. 735–740, 2003. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1225434

[20] H. Gossain, B. Sharma, R. Jain, and J. Garg, "Multi robot environment exploration using swarm," *AI and IoT for Smart City Applications*, pp. 171–183, 2022.

[21] Y. Kuwata and J. P. How, "Cooperative distributed robust trajectory optimization using receding horizon milp," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 2, pp. 423–431, 2010.

[22] V. Lumelsky and A. Stepanov, "Dynamic path planning for a mobile automaton with limited information on the environment," *IEEE Transactions on Automatic Control*, vol. 31, no. 11, pp. 1058–1063, 1986.

[23] I. Kamon, E. Rivlin, and E. Rimon, "A new range-sensor based globally convergent navigation algorithm for mobile robots," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1. IEEE, 1996, pp. 429–435.

[24] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[25] D. Ferguson, M. Likhachev, and A. Stentz, "A guide to heuristic based path planning," in *in: Proceedings of the Workshop on Planning under Uncertainty for Autonomous Systems at The International Conference on Automated Planning and Scheduling (ICAPS*, 2005.

[26] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong Planning A*," *Artificial Intelligence*, vol. 155, no. 1-2, pp. 93–146, may 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S000437020300225X

[27] A. Stentz, "The focussed d* algorithm for real-time replanning," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 1652–1659. [Online]. Available: http://dl.acm.org/citation.cfm?id=1643031.1643113

[28] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun, "Anytime search in dynamic graphs," *Artificial Intelligence*, vol. 172, no. 14, pp. 1613 – 1643, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S000437020800060X

[29] M. H. Overmars, "A random approach to motion planning," Dept. Computer Science, Utrecht Univ, Utrecht, The Netherlands, Tech. Rep. RUU-CS-92-32, October 1992.

[30] L. Kavraki and J.-C. Latombe, "Randomized preprocessing of configuration for fast path planning," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, vol. 3, 1994, pp. 2138–2145.

[31] N. Amato and Y. Wu, "A randomized roadmap method for path and manipulation planning," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1, 1996, pp. 113–120 vol.1.

[32] S. M. LaValle *et al.*, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[33] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, jul 1998.

[34] C. Park, J. Pan, and D. Manocha, "Real-time optimization-based planning in dynamic environments using GPUs," in *Proceedings - IEEE International Conference on Robotics and Automation*. IEEE, may 2013, pp. 4090–4097. [Online]. Available: http://ieeexplore.ieee.org/document/6631154/

[35] B. G. Anderson, E. Loeser, M. Gee, F. Ren, S. Biswas, O. Turanova, M. Haberland, and A. L. Bertozzi, "Quantitative Assessment of Robotic Swarm Coverage," in *Proc. 15th Int. Conf. on Informatics in Control, Automation, and Robotics*, 2018, pp. 91–101.

[36] Á. Madridano, A. Al-Kaff, D. Martín, and A. de la Escalera, "Trajectory planning for multi-robot systems: Methods and applications," *Expert Systems with Applications*, vol. 173, p. 114660, 2021.

[37] M. Sauer, A. Dachsberger, L. Giglhuber, and L. Zalewski, "Decentralized deadlock prevention for self-organizing industrial mobile robot fleets," in *2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*. IEEE, 2022, pp. 1–6.

[38] S. G. Lee, Y. Diaz-Mercado, and M. Egerstedt, "Multirobot Control Using Time-Varying Density Functions," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 489–493, apr 2015.

[39] K. Karur, N. Sharma, C. Dharmatti, and J. E. Siegel, "A survey of path planning algorithms for mobile robots," *Vehicles*, vol. 3, no. 3, pp. 448–468, 2021.

[40] R. Luna and K. E. Bekris, "Efficient and complete centralized multi-robot path planning," in *IEEE International Conference on Intelligent Robots and Systems*, 2011, pp. 3268–3275.

[41] N. Agmon, C.-L. Fok, Y. Emaliah, P. Stone, C. Julien, and S. Vishwanath, "On coordination in practical multi-robot patrol," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 650–656.

[42] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *International Journal of Advanced Robotic Systems*, vol. 10, 2013.
[43] L. S. Marcolino and L. Chaimowicz, "Traffic control for a swarm of robots: Avoiding group conflicts," *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, no. June 2014, pp. 1949–1954, 2009.
[44] M. Santos, Y. Diaz-Mercado, and M. Egerstedt, "Coverage control for multirobot teams with heterogeneous sensing capabilities," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 919–925, April 2018.
[45] Y. Chen, T. T. Georgiou, and M. Pavon, "Optimal transport in systems and control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, 2021.
[46] V. Krishnan and S. Martínez, "Distributed optimal transport for the deployment of swarms," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 4583–4588.
[47] K. F. Caluya and A. Halder, "Wasserstein proximal algorithms for the schrödinger bridge problem: Density control with nonlinear drift," *IEEE Transactions on Automatic Control*, vol. 67, no. 3, pp. 1163–1178, 2021.
[48] Y. Chen, T. T. Georgiou, and M. Pavon, "Stochastic control liaisons: Richard sinkhorn meets gaspard monge on a schrodinger bridge," *SIAM Review*, vol. 63, no. 2, pp. 249–313, 2021.
[49] C. Sinigaglia, S. Bandyopadhyay, M. Quadrelli, and F. Braghin, "Optimal-transport-based control of particle swarms for orbiting rainbows concept," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 11, pp. 2108–2117, 2021.
[50] S.-N. Chow, T.-S. Yang, and H. Zhou, "Global optimizations by intermittent diffusion," *Chaos, CNN, Memristors and Beyond*, pp. 466–479, 2013.