Obscuring Objectives with Pareto-Optimal Privacy-Aware Trajectories in Multi-Robot Coverage

Brennan Brodt and Alyssa Pierson

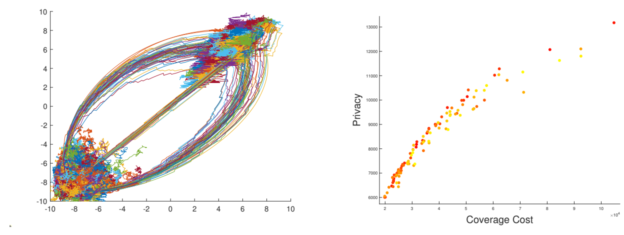
***Abstract*— This paper proposes an algorithm for generating Pareto-optimal privacy-aware trajectories for multi-robot coverage. Our approach utilizes a genetic algorithm to generate a set of modified trajectories for a team of robots that wishes to obscure its goal from an observer. A novel velocity-constrained crossover algorithm ensures all child trajectories are feasible for a holonomic vehicle. The Pareto front of generated trajectories allows a team to select an allowable trade-off between privacy and coverage cost given within their task. Simulation results demonstrate the performance of our algorithm in Voronoi-based coverage control. We show our approach successfully obscures the objective from our proposed observer.**

I. Introduction

Distributed multi-robot systems provide robust, adaptable, and versatile solutions to many problems in environmental coverage. Long-standing consensus controls, built on natural communications, provided a rigorous mathematical foundation for distributed systems [1], with many guarantees on robustness in the presence of problems such as switching network topologies [2] and communication limitations [3], [4]. Meanwhile, more modern automata can provide a framework for allocating tasks and coordinating behaviors to accomplish complex and abstract goals [5], [6]. Further, algorithms such as Voronoi-based coverage control [7] have paved the way for many expansions of the coverage control problem, from automated search and exploration [8] to guarantees on the capture of evaders [9].

While many of these algorithms are robust to performance variations, malicious incursions, and agent failures, they contain vulnerabilities in privacy. In particular, a malicious entity could predict the overall goal locations or targets of the team through observations and trajectory reconstruction. This provides a significant risk when using distributed teams for sensitive tasks. For example, consider a team of drones performing wildlife monitoring of endangered animals. A malicious poacher could observe the trajectories of the drones and identify the location of this wildlife. Another area of risk is foraging for competitive resources, where it is undesirable for rival teams to know the locations of resources. Robots can obscure their goals from malicious observers by introducing deviations in their trajectories, however, this also introduces inefficiencies. A key challenge in generating privacy-aware trajectories for the team is balancing the resultant increase in coverage cost.

Brennan Brodt and Alyssa Pierson are with the Department of Mechanical Engineering, Boston University, Boston, MA 02215, USA [brodt, pierson]@bu.edu. This work supported in part by a Boston University startup grant. We are grateful for their support.



(a) 8^{th}-Generation Trajectories (b) Pareto Front

Fig. 1: Our algorithm generates Pareto-optimal privacy-aware trajectories to obscure a coverage goal from an observer. (a) Sample trajectories generated during the genetic algorithm. (b) Pareto front trading off privacy versus coverage cost.

This paper proposes a new framework for generating privacy-aware trajectories in multi-robot coverage control applications. Given an allowable loss in optimality, we determine the Pareto-optimal trajectories that provide the most privacy for the team. Within this paper, we quantify privacy as the ability to obfuscate critical information from an observer. These trajectories arise from the manipulation of existing control laws to reveal observable features. These features can then be used as a privacy cost function to be optimized alongside the original coverage cost. These objective functions are inherently conflicting and require a trade-off. Evolutionary algorithms are excellent at solving these types of optimizations as their solution populations naturally approach and distribute themselves along the Pareto front. Of these algorithms, the popular NSGA-II [10] algorithm is chosen to generate our candidate trajectories, using a real valued crossover operator to solve the multi-objective optimization problem in free space. The resulting Pareto-optimal trajectories both satisfy prescribed velocity constraints and allow for a team to select trajectories that provide the most privacy for a known increase in coverage cost. The key contributions of this work are:

1) An algorithm that generates a set of Pareto-optimal privacy-aware trajectories for multi-robot coverage;
2) Velocity-constrained crossover for genetic algorithms that produces feasible new trajectories that obey original velocity constraints;
3) Demonstration in simulations that obscure a goal location from an observer.

The remainder of the paper is organized as follows: Section II provides an overview of related work. Section III presents our problem formulation, and Section IV details our trajectory generation algorithm. Simulation results are presented in Sections V, and we state our conclusions in Section VI.

II. Related Work

Most mathematical formalization for the concept of privacy stems from security of statistical databases and protecting the samples that create them [11]. Differential privacy refers to a guarantee that any analysis performed on a database will not be substantially affected by the addition or removal of a single database entry [12]. Such concepts have become very popular in when studying data collected by ridesharing companies and other GPS services [13], [14].

These notions of privacy are also of interest to the robotics community. One early mention of privacy in robotics comes from [15], where a hypothetical panda that is under surveillance could become compromised. This privacy setting was later classified into a set of of privacy problems [16]. Differential privacy appears in [17], where the authors create a model for evaluating privacy in heterogeneous swarms, as well as in limiting the information disclosure between agents during cooperative actions [18]. In [19], the authors balance privacy within flocking behaviors of multi-agent systems with the aid of a genetic algorithm. Within this paper, we assess privacy in our system by the performance of an observer. We specifically explore this in the context of Voronoi-based multi-robot coverage control [7], [20], [21], [22], [23], [24], [25], which is known to be adaptable to changes in environment and team composition. However, to our knowledge, the question of keeping the coverage goal private from an observer within these coverage tasks remains an open question.

Genetic Algorithms: Our approach utilizes a genetic algorithm to analyze trade-offs between privacy and coverage cost. Genetic algorithms are a family of search based optimization algorithms, inspired by the ideas of natural selection and biological evolution [26]. Traditionally, these algorithms simulate a population of candidate solutions which are then subjected to some selection process followed by a mixing of information, also known as crossover. During each generation of the algorithm, new higher-performing children candidates replace the worst solutions in the population. Mutation can also occur at a small rate to randomly change some characteristics of a candidate. Due to the nature of simulating large populations of potential solutions, genetic algorithms excel at solving optimization problems with multiple objective functions [27], [28]. As a genetic algorithm progresses, its candidate solutions will approach the optimal Pareto solutions for the multi-objective problem. In [10], NSGA-II proposed a computationally efficient way to rank solutions and preserve diversity along the front through the fast non-dominated sort and crowding distance comparison functions. Genetic algorithms have an enormous breadth of applications, and robotic path planning is no exception. Since traditional genetic algorithms operate in discrete spaces they are good at path planning on complex graphs which are common problems in robotics. [29], [30], [31]. As for free space planning, genetic algorithms require fine tuning and careful selection of crossover and mutation operators to ensure feasibility. If done properly, these algorithms can solve very challenging multi-objective path planning problems such as the optimal control of a space robot [32].

III. Problem Formulation

In this section, we present preliminaries on the multi-robot coverage control, as well as our observer model. Consider a bounded, convex environment $Q \subset \mathcal{R}^2$, with points in $q \in Q$. For a team of n robots, we denote the positions of the robots as $p_i \in Q$ for $i \in \{1, ..., n\}$. We assume all robots have integrator dynamics,

$$\dot{p}_i = u_i, \tag{1}$$

where u_i is the desired control input for each robot i. All robots are constrained with the same maximum velocity, such that $\|\dot{p}_i\| < v_{max}$. Our team of n robots is tasked with providing coverage in the environment. The function $\phi(q)$ describes the relative importance of a point q.

We also assume the presence of an observer in the environment, which aims to uncover the importance function $\phi(q)$. The observer is not able to gather information directly about $\phi(q)$, however, can observe the motion of the team of agents. To demonstrate the effectiveness of our approach, we will assume a highly-capable observer that can both observe the positions and velocities of all robots. Additionally, we assume the observer knows the team is tasked with Voronoi-based coverage control.

A. Voronoi-Based Coverage Control

We assume the robots are initially tasked with providing coverage of an environment, and start from a Voronoi-based coverage control policy. We define the configuration cost $\mathcal{H}(\cdot)$ to assess the quality of a particular configuration of all robots given the information density $\phi(q)$, defined

$$\mathcal{H}(p_1, ...p_n) = \sum_{i=1}^{n} \int_{V_i} \|q - p_i\|^2 \phi(q) dq, \tag{2}$$

where V_i is the Voronoi cell corresponding to robot i. We can define a mass M_i and centroid C_{V_i} for each cell as

$$M_i = \int_{V_i} \phi(q) dq, \quad C_{V_i} = \frac{1}{M_{V_i}} \int_{V_i} q\phi(q) dq. \tag{3}$$

Although the relationship between the robot positions and the configuration cost is complex, it turns out a control policy that moves all robots towards their centroids will minimize the configuration cost in (2) [7]. This move-to-centroid control policy is defined

$$\dot{p}_i = u_i = -k_{prop}(p_i - C_{V_i}), \tag{4}$$

where k_{prop} is a proportional gain constant. Although this move-to-centroid policy provides robust decentralized coverage, it does not obscure the information density function $\phi(q)$ from an outside observer. The next section demonstrates how an observer can recover this information.

B. Information Observer

The aim of this paper is to generate privacy-aware trajectories that obscure the the team's coverage goal from a highly-capable observer. Here, we present our observer model, which comprises a parameterized fitting function, an observation cost, and is optimized with a Simulated Annealing-based algorithm, outlined in Algorithm 1.

Recall the overall objective is for the observer to reconstruct the information density $\phi(q)$. We employ a Gaussian Mixture Model to create an estimate of $\phi(q)$, denoted $\hat{\phi}(q)$:

$$\hat{\phi}(q, \theta) = \sum_{j=1}^{m} \mathcal{N}(q|\mu_j, \Sigma_j), \tag{5}$$

where θ is a parameter vector containing the values of μ and Σ for the m Gaussians. The observer manipulates values of the parameter vector to approximate the true information density. Naturally, to drive (5) towards the true value, we could create a squared error cost function:

$$\min_{\theta} \mathcal{J}_{\phi} = \|\phi(q) - \hat{\phi}(q, \theta)\|^2. \tag{6}$$

Unfortunately, the observer cannot directly compute $\mathcal{J}_{\phi}$, as it does not have access to $\phi(q)$. Instead, we use knowledge of the the team's control policies to create an alternative cost. Recall form (3) the centroid of of an individual Voronoi cell contains $\phi(q)$. Therefore, by estimating the centroid of the robots' Voronoi cells, we can indirectly also estimate the density function. Consider a cost that assesses the error between true and estimated centroids:

$$\min_{\theta} \mathcal{J}_C = \sum_{i=1}^{n} \|C_{V_i} - \hat{C}_{V_i}\|^2, \tag{7}$$

where $\hat{C}_{V_i}$ is calculated from $\hat{\phi}(q, \theta)$ as

$$\hat{C}_{V_i} = \frac{\int_{V_i} q\hat{\phi}(q, \theta)dq}{\int_{V_i} \hat{\phi}(q, \theta)dq}. \tag{8}$$

Although C_{V_i} is not directly available to the observer, knowledge of the team's control policies allow the observer to minimize (7).

Proposition 1. *The solution that minimizes the cost*

$$\min_{\theta} \mathcal{J}(\theta, p, \dot{p}) = \sum_{i=1}^{n} \|(\frac{\dot{p}_i}{k_{prop}} + p_i) - \frac{\int_{V_i} q\hat{\phi}(q, \theta)dq}{\int_{V_i} \hat{\phi}(q, \theta)dq}\|^2 \tag{9}$$

also minimizes the cost $\mathcal{J}_C$ in (7).

Proof. We assume that the observer has knowledge of the control policies, as it knows the team is performing Voronoi-based coverage control. We can re-arrange an individual robot's control policy (4) as

$$C_{V_i} = \frac{\dot{p}_i}{k_{prop}} + p_i. \tag{10}$$

Substituting (10) and (8) into (7) yields the cost function (9), thus completing the proof. □

Algorithm 1 Simulated Annealing Information Observer

Input: Temperature T, Cooling Schedule N_T, Cooling Rate C_T, Move Limiter α, Feature Vector θ
Initialize random values for θ
Evaluate current energy: $\mathcal{J}(\theta)$ ▷ (9)
Initialize Iteration Counter: $i = 0$
while Observing **do**
    $i \mathrel{+}= 1$
    Generate $\theta_{NEW} = \theta + \alpha * rand$
    Evaluate new energy: $\mathcal{J}(\theta_{NEW})$ ▷ (9)
    if $\mathcal{J}(\theta_{NEW}) \leq \mathcal{J}(\theta)$ **then**
        $\theta = \theta_{NEW}$
    else if $rand \leq \exp(-\frac{\mathcal{J}(\theta_{NEW}) - \mathcal{J}(\theta)}{T})$ **then**
        $\theta = \theta_{NEW}$
    if $N_T \mod i = 0$ **then**
        $T = C_T * T$
        $i = 0$

Although the cost functions (7) and (9) are equivalent, the observer uses the form in (9) as it can determine all necessary variables from observations. We also note this cost in (9) has two desirable properties: first, the cost is positive semi-definite for all possible parameter values. Second, the cost is exactly equal to zero if the parameter estimations are identical to the ground truth. With these properties, we can ensure that a solution to the minimization exists.

We choose to optimize the observer with a Simulated Annealing [33] based approach, presented in Algorithm 1. It is well-known that the configuration cost function (2) contains many local minima, therefore, we require an optimization algorithm that can explore past these local minima. Simulated Annealing provides a balance between exploration and exploitation of the space within quick iterations, allowing it to make predictions in real time alongside a team of robots performing coverage control. Further, it is adaptable to varying density functions. We quantify our observer performance in Section V.

IV. Generating Privacy-Aware Trajectories with a Genetic Algorithm

Given our multi-robot coverage objective, as well as a model of our observer, we now present our algorithm for generating Pareto-optimal privacy-aware trajectories. Within the context of this paper, we define privacy as counter to the objectives of the observer presented in Section III.

Definition 1 (Privacy-Aware Trajectory)**.** *A robot's trajectory that increases the observer cost in* (9)*.*

Here, we now assume the team of robots knows it is under observation, and therefore can modify its behavior to trick the observer. Intuitively, we can define a privacy cost $\mathcal{R}$ that counters the observer cost in (9). We define the privacy cost

$$\max_{p, \dot{p}} \mathcal{R}(p, \dot{p}) = \sum_{i=1}^{n} [(\frac{\dot{p}_i}{k_{prop}} + p_i) - \frac{\int_{V_i} q\phi(q)dq}{\int_{V_i} \phi(q)dq}]^2. \tag{11}$$

Note that while the robots do not know the observer's estimate of the information density function, $\hat{\phi}(q, \theta)$, they do have access to the true information density function $\phi(q)$ and their true centroids C_{V_i}. Trivially, if the robots only maximize the privacy cost (11), then they move in the opposite direction of their centroid, which also maximizes their configuration cost (2). Since their goal is to provide coverage of this information density function, the team must balance the privacy cost (11) with the configuration cost (2), summarized in Problem 1.

Problem 1. *For a given increase in the coverage cost, find the most private trajectories for a team of robots under observation.*

To solve Problem 1, we must quantify this trade-off between trajectory level privacy and optimality. We propose a multi-objective optimization algorithm that generates Pareto-optimal solutions based upon NSGA-II [10], a popular evolutionary algorithm. We also introduce a new crossover method which guarantees feasible child trajectories from any two parent trajectories.

A. Evaluating Trajectory Costs

Although traditional coverage-control algorithms run online, here, we generate the trajectories for the team offline with a genetic algorithm. We define a trajectory for a robot i as W_i, with W representing the trajectories of all robots. Waypoints within the trajectory are denoted w_k for $k = \{1, ..., m\}$ waypoints. Our genetic algorithm evaluates two costs that represent the trade-off between coverage cost $\mathcal{H}$ and privacy $\mathcal{R}$:

$$\min_{W} \mathcal{H}(W) = \sum_{j=1}^{m} \sum_{i=1}^{n} \int_{V_{i_j}} \|q - w_{i_j}\|^2 \phi(q) dq, \tag{12}$$

$$\max_{W} \mathcal{R}(W) = \sum_{j=1}^{m} \sum_{i=1}^{n} [(\frac{\dot{p}_{i_j}}{k_{prop}} + w_{i_j}) - \frac{\int_{V_{i_j}} q\phi(q) dq}{\int_{V_{i_j}} \phi(q) dq}]^2. \tag{13}$$

As these costs exist in a continuous space, a real-valued genetic algorithm must be used. Algorithm 3 utilizes these costs, starting from an initial set of feasible trajectories. Given the maximum velocity constraints on our robots, we require that the children trajectories have a feasible set of waypoints. Typical point crossover methods do not incorporate these constraints; if two parent solutions are very different, blending crossover methods may generate an infeasible trajectory. Therefore, we propose a new crossover method for generating feasible velocity-constrained trajectories with real-valued genetic algorithms.

B. Real Valued Velocity Constrained Crossover

Our proposed crossover method in Algorithm 2, balances the dissemination of information between child solutions while guaranteeing feasibility. We enforce the maximum velocity constraint for our robots with a maximum allowable distance between any two waypoints in the trajectory, represented with a circle of radius r_v, illustrated in Figure 2. During crossover, any new waypoint generated within this circle is considered a valid next waypoint. By computing the intersections between constraints and lines connecting crossover points, this method shares information similarly to other real value blending methods. By including mutation in the crossover process, we can ensure that feasibility is maintained while providing a means for information outside the parent populations extrema to be introduced to the algorithm.

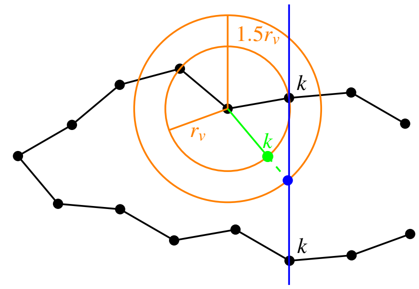


Fig. 2: Visualization of a simple crossover case. For two candidate trajectories, new crossover points are calculated at the k-th waypoint. Geometric constraints on generation ensure new candidates are feasible.

Algorithm 2 Real Valued Velocity Constrained Crossover

Input: Parent Trajectories $W_{parent_1}, W_{parent_2}$, Travel Constraint r_v, Crossover indices W_i, Mutation rate λ
Generate children c_1 and c_2 by replicating W_{parent_1} and W_{parent_2}
for $k \in W_i$ **do**
  Generate line, ℓ, intersecting $c_1(k)$ and $c_2(k)$
  for c_1, c_2 **do**
    Generate circular constraints, $circ_v$
    Generate circular hyper-constraint, $hcirc_v$, of radius $1.5r_v$ centered on $c(k-1)$
    Calculate intersections, int_1, int_2, of ℓ with $hcirc_v$
    Classify intersections as $int_{cross} = int_1, int_2 \in \{c_1(k) - c_2(k)\}$ and $int_{mut} = int_1, int_2 \notin \{c_1(k) - c_2(k)\}$
    if $\forall int \notin \{c_1(k) - c_2(k)\}$ **then**
      $c(k)_{NEW} = circ_v \cup (c(k-1) - \frac{1}{2}(c_1(k) + c_2(k)))$
    else if $rand < \lambda$ **then**
      $c(k)_{NEW} = int_{mut}$
    else
      $c(k)_{NEW} = int_{cross}$
    $[c(k), ..., c(end)] += c(k)_{NEW} - c(k)$

C. Algorithm Overview

With established cost functions (12) and (13), and the proposed crossover method in Algorithm 2, we now present our algorithm for generating privacy-aware trajectories, detailed in Algorithm 3. First, we generate an initial population of feasible trajectories, based on the initial locations of all robots. Note that the trajectory that solves (2) repre-

Algorithm 3 Privacy-Aware Trajectory Generation

Input: Initial Agent Positions p, Information Density $\phi(q)$
Generate Voronoi Optimal Trajectory $W(1)$
Generate Random Initial Trajectories $W(2 : P_{MAX})$
for $G \leq G_{MAX}$ **do**
  Compute Costs $\mathcal{H}, \mathcal{R} \ \ \forall W_G$ ▷ (12,13)
  Rank solutions in W_G by Pareto dominance ▷ [10]
  Sort Pareto fronts by crowding distance ▷ [10]
  Add $W_G(1 : P_{MAX}/3)$ to W_{G+1}
  while $W_{G+1} < W_G$ **do**
    Select two parent W's from W_G
    Apply crossover to generate c_1 and c_2 ▷ Alg. 2
    Add c_1 and c_2 to W_{G+1}

sents a Pareto-optimal solution. We choose this trajectory to seed the initial population, ensuring that high quality information on optimality rapidly spreads through the early generations of populations. The rest of the initial population comprises randomly-generated trajectories that obey velocity constraints. Figure 4(a) illustrates an initial population with $k = 250$ waypoints and a population of 120.

We evaluate each trajectory for both coverage cost (12) and privacy (13), then sort solutions within the population by the fast non-dominating sort with crowding distance proposed in [10]. Next, we replicate the top third of the population into the new generation. Tournament selection is held for between all solutions in the old generation, removing selected trajectories from the tournament. This style of tournament favors more optimal solutions for crossover, while still allowing for exploration with less optimal solutions. From the parent pairs, crossover and mutation occurs with Algorithm 2. We repeat this process until the new generation is the same size as the old generation, and the cycle repeats. Algorithm 3 summarizes these steps.

V. Simulations

Simulations conducted in Matlab demonstrate the efficacy of our algorithms. Within our simulated environment, we first optimize for the privacy-aware trajectories with Algorithm 3. Then, we assess the performance by running these trajectories in parallel with the observer (Algorithm 1). Our results show the ability to create a diverse set of trajectories, which reduce the performance and efficacy of an observer. Videos with animations of the simulations are included with the submission of this paper.

A. Baseline Observer Performance

The performance of the observer can be quantified by calculating the Mean Square Error (MSE) between the peak of $\phi(q)$ and the peak of its estimate $\hat{\phi}(q, \theta)$. Figure 3 depicts the mean-variance plot of this observer error over 100 randomized simulations. We tested our observer against teams of three robots performing Voronoi-based coverage control for 250 seconds. For these simulations, the information density function evolves with an elliptical orbit. As shown in

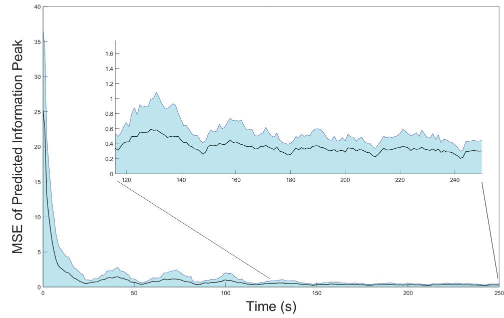


Fig. 3: MSE for an observer tracking a moving $\phi(q, t)$ function for 100 randomized trials.

Figure 3, the MSE quickly converges to zero, implying the observer identifies the true peak location, and successfully tracks the movement of the peak over time. In subsequent simulations, we assume a static $\phi(q)$, but include it here for demonstration.

B. Generating Diverse Trajectories

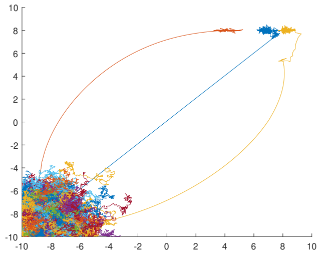
Figures 1 and 4 illustrate the evolution of privacy-aware trajectories for a team of $n = 3$ robots. The initial population in Figure 4a contains a total of 120 trajectories, including the Voronoi-optimal trajectory alongside 119 randomized feasible trajectories. Each trajectory contains $k = 250$ waypoints, and we run Algorithm 3 for 60 generations. After 8 generations, we begin to see a variety of trajectories emerge in Figure 1a, with the complete set in Figure 4b illustrating the diversity. We also see after 60 generations that these solutions fill out the Pareto front in Figure 4f. In this scenario, the information density $\phi(q)$ contains a single peak at $[7.5, 7.5]$ in the upper-right corner. Figure 4c illustrates the trajectory with the highest privacy score on the Pareto front, while Figure 4d balances privacy with optimality. As expected, the most optimal trajectory in Figure 4e is the Voronoi-based coverage control solution without modification.

The resulting Pareto front highlights the ability to choose an allowable trade-off between privacy and coverage cost. Our algorithm successfully populates a diverse set of candidate trajectories, which allow a team to improve their privacy based on an acceptable increase in coverage cost. Further, all trajectories are feasible, guaranteed by the velocity-constrained crossover method in Algorithm 2. Next, we examine the effect of increasing privacy on the observer's performance.

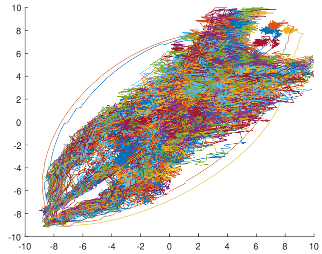
C. Impact on Observer's Performance

We replay the trajectories of the team and run the observer alongside the trajectories to assess the observer's ability to predict the information density $\phi(q)$, which encodes the team's goal. We then assess the performance of the observer in the same manner as section V.A.

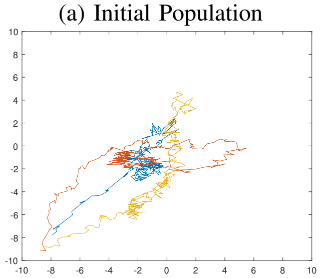
Recall in Figure 3, without any trajectory modification the observer quickly converged to the actual value of $\phi(q)$, repre-



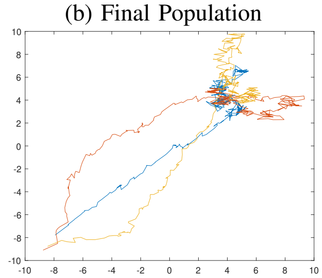
(a) Initial Population



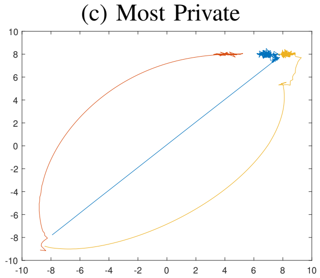
(b) Final Population



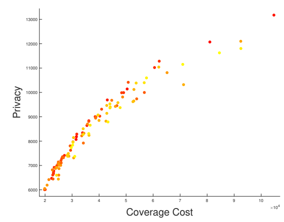
(c) Most Private



(d) Intermediate

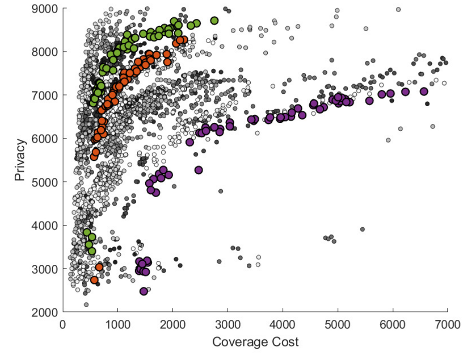


(e) Most Optimal

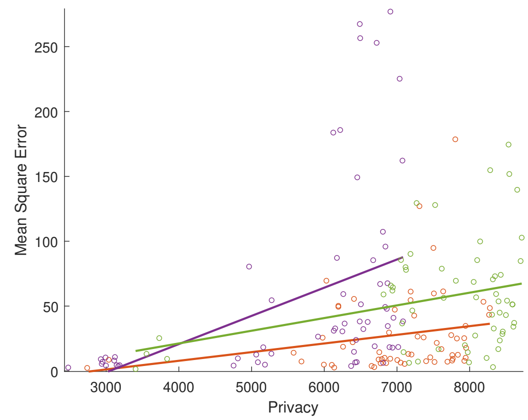


(f) Pareto Front

Fig. 4: Algorithm 3 produces privacy-aware trajectories that allows a team to trade-off between privacy and optimality. (a) Initial population of trajectories for $n = 3$ robots, with the Voronoi-based solution provided. (b) Final trajectories after 60 generations of all 120 trajectory sets. (c) The trajectory with the most private cost along the Pareto front. (d) A trajectory that balances optimality with privacy. (e) The most-optimal trajectory, which is the Voronoi-based solution. (f) Pareto front of all trajectories.



(a) Pareto Fronts for Randomized Scenarios



(b) Observer MSE Across the Pareto Front

Fig. 5: (a) Pareto fronts for randomized scenarios. Each color represent a different set of trajectories. (b) Corresponding observer MSE along the Pareto fronts for a given privacy score of a trajectory. As the privacy score increases, so does the observer MSE, successfully demonstrating the obfuscation of the goal location.

senting a large privacy risk. Figure 5 illustrates three different randomized scenarios, comparing the generated Pareto front of trajectories to observer performance. Each of these random initializations contains a population size of 60 trajectories, $k = 120$ waypoints, and ran for 60 generations. To give our observer the best performance, in each case the information density function $\phi(q)$ was a single Gaussian peak with a randomized peak location. In total, we ran 44 randomized scenarios. Figure 5a highlights three selected Pareto fronts from the set of all scenarios. Due to randomization in initial conditions and peak location, the fronts have varying slopes and ranges of coverage and privacy costs. Figure 5b plots the corresponding observer MSE for the varying privacy values in each Pareto front. The MSE was averaged over the final 20 waypoints of the trajectories, in order to assess a near steady-state performance. Across all three scenarios, we see a clear trend that as privacy increases, the MSE of the observer also increases. These results validate our approach, and show that our privacy-aware trajectory generation algorithm can successfully help a team obscure their goal from an observer.

VI. Conclusions

In this paper we have proposed a new method for including privacy in coverage path planning for robotics. The multi-objective genetic algorithm has been demonstrated as effective for similar problems before and continues to be a powerful tool here. The newly proposed crossover method for genetic algorithms also offers great value for the use of genetic algorithms on other robotics problems, as it extends more traditional methods to allow for easy inclusion of common constraints and provides both exploration and exploitation of the free space. While the focused control law in this work was that of Voronoi-based coverage control, the methods discussed represent a broader manner of investigation for other possible controllers. By determining information to be kept private and deriving a privacy cost from a similar worst-case observer, the same optimization could be run to obfuscate the information and improve privacy.

References

[1] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[2] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[3] C. Huang, H. Li, D. Xia, and L. Xiao, "Distributed consensus of multi-agent systems over general directed networks with limited bandwidth communication," *Neurocomputing*, vol. 174, pp. 681–688, 2016.

[4] Z. Qiu, L. Xie, and Y. Hong, "Quantized leaderless and leader-following consensus of high-order multi-agent systems with limited data rate," *IEEE Transactions on Automatic Control*, vol. 61, no. 9, pp. 2432–2447, 2016.

[5] M. Khani, A. Ahmadi, and H. Hajary, "Distributed task allocation in multi-agent environments using cellular learning automata," *Soft computing (Berlin, Germany)*, vol. 23, no. 4, pp. 1199–1218, 2017.

[6] D. Rus, B. Donald, and J. Jennings, "Moving furniture with teams of autonomous robots," in *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, vol. 1, 1995, pp. 235–242 vol.1.

[7] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[8] K. R. Guruprasad and D. Ghose, "Automated multi-agent search using centroidal voronoi configuration," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 2, pp. 420–423, 2011.

[9] A. Pierson, Z. Wang, and M. Schwager, "Intercepting rogue robots: An algorithm for capturing multiple evaders with multiple pursuers," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 530–537, 2017.

[10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[11] N. R. Adam and J. C. Worthmann, "Security-control methods for statistical databases: A comparative study," *ACM Comput. Surv.*, vol. 21, no. 4, p. 515–556, dec 1989.

[12] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation*, M. Agrawal, D. Du, Z. Duan, and A. Li, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1–19.

[13] A. Prorok and V. Kumar, "Privacy-preserving vehicle assignment for mobility-on-demand systems," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1869–1876.

[14] B. Li, H. Zhu, and M. Xie, "Releasing differentially private trajectories with optimized data utility," *Applied Sciences*, vol. 12, no. 5, 2022.

[15] J. M. O'Kane, *On the Value of Ignorance: Balancing Tracking and Privacy Using a Two-Bit Sensor*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 235–249.

[16] Y. Zhang and D. A. Shell, "Complete characterization of a class of privacy-preserving tracking problems," *The International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 299–315, 2019.

[17] A. Prorok and V. Kumar, "A macroscopic privacy model for heterogeneous robot swarms," in *Swarm Intelligence*, M. Dorigo, M. Birattari, X. Li, M. López-Ibáñez, K. Ohkura, C. Pinciroli, and T. Stützle, Eds. Cham: Springer International Publishing, 2016, pp. 15–27.

[18] L. Li, A. Bayuelo, L. Bobadilla, T. Alam, and D. A. Shell, "Coordinated multi-robot planning while preserving individual privacy," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 2188–2194.

[19] H. Zheng, J. Panerati, G. Beltrame, and A. Prorok, "An adversarial approach to private flocking in mobile robot teams," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1009–1016, 2020.

[20] M. Schwager, D. Rus, and J.-J. Slotine, "Decentralized, adaptive coverage control for networked robots," *The International Journal of Robotics Research*, vol. 28, no. 3, pp. 357–375, 2009.

[21] J. Cortés, "Coverage optimization and spatial load balancing by robotic sensor networks," *Automatic Control, IEEE Transactions on*, vol. 55, no. 3, pp. 749–754, 2010.

[22] S. G. Lee and M. Egerstedt, "Controlled coverage using time-varying density functions," in *Proc. of the IFAC Workshop on Estimation and Control of Networked Systems*, 2013.

[23] A. Pierson, L. C. Figueiredo, L. C. Pimenta, and M. Schwager, "Adapting to sensing and actuation variations in multi-robot coverage," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 337–354, 2017.

[24] A. Sadeghi and S. L. Smith, "Coverage control for multiple event types with heterogeneous robots," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 3377–3383.

[25] J. Kennedy, P. M. Dower, and A. Chapman, "Exponential convergence in voronoi-based coverage control," in *2021 Australian & New Zealand Control Conference (ANZCC)*, 2021, pp. 226–231.

[26] D. E. D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Reading, Mass.: Addison-Wesley Publishing Company, 1989.

[27] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[28] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Comput. Surv.*, vol. 48, no. 1, sep 2015.

[29] J. Lee and D.-W. Kim, "An effective initialization method for genetic algorithm-based robot path planning using a directed acyclic graph," *Information Sciences*, vol. 332, pp. 1–18, 2016.

[30] A. Tuncer and M. Yildirim, "Dynamic path planning of mobile robots with improved genetic algorithm," *Computers & Electrical Engineering*, vol. 38, no. 6, pp. 1564–1572, 2012.

[31] K. Ellefsen, H. Lepikson, and J. Albiez, "Multiobjective coverage path planning: Enabling automated inspection of complex, real-world structures," *Applied Soft Computing*, vol. 61, pp. 264–282, 2017.

[32] A. Seddaoui and C. M. Saaj, "Collision-free optimal trajectory generation for a space robot using genetic algorithm," *Acta Astronautica*, vol. 179, pp. 311–321, 2021.

[33] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.