electronics

MDPI

Article

Opponent-Aware Planning with Admissible Privacy Preserving for UGV Security Patrol under Contested Environment

Junren Luo, Wanpeng Zhang *, Wei Gao, Zhiyong Liao, Xiang Ji and Xueqiang Gu

College of Intelligence Science and Technology, National University of Defense and Technology, Changsha 410073, China; luojunren17@nudt.edu.cn (J.L.); gaowei14@nudt.edu.cn (W.G.); liaozhiyong18@nudt.edu.cn (Z.L.); jixiang14@nudt.edu.cn (X.J.); xqgu_nudt@163.com (X.G.)

* Correspondence: wpzhang@nudt.edu.cn

Received: 13 October 2019; Accepted: 4 December 2019; Published: 18 December 2019

Abstract: Unmanned ground vehicles (UGVs) have been widely used in security patrol. The existence of two potential opponents, the malicious teammate (cooperative) and the hostile observer (adversarial), highlights the importance of privacy-preserving planning under contested environments. In a cooperative setting, the disclosure of private information can be restricted to the malicious teammates. In adversarial setting, obfuscation can be added to control the observability of the adversarial observer. In this paper, we attempt to generate opponent-aware privacy-preserving plans, mainly focusing on two questions: what is opponent-aware privacy-preserving planning, and, how can we generate opponent-aware privacy-preserving plans? We first define the opponent-aware privacy-preserving planning problem, where the generated plans preserve admissible privacy. Then, we demonstrate how to generate opponent-aware privacy-preserving plans. The search-based planning algorithms were restricted to public information shared among the cooperators. The observation of the adversarial observer could be purposefully controlled by exploiting decoy goals and diverse paths. Finally, we model the security patrol problem, where the UGV restricts information sharing and attempts to obfuscate the goal. The simulation experiments with privacy leakage analysis and an indoor robot demonstration show the applicability of our proposed approaches.

Keywords: UGV; privacy-preserving planning; information leakage; security patrol

1. Introduction

With the development of intelligent unmanned system technology, unmanned ground vehicles (UGVs) have become extremely rugged for the harshest military use, such as executing monitoring tasks in harsh and complex urban environments [1]. As our world becomes increasingly well-connected, there is an increased need to enable UGVs to cooperate in generating plans for security patrol. For example, the iRobot's PackBot, which has played a critical role in providing situational awareness for anti-terrorist operations [2].

Several approaches have been proposed in recent years to address the conundrum of privacy preservation through controlling privacy leakage for different requirements under contested environments. One of them is differential privacy [3], which adds appropriate noise to the transmission state in order to limit the opponent to acquiring only the true state of the transmitted signal at a predetermined level of accuracy. Another approach uses cryptography for secure multi-party computation (MPC). In [4,5], the authors encrypt the messages with a public-key homomorphic cryptosystem and apply techniques (e.g., random masking and random permutation) to protect the

agents' privacy. So, the encrypted messages can be exchanged among the agents in various ways [6]. A third approach tries to guarantee privacy as a loss of observability [7,8], but it is difficult to achieve strong privacy of this form. All these mainly focus on the privacy leakage from the information perspective, while decision-related privacy preservation (e.g., privacy-preserving planning) has mostly been neglected.

Several recent pieces of research on privacy-preserving planning for multi-agent systems have captured the attention of the planning community [9–11]. Privacy-preserving plans represent plans that do not actively disclose sensitive private information. In fact, privacy preservation is the goal pursued by multi-agent planning, which has been a crucial concern for multi-agent systems in some contexts, such as agent negotiation [12], multi-agent reinforcement learning and policy iteration [4,5], deep learning [13], and distributed constraint optimization problems (DCOPs) [14–16]. Multi-agent planning (MAP) in cooperative environments aims at generating a sequence of actions to fulfill some specified goals [17]. Most multi-agent systems rely intrinsically on collaboration among agents to accomplish a joint task, in which the collaboration depends on the exchange of information among them, so the privacy preservation of the information naturally rises.

Security patrol has been widely studied among the defense and security fields in the past decade. Unmanned ground vehicle patrols have gained increased interest in recent decades mainly due to their relevance to various security applications [18]. The common mode of urban security patrol is to patrol checkpoints. As illustrated in Figure 1, the UGVs perform security patrol, one supply center, and some determined checkpoints are distributed across the security patrol environment. The UGVs are required to repeatedly visit some checkpoints to monitor the local area, but the UGVs do not share information about the task plan with the supply center. We assume that the adversary can exploit any predictable behavior of the UGVs, which means the adversary has full knowledge of the patrolling task. Since the opponent is collecting information, the objective of privacy-preserving planning is to protect private information in different situations.

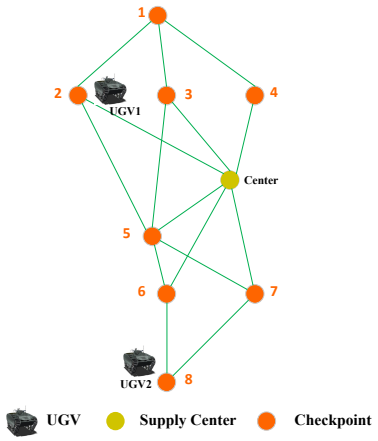


Figure 1. Typical urban security patrol scenario with some checkpoints on a simplified road network. The UGVs should patrol two zones, each with four candidate checkpoints, and the supply center will provide support for the UGVs.

Regarding urban security patrol, some checkpoints located around the urban trunk road are high risk. Thus, it is feasible to deploy UGVs to patrol such checkpoints regularly and collect information (e.g., images, video, etc.). Although UGVs have become quite ubiquitous in patrols with logging and tracking capabilities, they are mostly at risk. The hostile observer will constantly monitor the task execution and get access to the UGVs' data and actions. The challenge of privacy preservation arises because all aspects of information are private and the UGVs are not eager to share, which drives us to compute privacy-preserving plans that can protect privacy when executed in cooperative and adversarial environments.

In this paper, we address the problem of opponent-aware privacy-preserving planning for security patrol and attempt to answer the following questions: what is opponent-aware privacy-preserving planning, and how can we generate opponent-aware privacy-preserving plans? Our contribution lies in the opponent-aware privacy-preserving planning architecture. In a cooperative setting, the search-based planning method could be restricted to obtain the public information shared by the cooperative agents. Whereas, in an adversarial setting, the observation of the adversary could be purposefully controlled by exploiting decoy goals and diverse paths. Finally, simulation experiments with privacy leakage analysis and indoor robot demonstration show the applicability of our proposed approaches.

The rest of this paper is organized as follows. In Section 2, some related works about privacy, security, and metrics are presented. In Section 3, we decompose the opponent-aware privacy-preserving planning problem into two subproblems from different perspectives. In Section 4, experimental evaluations of plan generation and information leakage analysis were conducted. In Section 5, we conclude this paper and point out further directions.

2. Background and Related Work

2.1. Privacy and Security Assumption

Privacy and Security: Many privacy models have been adopted in multi-agent planning according to three different criteria: the information model (imposed privacy [19], induced privacy [20]), the information-sharing scheme (MA-STRIPS [19], subset privacy [21]), and practical privacy guarantees (no privacy [22], weak privacy [23], object cardinality privacy [24], and strong privacy [9]). Privacy can be divided into different categories, such as agent privacy, model privacy, decision privacy, topology privacy, and constraint privacy [4,14]. Here we introduce some widely used types of privacy.

Definition 1 (Agent privacy)**.** *No agent should be able to recognize the identity or existence of another agent.*

Agent privacy can be achieved by employing anonymous or coded names. Such as one agent would not want the opponents to know the identity or existence.

Definition 2 (Model privacy)**.** *No agent should be able to recognize the model of another agent, including environmental and algorithmic models which are related to states, actions, observations, transition probability, and rewards.*

Model privacy is the key issue in adversarial environments; one agent will not get more information about others except for what has been revealed.

Planning algorithms for privacy preserving can be divided into weak or strong privacy [17], ε-strong privacy [25], and provable guarantees privacy [26].

Definition 3 (Weak privacy preserving)**.** *The agent will not disclose private information of the states, private actions, and private parts of the public actions during the whole run of the algorithm.*

In other words, the agent will share only the information in the public part. Even if not communicated, the adversary will deduce the existence and values of private variables, preconditions, and effects from the (public) information communicated.

Definition 4 (Strong privacy preserving)**.** *The adversary can deduce no information about the private variables, preconditions, or effect of the actions, beyond the shared public projection of actions and plans.*

Privacy essentially concerns a semi-honest adversary who is interested in learning the information. Privacy is equivalent to the concept of unobservability among the control community, and it is closely

related to the concept of semantic security from cryptography [27], where secure plans build on the concept of independent inputs [28]. A secure plan is always private, which imposes an additional constraint (all possible goals must result in to the same observations) to the privacy problem [29].

Security Assumption: In [30], the authors define the notion of privacy-preserving planning based on secure MPC and provide some proper analysis of privacy leakage in multi-agent planning. Many assumptions specify the properties of the agent, environment, and algorithm in some secure multi-party computation literature [10,28,31].

Assumption 1 (Adversary model)**.** *An honest but curious adversary who is passive and follows the algorithm and the protocol correctly but may glean information from the execution and communicated data to learn about the privacy. A malicious adversary, who can actively deviate from the protocol specification.*

Assumption 2 (Algorithm known)**.** *The adversary has access to the algorithm and knows how the algorithm works. The agent should not rely on the privacy of the algorithmic mechanism itself.*

Assumption 3 (Input independent)**.** *The adversary can rerun the algorithm by setting different goals as real goals to check the variability of the output.*

Assumption 4 (FIFO)**.** *When the actor takes action to reach a corresponding state, only then does the adversary receive the corresponding observation in the order which was emitted by the plan execution.*

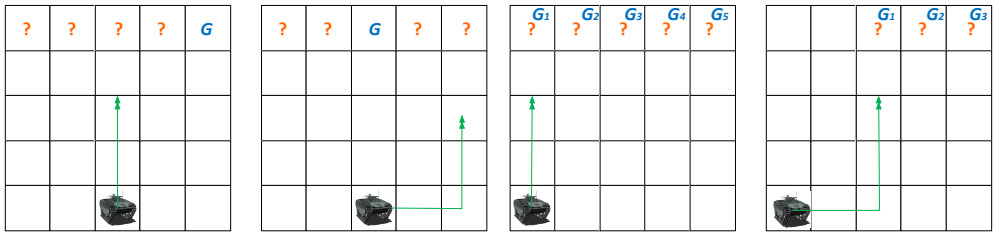
As is usually done by cryptography, these assumptions do not take the adversary's recognition model into consideration, which is quite different from the artificial intelligence (AI) community.

2.2. Privacy-Preserving Planning

The planning problem of privacy preserving can be modeled as a multi-agent planning (MAP) problem with a privacy-preserving requirement. MAP comes in different types, such as deterministic MAP (DMAP) [19,32], interactive partially observable Markov decision processes (I-POMDPs) [33], and decentralized POMDPs (Dec-POMDPs) [34]. Regarding privacy, there are many synonymous concepts in the recent literature, which all aim at generating obfuscated behavior, such as deception, security, and obfuscation, as shown in Table 1. A secure plan is always private; a deceptive plan is always obfuscating, but may or may not be dissimulating [29]. A simple illustration of different strategies is shown in Figure 2.

Table 1. Some synonymous concepts of privacy.

Concepts	Main Contributions
Obfuscation	k-ambiguous and d-diverse [35]
	one candidate goal [36]
	secure MAFS [9]
Privacy	privacy leakage [10]
	plan set intersection [11]
	privacy-preserving policy iteration [4]
Security	equidistant states [28]
	last deceptive point [37,38]
	deceptive shortest path [39]
	equidistant states [28]
Deception	bounded deception [40]
	hide intention [41]
	λ Deception [42]
	deceptive adversary [43]



(**a**) Simulation (**b**) Dissimulation (**c**) Privacy (**d**) Security

Figure 2. Deception Strategies: (**a**) Simulation with the UGV hiding the true goals and going to either of the five goals. (**b**) Dissimulation with the UGV showing false goals; the probability of decoy goals are higher than the true goal. Obfuscation strategies: (**c**) Privacy with the UGV going to either of the five goals; the resulting plans could be deceptive. (**d**) Security with the UGV could go to either of the three goals under rational assumption.

In a cooperative environment, many multi-agent planners have been proposed to address privacy-preserving planning problems, such as MAFS (multi-agent forward search) [30], MADLA (multi-agent distributed and local asynchronous) [44], and PSM (planning state machine) [11]. In [11], the author proposed one secure planner for multi-agent planning, but this planner is impractical to compute all possible solutions. In [9], the authors introduce a modified version of the multi-agent forward search algorithm, Secure-MAFS [30], which is implemented based on an equivalent macro sending technique [24]. Some privacy guarantee planning algorithms have been provided in [9], but they are restricted to very special cases.

In an adversarial environment, the adversary implicitly uses the signal behavioral cues of the actors during the plan execution, and perform diagnosis on the internal information based on the resulting observations. Recently, there has been some interest in exploring privacy preservation [36], goal obfuscation [28,35], deception [37,38], intention hiding [41], etc. In [35], Kulkarni et al. attempted to make plans with k-ambiguous goals, but they were not guaranteed to be secure. In [36], Keren et al. proposed to preserve privacy by keeping their goal ambiguous for as long as possible, but there was only one candidate goal and one partially obfuscated plan. In [38], Masters et al. applied some deceptive strategies for path planning, but these do not support deception when the adversary knows the explicit model. In [28], Kulkarni et al. proposed to securely obfuscate the real goal by making all candidate goals equally likely for as long as possible, but the heuristic deployed makes the planner incomplete. All these studies employ goal or plan recognition modules.

2.3. Information Leakage Metric

Although the key motivation for privacy-preserving planning is preserving privacy, some private information will be leaked during the planning, which means it is impossible to achieve complete privacy. If one malicious teammate directly receives any of the private information, or can indirectly deduce the privacy from the communicated public information, the privacy information will be leaked. To evaluate the privacy leakage, we consider the foundations of quantitative information flow [45]. The leakage of the private information is based on the uncertainty of the adversary about the input. Here we use the min-entropy (an instance of Rényi entropy [46]) as a better measure of the privacy information leakage (PIL):

$$PIL = H_\infty(H) - H_\infty(H|L), \tag{1}$$

where the initial uncertainty is $H_\infty(H)$, the residual uncertainty is $H_\infty(H|L)$.

Using the uniform distribution case, we denote the number of states as t_{prio} and t_{post}, then the remaining uncertainty gives a security guarantee. The expected probability that the adversary could

guess H given L decreases exponentially with $H_\infty(H|L)$: $2^{-H_\infty(H|L)} = 2^{-\log t_{\text{post}}} = 1/t_{\text{post}}$, and we can obtain the privacy information leakage:

$$PIL = \log t_{\text{prio}} - \log t_{\text{post}} = \log \frac{t_{\text{prio}}}{t_{\text{post}}}. \quad (2)$$

3. Methodology

3.1. Opponent-Aware Privacy-Preserving Planning

In privacy-preserving planning (PPP), it is important to acknowledge that two potential opponents are involved: the malicious teammate (cooperator) and the hostile observer (adversary). PPP should produce plans that reveal neither the goal nor the activities of the agents, but many planners cannot have completeness, strong privacy preserving, and efficiency together. So, it is practical for them to achieve opponent-aware privacy preservation within bounded privacy information leakage. As illustrated in Figure 3, privacy leakage will occur at the information layer and decision-making layer. At the information layer, differential privacy and homomorphic encryption are applicable techniques to protect private information.

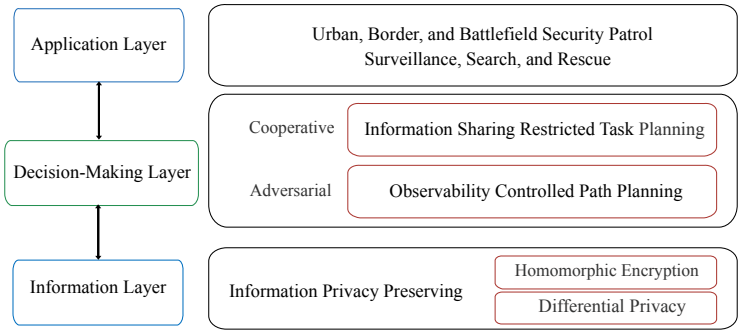


Figure 3. Privacy-preserving methods for decision-making layer and information layer, and the application layer for some application areas.

In this paper, we mainly focus on the middle layer for decision-making. For task planning in cooperative environments, we need to restrict the information sharing to malicious teammates, and for the path planning in adversarial environments, we need to control the observability of the adversary. Here, we define the opponent-aware privacy-preserving planning problem as follows:

Definition 5 (Opponent-aware privacy-preserving planning)**.** *We define opponent-aware privacy-preserving planning as a multi-agent planning problem of protecting privacy secure to certain extent considering two opponents.*

As a result, the generated plans protect privacy from two potential opponents: the malicious teammate and the hostile observer. In a cooperative setting, to cope with malicious teammates, we should restrict the disclosure of private information to malicious teammates. In adversarial settings, real combat scenarios often consist of hostile opponents, so we will add obfuscation to control the observability of the opponents.

3.2. Information Sharing Restricted Task Planning

In cooperative environments, the agents are cooperative in concurrently planning and executing their local plans to achieve a joint goal. We could model all other agents as a single adversary, who can collect the information to infer more. Information sharing restricted task planning with privacy preservation can be defined as follows [10]:

Definition 6 (Information sharing restricted task planning)**.** *For a set of agents $\mathcal{N}$, the information sharing restricted task planning problem for multi-agent $\mathcal{M} = \{\Pi_i\}_{i=1}^{|\mathcal{N}|}$ is a set of agent problems, where for each agent $n \in \mathcal{N}$ the problem is:*

$$\Pi_i = \left\langle \mathcal{V}_i = \mathcal{V}^{pub} \cup \mathcal{V}_i^{priv}, \mathcal{A}_i = \mathcal{A}_i^{pub} \cup \mathcal{A}_i^{priv}, \mathcal{I}, \mathcal{G} \right\rangle, \tag{3}$$

where $\mathcal{V}_i$ is a set of variables, s.t. each $V \in \mathcal{V}_i$ has a finite domain $dom(V)$, if $|dom(V)| = 2$, then all variables are binary. $\mathcal{V}^{pub}$ is the set of public variables common to all agents and $\mathcal{V}_i^{priv}$ is the set of variables private to agent $n_i \in \mathcal{N}$, s.t. $\mathcal{V}^{\text{pub}} \cap \mathcal{V}_i^{\text{priv}} = \emptyset$. The state $\mathcal{I}$ is the initial state and $\mathcal{G}$ is the goal.

Each action is defined as a tuple $a = \langle \text{pre}(a), \text{eff}(a), \text{cost}(a) \rangle$, where $\text{pre}(a)$ and $\text{eff}(a)$ are partial states representing the precondition and effect, respectively, $\text{cost}(a)$ is the cost of action a. So, the state transition can be defined as $\Gamma(s, a) \models s \cup \text{eff}(a)$. We follow the formal treatment of privacy-preserving planning from [10,30], for each agent $n \in \mathcal{N}$, the private parts of the problem Π_i are:

- The set of private variables $\mathcal{V}_i^{priv}$ and the number $|\mathcal{V}_i^{priv}|$, the domains $dom(V)$ and the size $|dom(V)|$.
- The set of private actions $\mathcal{A}_i^{priv}$ and the number $|\mathcal{A}_i^{priv}|$, the number and values of variables in $\text{pre}(a)$ and $\text{eff}(a)$.
- The private parts of the public actions in $\mathcal{A}^{\text{pub}}$, such as the numbers and values of private variables in $\text{pre}(a) \cap \mathcal{V}_i^{priv}$ and $\text{eff}(a) \cap \mathcal{V}_i^{priv}$ for each action $a \in \mathcal{A}_i^{pub}$.

3.2.1. Task Plan Generation

The multi-agent planning problem $\mathcal{M} = \{\Pi_i\}_{i=1}^{|\mathcal{N}|}$ can be viewed from different perspectives, called projections. The view of a single agent n_i on the global problem is not the only Π_i, projections of other agents are available as well. As for agent n_i, the public projection of an action $a \in \mathcal{A}_i^{pub}$ is $a^{\triangleright} = \langle \text{pre}(a)^{\triangleright}, \text{eff}(a)^{\triangleright} \rangle$, the public projection of Π_i can be represented as follows:

$$\Pi_i^{\triangleright} = \left\langle \mathcal{V}^{pub}, \mathcal{A}_i^{\triangleright} = \{a^{\triangleright} | a \in \mathcal{A}_i^{pub}\}, \mathcal{I}^{\triangleright}, \mathcal{G}^{\triangleright} \right\rangle \tag{4}$$

So, the task planning solution to Π_i is a sequence π_i of actions from $\mathcal{A}_i \cup \bigcup_{j \neq i} \mathcal{A}_i^{\triangleright}$, the goal state $\mathcal{G}_k = \pi_i \circ \mathcal{I}$, which means $\Gamma(\mathcal{I}, \pi_i) \models \mathcal{G}_k$. The public projection of π is $\pi^{\triangleright} = (a_1^{\triangleright}, ..., a_k^{\triangleright})$ with all private actions omitted. The global solution of $\mathcal{M}$ is a set of task plans $\{\pi_i\}_{i=1}^{|\mathcal{N}|}$, s.t. each π_i is a local solution to Π_i. If $\pi_i^{\triangleright} = \pi_j^{\triangleright}$ for the public action, we call these local solutions equivalent.

3.2.2. Privacy Leakage Analysis

We adopt the privacy leakage metric from [47,48], as we set the number $|\mathcal{V}_i^{priv}| \leq p$ and the size $d = \max_{V \in \mathcal{V}_i^{priv}} |dom(V)|$. The prior information is a tuple:

$$\text{I}_{prio} = \langle \Pi^{\triangleright}, \pi^{\triangleright}, p, b \rangle. \tag{5}$$

The additional information obtained by the adversary is a sequence of messages exchanged between the agents $\mathcal{N} = (n_1, ...n_k)$. After information exchange during the planning process, the posterior information available to the adversary is a tuple:

$$\text{I}_{post} = \langle \Pi^{\triangleright}, \pi^{\triangleright}, p, b, \mathcal{N} \rangle. \tag{6}$$

Considering the transition system of the Π_i, we associate the prior information I_{prio} and I_{post} with variables $\tau(\mathrm{I}_{prio})$ and $\tau(\mathrm{I}_{post})$, which represent the uncertainty of the planning algorithm. So, the final information leakage is computed as:

$$PIL = \log \tau(\mathrm{I}_{prio}) - \log \tau(\mathrm{I}_{post}) = \log \frac{\tau(\mathrm{I}_{prio})}{\tau(\mathrm{I}_{post})}. \tag{7}$$

The upper bound of all transition systems' number is $t^0 = \left(2^{d^2} - 1\right)^p$. After classifying the actions into five categories, i.e., inital -applicable (ia), not-inital-applicable (nia), privately-dependent (pd), privately-independent (pi), privately-nondeterministic (pn) [47], the final information leakage formula is as follows:

$$PIL = \log \frac{\prod_{a^{\triangleright} \in \mathcal{A}^{\triangleright}} \tau_{prio}(a)}{\prod_{a^{\triangleright} \in \mathcal{A}^{\triangleright}} \tau_{post}(a)}. \tag{8}$$

In this paper, we mainly use MAFS algorithms for task planning, and the privacy leakage can be computed as follows: we first reconstruct the search tree, then identify the parent states and applied actions, and classify the actions into five classes (ia, nia, pd, pi, pn). Finally, we compute the information leakage (see Algorithm 1 for details) . Here, the privacy leakage computation with sets of actions can be reformulated as a mixed-integer linear program (MILP) problem with disjunctive constraints.

Algorithm 1: Privacy information leakage analysis based on the MAFS algorithm.

Input: $\mathcal{M} = \{\Pi_i\}_{i=1}^{|\mathcal{N}|}$, number p, and size d
Output: privacy information leakage PIL
reconstruct the search tree based on the MAFS algorithm [30].
  identify possible parent states.
  identify possible applied actions.
classify actions into five classes (ia, nia, pd, pi, pn).
compute privacy information leakage using the Equation (8).
return PIL

For the possible number of the transition system, we construct the following combinatorial optimization problem, which can be solved using the off-the-shelf solver IBM CPLEX [49].

$$\max \quad log(\prod_{a^{\triangleright} \in \mathcal{A}^X} \tau_{post}(a^{\triangleright})) \tag{9}$$

$$s.t. \quad \bigvee_{a^{\triangleright} \in \mathcal{A}^X} \tau_{post}(a^{\triangleright}) \leq t^X, \tag{10}$$

where the $t^X \leq t^0$, action type $X \in \{ia, nia, pd, pi, pn\}$, $\mathcal{A}^X \subseteq \mathcal{A}^{\triangleright}$.

3.3. Observability Controlled Path Planning

In adversarial environments, the observed agents try to control the observation of the adversary by obfuscating their goals. Considering the observation of the adversary in the adversarial setting (mission planning, reconnaissance, etc.), privacy immediately follows from setting with partial observation [28,35,40]. The observability controlled path planning problem is to find a path from the start location to the goal on the navigation map (discrete grid, connected graph, or continuous space representation). So, the discrete path planning problem can be defined as follows:

Definition 7 (Observability controlled path planning)**.** *For every agent $n \in \mathcal{N}$, the observability controlled path planning problem is a tuple [35]:*

$$\Phi = \langle \mathcal{D}, \mathcal{I}, \mathcal{G}, \mathcal{P}, \Omega, \mathcal{O} \rangle \tag{11}$$

- $\mathcal{D} = \langle \mathcal{S}, \mathcal{A}, \mathrm{c} \rangle$ *is the path planning domain,* $\mathcal{S}$ *is a non-empty set of location nodes,* $\mathcal{A} \subseteq \mathcal{S} \times \mathcal{S}$ *is a set of action-related edges,* $\mathrm{c} : \mathcal{E} \mapsto R_0^+$ *returns the cost of traversing each edge.*
- $\mathcal{I} \in \mathcal{S}$ *is the start location and* $g_r \in \mathcal{G}$ *is the real goal;*
- $\mathcal{G} = \{g_r \cup g_0 \cup g_1 \dots\}$ *is a set of candidate goals, where* g_r *is the real goal*
- $\Omega = \{o_i | i = 1, \dots\}$ *is a set of m observations that can be emitted as a result of the action taken and the state transition.*
- $\mathcal{O} : (\mathcal{A} \times \mathcal{S}) \to \Omega$ *is a many-to-one observation function which maps the taken action and the next state reached to an observation in* Ω*.*

In adversarial environments, the adversary will receive the observation sequence associated the actions performed by the observed agent. We could model this process as a one-sensor model, where the adversary maintains one belief space according to the observations. Following the definition of belief space from [35], we take the belief space of the adversary into account in path planning, so as to control the observability of the adversary.

Definition 8. *A belief* b_n *is induced by observation* $\mathcal{O}_i$*, emitted by action* a_i*, resulting in state* $\hat{s}_i$*. The belief state and belief update are defined as:*

$$b_0 = \{\hat{s}_0 | \mathcal{O}(\emptyset, \mathcal{I}) = o_0 \wedge \mathcal{O}(\emptyset, \hat{s}_0) = o_0\}, \tag{12}$$

$$b_{i+1} = update(b_i, o_{i+1}) = \{\hat{s}_{i+1} | \exists \hat{a}, \Gamma(\hat{s}_i, \hat{a}) \models \hat{s}_{i+1} \wedge \hat{s}_i \in b_i \wedge \mathcal{O}(\hat{a}, \hat{s}_{i+1}) = o_{i+1}\}. \tag{13}$$

3.3.1. Path Plan Generation

Decoy Goals: If the adversary is aware of the actor's candidate goals but not the real goal. An observability controlled path plan is to hide the real goal with decoy goals, where $\mathcal{G}_n = \{g_0 \cup \dots \cup g_{n-1}\}$ is the set of decoy goals, the observation sequence should not be biased by any goal in $\mathcal{G}_n$. The objective here is to minimize the privacy leakage by making the adversary's belief space consistent with the decoy goals.

Definition 9. *An observability controlled path plan with decoy goals:*

$$\Gamma(\mathcal{I}, \pi_k) \models g_r \text{ and } |G \in \mathcal{G} : \exists s \in b_n, s| = G| \geqslant m \tag{14}$$

where $m < n$*, for ease of computation, in this paper, we set* $m = 2$*, as in the final stage, path generation will only depend on two goals.*

Observability Controlled Path: Predictability and obfuscation are a pair of incompatible concepts. With the decoy goals selected, patrolling on an observability controlled plan, the observed agent will start with obfuscated steps, but the agent will adopt predictable steps when approaching the goal in the end. So, one observability controlled path is the one with steps that are obfuscated for as long as possible. There is one obfuscated turning point, where all subsequent steps are predictable.

We will employ one probabilistic goal recognition model as the adversary's sensor model.

Definition 10. *An obfuscated turning point is the final state in the observation sequence* $\mathcal{O}_n = \{o_1, \dots o_n\}$*, s.t. the posterior probability of the real goal does not exceed any selected decoy goals, otherwise, the point is predictable to the adversary.*

$$\mathcal{P}(g_r | \mathcal{O}_n) \leq \mathcal{P}(g | \mathcal{O}_n), \forall g \in \mathcal{G}_n \backslash \{g_r\} \tag{15}$$

Definition 11. *A last obfuscated turning point is the last state* π^i *of one given path* π*, which all subsequent states,* $\pi^j, \forall j \in \{i+1, \dots |\pi|\}$ *are predictable to the adversary.*

Here, we mainly focus on the last obfuscated turning point. The observability controlled path plan will cover two parts. As shown in Figure 4, one part of the obfuscated path from the start point to the last obfuscated turning point, and one part of the predictable path from the last obfuscated turning point $(LOTP)$ to the real goal. We can get the strong goal obfuscated path π with continually obfuscated steps to the $LOTP$. Using the cost-difference-based probabilistic goal recognition model introduced in [50], we can get the $LOTP$ after selecting the decoy goals:

$$\text{optc}\,(LOTP, g_r) \approx \frac{\text{optc}\,(g_r, g_d) + \text{optc}\,(s, g_r) - \text{optc}\,(s, g_d)}{2}, \tag{16}$$

where g_d is the selected decoy goal, and $\text{optc}(a, b)$ is the optimal cost from the state a to b. If we adopt discrete grid or graph-based discrete domain representations for path planning, we will approximate the $LOTP$ to the closet state.

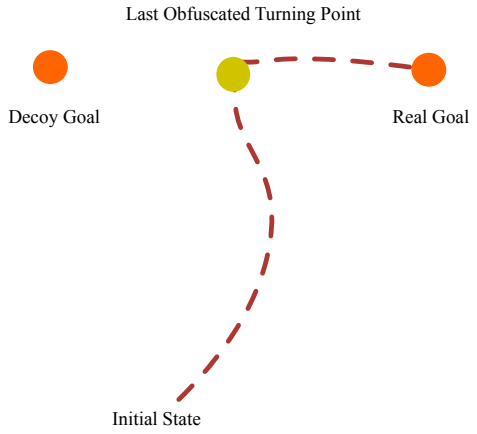


Figure 4. The last obfuscated turning point.

Diverse Path: When the adversary knows the observed agent's goal, in order to control the adversary's observability, we need diverse paths. We can compute the diversity between all the pairs of plans using one plan distance metric mentioned in Appendix A.1. Two plans are a δ distant pair with respect to distance metric d, if $d(p_1, p_2) = \delta$. A path plan set (PPS) induced by plan p starting at $\mathcal{I}$ is minimally δ distant if $\delta = \min_{p1,p2 \in PPS} d(p1, p2)$.

Definition 12. *A plan π_k is k diverse path plan $(k \geq 2)$:*

$$d_{min}(PPS(\mathcal{I}, \pi_k)) \geq \delta \text{ and } |PPS(\mathcal{I}, \pi_k)| \geq k. \tag{17}$$

As a result, if the adversary does not know the real goal, the first part of the path is done by performing a two-decoy-goals path planning. After getting the $LOPT$, we can compute the whole path plan. If the adversary does know the real goal, we need to generate diverse path plans. The details of an observability controlled path plan are given in Algorithm 2.

3.3.2. Privacy Leakage Analysis

Planning with obfuscated goals involves preserving privacy with minimized information leakage. Under the requirement of privacy preservation, the observed agent will deliberately choose misleading actions to obfuscate the goal. We can quantify the information leakage of the states and actions as follows:

Algorithm 2: An observability controlled path plan generation algorithm.

Input: $\Phi = \langle \mathcal{D}, \mathcal{S}, \mathcal{G}, \mathcal{P}, \Omega, \mathcal{O} \rangle$
Output: An observability controlled path plan π
if *GoalUnknown* **then**
  $LOTP$=GetLOTP($\mathcal{I}, \mathcal{G}_r$)
  obfpath=ObfusPath($\mathcal{I}$, $LOTP$)
  intpath=PredPath($LOTP$, $\mathcal{G}_r$)
  return obfpath+intpath
end
else
  divepath=DivePath($\mathcal{I}$, $\mathcal{G}_r$)
  return divepath
end
procedure GetLOTP($\mathcal{I}, \mathcal{G}_r$) ▷ `get the` $LOTP$
procedure ObfusPath($\mathcal{I}$, $LOTP$) ▷ `generate obfuscated path`
procedure PredPath($LOTP$, $\mathcal{G}_r$) ▷ `generate predictable path`
procedure DivePath($\mathcal{I}, \mathcal{G}_r$) ▷ `generate diverse path`

Definition 13 (S-PI). *The privacy information leakage based state privacy information metric is defined as:*

$$\mathcal{I}_{S-PI}(s_j) = H(\max_{g_i \in G \setminus \{g_r\}} P\left(g_i|s_j\right) - P\left(g_r|s_j\right)) = -log(\max_{g_i \in G \setminus \{g_r\}} P\left(g_i|s_j\right) - P\left(g_r|s_j\right)). \quad (18)$$

Definition 14 (A-PI). *As for $a_i \in \mathcal{E}(s)$, $a_j \in \mathcal{E}'(s)$, and $\mathcal{E}'(s) = \mathcal{E}(s) \setminus a_i$. The information leakage based action privacy information metric is defined as:*

$$\mathcal{I}_{A-PI}(a_i) = \frac{\sum_{a_j \in \mathcal{E}'(s)} I_{S-PI}(s)}{|\mathcal{E}'(s)|}. \quad (19)$$

Using the action privacy information metric $\mathcal{I}_{S-PI}(s)$ as additional action cost, we can analyze the privacy leakage of the observability controlled path plan.

4. Experiments

In this section, experiments were conducted for opponent-aware privacy-preserving planning. All the experiments were executed on one Alienware running Ubuntu 16.04 with 4 CPU cores and 8 GB of RAM. We used the MAFS algorithm [30] for information sharing restricted task plan generation. The algorithms for privacy leakage analysis and observability controlled path planning were coded with Python.

4.1. Plan Generation and Privacy Leakage Analysis

Here, we first generate task plans for a robot in the urban security patrol scenario. Then we present three different goal configuration scenarios for path planning. Besides, we analyze the privacy leakage for the task plan and path plan. Finally, we present an indoor robot demonstration using the TurtleBot3 Burger [51].

4.1.1. Task Plan Generation and Privacy Leakage Analysis

As shown in Figure 1, we now define some variables for security patrol scenario. The interaction of the simplified security patrol scenario can be modeled between four agents: two UGVs, one supply center (the malicious teammate), and the hostile observer (opponent).

Variables Definition: For task planning under a cooperative environment, $\mathcal{N} = \{UGV1, UGV2, SC\}$. After patrolling any candidate checkpoint in zone 1, UGV will return to the supply center to charge and transmit collected data, and the task will be completed after patrolling the two zones.

As shown in Table 2, we set the binary variables with T/F values. In the initial state, the supply center has enough supplies, the UGV is charged. In the goal state, the task of the UGV is complete. The following set of variables can be used to describe the task planning problem.

Information Sharing Restricted Task Plan: Here, we simply set UGV1 for zone 1 and UGV2 for zone2. Each UGV will choose two checkpoints to patrol (e.g., checkpoint 1 and 3). The actions of $\mathcal{A}_{UGV1}$ and $\mathcal{A}_{SC}$ can be formulated as shown in Table 3. We provide the action descriptions for UGVs and the supply center in Figure 5.

Table 2. Variables for task planning.

Variable in	Description	Variable	Values	$\mathcal{I}$	$\mathcal{G}$
$\mathcal{V}^{pub}$	UGV1 is charged	cg1	T/F	T	-
	UGV2 is charged	cg2	T/F	T	-
	task1 is complete	tc1	T/F	F	T
	task2 is complete	tc2	T/F	F	T
$\mathcal{V}^{priv}_{UGV1}$	checkpoint 1 is patrolled	cp1	T/F	F	-
	checkpoint 2 is patrolled	cp2	T/F	F	-
	checkpoint 3 is patrolled	cp3	T/F	F	-
	checkpoint 4 is patrolled	cp4	T/F	F	-
	zone 1 is patrolled	zn1	T/F	F	T
$\mathcal{V}^{priv}_{UGV2}$	checkpoint 5 is patrolled	cp5	T/F	F	-
	checkpoint 6 is patrolled	cp6	T/F	F	-
	checkpoint 7 is patrolled	cp7	T/F	F	-
	checkpoint 8 is patrolled	cp8	T/F	F	-
	zone 2 is patrolled	zn2	T/F	F	T
$\mathcal{V}^{priv}_{SC}$	supply center can provide support	sc	T/F	T	-

Table 3. Actions for UGV and supply center.

Action	Description	Label	pre(a)	eff(a)
$\mathcal{A}^{pub}_{UGV1}$	patrol checkpoint 1	PC1	{cg1 = T}	{cp1 = T, cg1 = F}
	patrol checkpoint 3	PC3	{cg1 = T}	{cp3 = T, cg1 = F}
	task1 is complete	TC	{cp1 = T, cp3 = T, tc1 = F}	{zn1 = T, tc = T}
$\mathcal{A}^{pub}_{SC}$	recharge	RC	{sc = T, cg1 = F}	{sc = F, cg1 = T}
	recharge and resupply	RR	{sc = F, cg1 = F}	{sc = T, cg1 = T}

```
(define (domain ugv)
        (:requirements :factored-privacy)
(:constants
        (:private
                cp1 cp3
        )
)
(:predicates
        (ugv-is-charged)
        (task-complete)

        (:private
                (checkpoint-complete ?cp)
        )
)
(:action patrol-checkpoint
        :parameters (?cp)
        :precondition (and
                (ugv-is-charged)
                (not (checkpoint-complete ?cp))
        )
        :effect (and
                (not (ugv-is-charged))
                (checkpoint-complete ?cp)
        )
)
(:action task-complete
        :parameters ()
        :precondition (and
                (checkpoint-complete cp1)
                (checkpoint-complete cp3)
                (not (task-complete))
        )
        :effect (and
                (task-complete)
        )
)
)
```

```
(define (domain ugv)
        (:requirements :factored-privacy)

(:predicates
        (ugv-is-charged)
        (task-complete)

        (:private
                (center-can-support)
        )
)

(:action recharge
        :parameters ()
        :precondition (and
                (not (ugv-is-charged))
                (center-can-support)
        )
        :effect (and
                (ugv-is-charged)
                (not (center-can-support))
        )
)

(:action recharge-and-resupply
        :parameters ()
        :precondition (and
                (not (ugv-is-charged))
                (not (center-can-support))
        )
        :effect (and
                (ugv-is-charged)
                (center-can-support)
        )

)

)
```

Figure 5. Action descriptions for UGV and supply center for the security patrol scenario.

In the following, we will compute the task plan of UGV1 for zone 1 security patrol. The projection of public actions and related transition system are shown in Figure 6. The projection results of the public actions are shown in Table 4. The actions $PC1^{\triangleright}, PC3^{\triangleright} \in \mathcal{A}_{UGV}$, $RC^{\triangleright}, RR^{\triangleright} \in \mathcal{A}_{SC}$ both have an equal projection. We denote them simply as $PC^{\triangleright}$ and $R^{\triangleright}$.

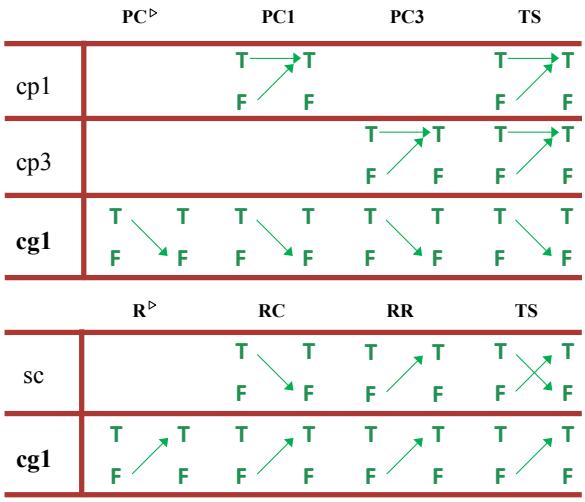


Figure 6. The public projection and related transition systems of the patrol action. The arrows represent the transition for the given variable.

Table 4. Projection the public actions of the UGV and the supply center.

Action	pre(*a*)	eff(*a*)
$PC^{\triangleright}$	{cg1 = T}	{cg1 = F}
$TC^{\triangleright}$	{tc = F}	{tc = T}
$R^{\triangleright}$	{cg1 = F}	{cg1 = T}

We chose MAFS and Secure-MAFS algorithms for task plan generation. The solution of UGV1 to the security patrol scenario is $\pi^{\triangleright}_{UGV1} = \{R, PC, R, PC, TC\}$, which is public to the supply center.

Privacy Leakage Analysis: Then, complete transition is shown in Figure 7. In MAFS, if the state of the UGV is expanded using public action, the resulting public projection state will be sent to the supply center. We analyzed the privacy leakage based on the sent and received states from the UGV.

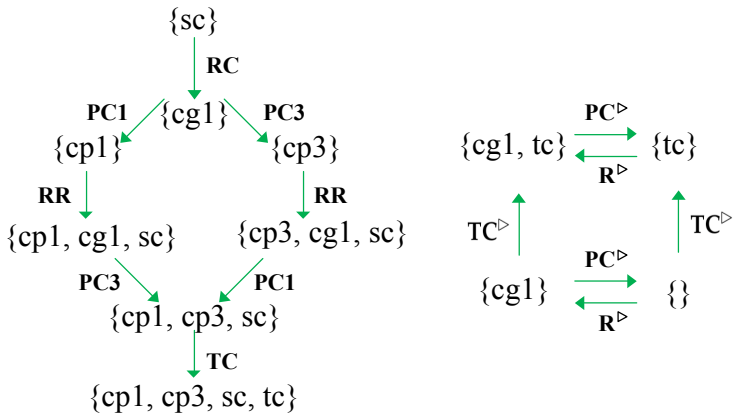
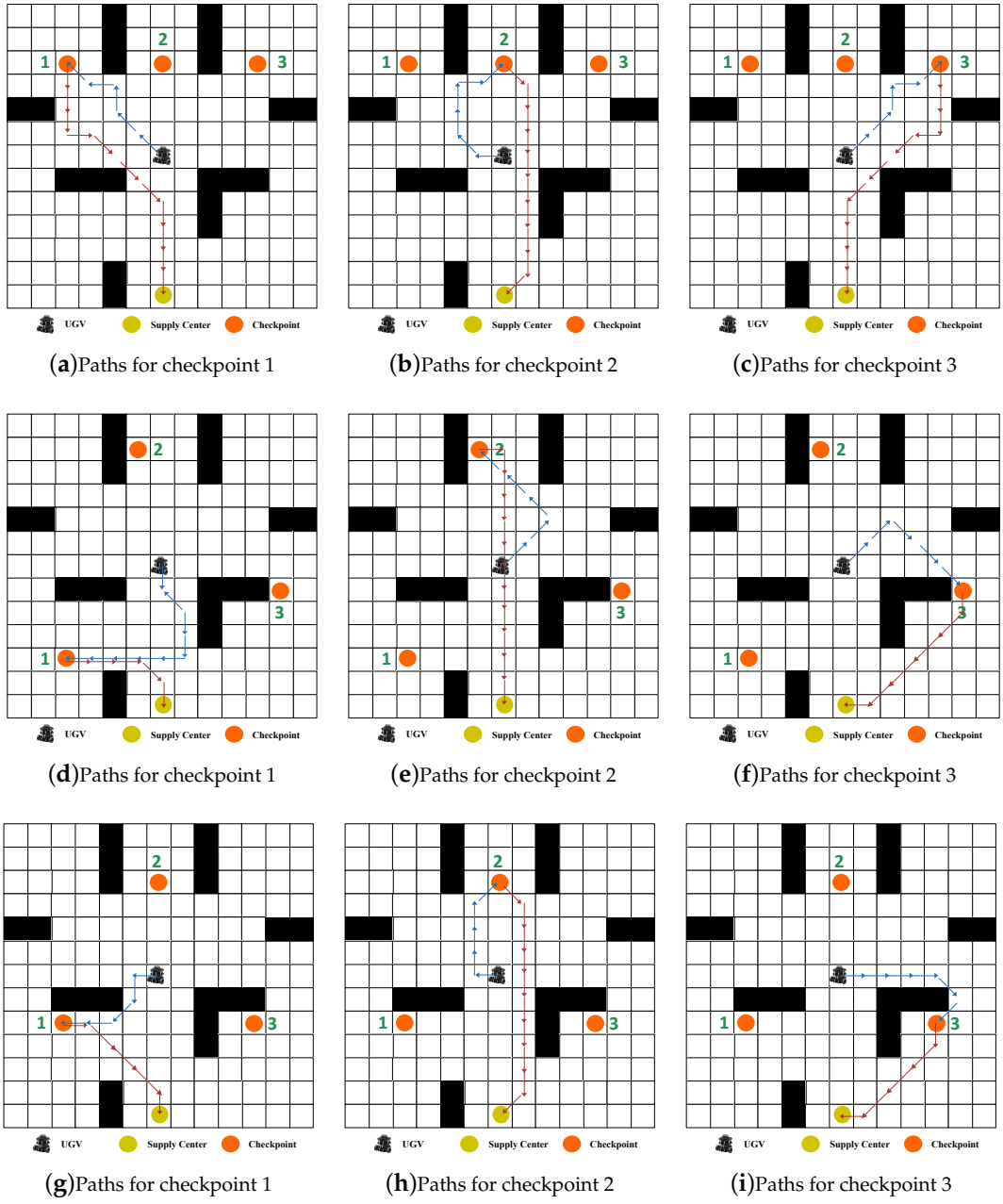


Figure 7. The public projection of actions and the related transition system. The arrows represent transition for the given variable.

An upper bound of the UGV transition system is $t^0 = 15^p$, where $p = |\mathcal{V}^{priv}_{UGV1}|$. After classifying the action types, the $PC^{\triangleright}$ belongs to $\{ia, pi, pn\}$, $TC^{\triangleright}$ belongs to $\{pd\}$. $\tau^{ia}_{PC} = 12^p$, $\tau^{pi}_{PC} = 15^p - 6^p$, $\tau^{pn}_{PC} = 15^p - 8^p$, $\tau^{ia \times pi}_{PC} = 12^p - 3^p$, $\tau^{ia \times pn}_{PC} = 12^p - 6^p$, $\tau^{ia \times pi}_{TC} = 15^p - 3^p$. Using Algorithm 1 with Equation (8), we could compute the privacy information leakage for UGV1: $PIL = \log \tau(\mathrm{I}_{prio}) - \log \tau(\mathrm{I}_{post}) = 10.4 - 9.7 \approx 0.7$.

4.1.2. Path Plan Generation and Privacy Leakage Analysis

Observability Controlled Path Plan: As shown in Figure 8, we used a 13×13 discrete grids based simulation environment with different configurations (line, circular, and triangular) for experimental evaluation. We simply set $m = 2$, $k = 2$, and the UGV patrolled one checkpoint through one observability controlled path and chose one diverse path back to the supply center. For any checkpoint, after choosing the candidate decoy checkpoints, we used Algorithm 2 to generate an observability controlled path.



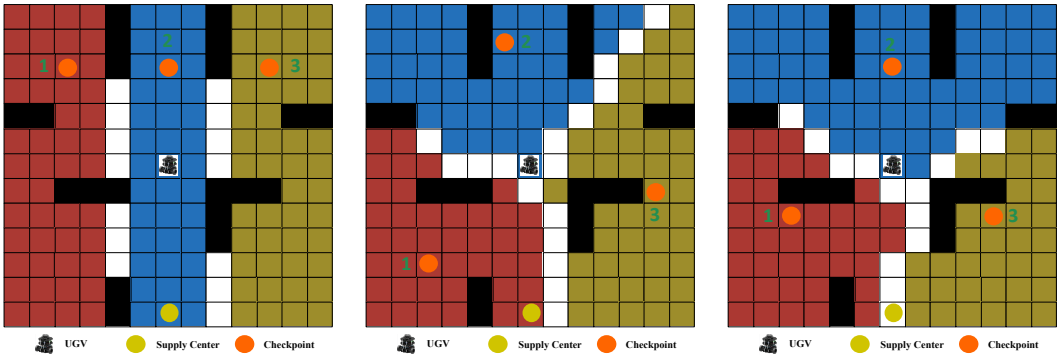
(**a**)Paths for checkpoint 1 (**b**)Paths for checkpoint 2 (**c**)Paths for checkpoint 3

(**d**)Paths for checkpoint 1 (**e**)Paths for checkpoint 2 (**f**)Paths for checkpoint 3

(**g**)Paths for checkpoint 1 (**h**)Paths for checkpoint 2 (**i**)Paths for checkpoint 3

Figure 8. Observability controlled paths to the real checkpoint (blue), and diverse paths back to the supply center (red): (**a**) Line configuration. (**b**) Circular configuration. (**c**) Triangular configuration.

Privacy Leakage Analysis: Following the "single-observation" cost difference based probabilistic goal recognition model from [50], we could pre-compute the cost difference for each state offline to calculate the likelihood that each goal will be the selected checkpoint. As shown in Figure 9, we could create heatmaps for the discrete grids domain, showing the posterior goal probability of each goal at each state. Armed with heatmaps, we could use the state/action privacy information metrics (Equations (13) and (14)) for privacy leakage analysis. The results of the privacy leakage of the paths to each checkpoint under different configurations are shown in Table 5.



(**a**) Line configuration (**b**) Circular configuration (**c**) Triangular configuration

Figure 9. Heatmaps for different configurations.

Table 5. The privacy leakage of the paths to each checkpoint under different configurations.

Configuration	Checkpoint1	Checkpoint2	Checkpoint3
Line	13.5	14.9	13.5
Circular	20.7	10.5	10.5
Triangular	12.3	12.2	14.7

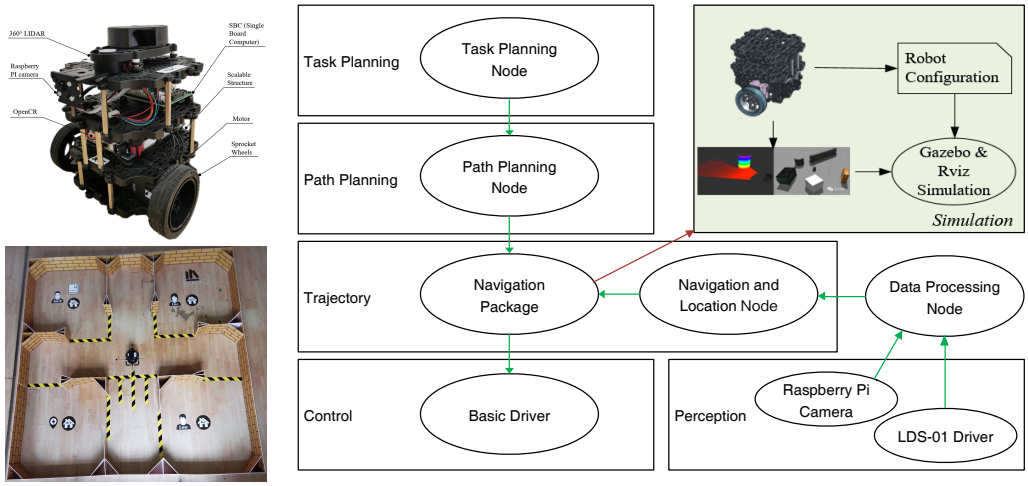
4.2. Indoor Robot Demonstration

To simulate the security patrol scenario with an internal robot and an external human, we used the TurtleBot3 Burger for an indoor robot demonstration. The TurtleBot3 Burger is a mobile robot platform established on ROS (robot operation system). Table 6 shows the configuration. As shown in Figure 10, the TurtleBot3 Burger contains several modules, and we designed the ROS nodes for the software framework and built an experimental scene with four checkpoints. The initial state of the robot is in the middle of the scene.

Table 6. The configuration of TurtleBot3 Burger.

Items	Configuration
Lidar	360-degree laser Lidar LDS-01 (HLS-LFCD2)
SBC	Raspberry PI 3 and Intel Joule 570x
Battery	Lithium polymer 11.1V 1800 mAh
IMU	Gyroscope 3 Axis, Accelerometer 3 Axis, Magnetometer 3 Axis
MCU	OpenCR (32-bit ARM Cortex M7)
Motor	DYNAMIXEL(XL430)

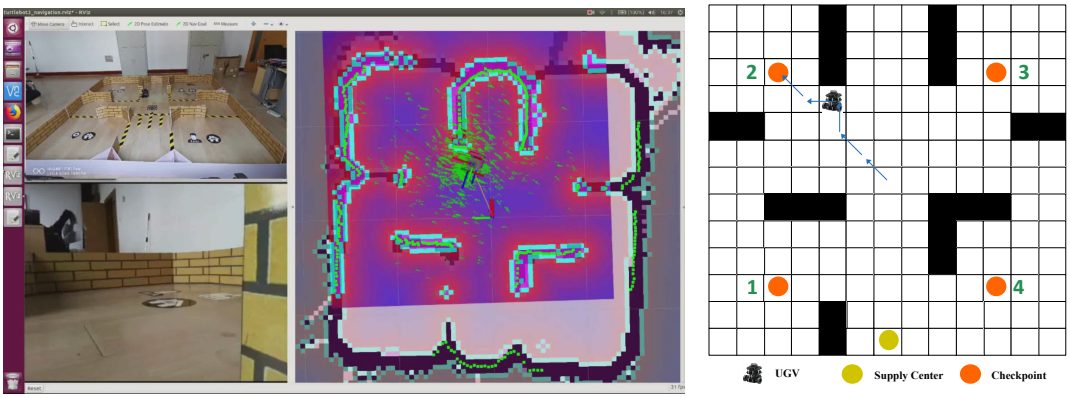
As shown in Figure 11, after generating the information sharing restricted task plan, the robot should generate an observability controlled path plan for checkpoint patrol. As for any checkpoint, the robot will follow the generated path to visit. The trajectories of the robot and the objects in the scene were visualized through RVIZ, and the environment map was built through the Lidar LDS-01.



(**a**)Simulation environment with TurtleBot3 Burger

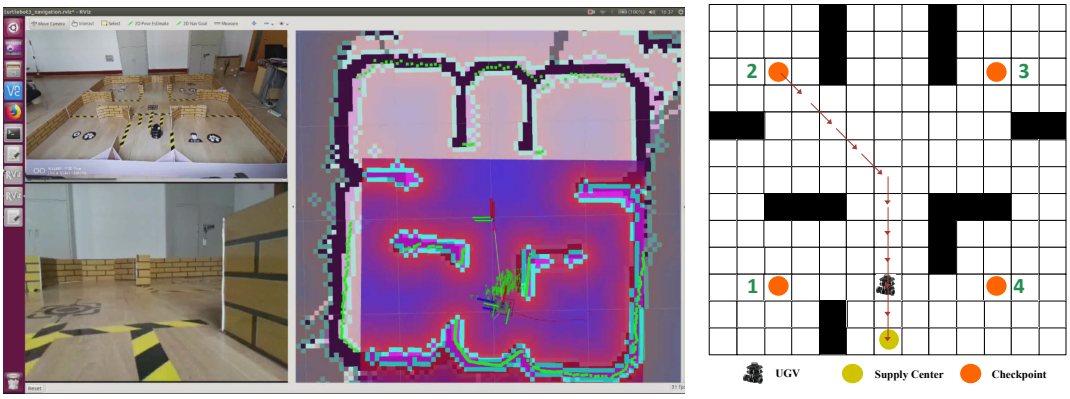
(**b**)Software framework

Figure 10. Indoor simulation environment.



(**a**) Robot on the path to checkpoint 2

(**b**) Path on discrete grids



(**c**) Robot on the path to supply center

(**d**) Path on discrete grids

Figure 11. Indoor robot demonstration: (**a**,**c**) the robot heads to checkpoint 2 with an observability controlled path, and returns to the supply center with a diverse path. (**b**,**d**) The corresponding paths on discrete grids.

5. Conclusions and Future Work

In this paper, the opponent-aware privacy-preserving planning problem in a complex environment is addressed and two questions are answered. Owing to the explosion of privacy preservation in planning, we first define opponent-aware privacy-preserving planning. Then, we present approaches for information sharing restricted task plan generation and observability controlled path plan generation. The final experiments with privacy leakage analysis and the indoor robot demonstration show the applicability of the proposed approaches to generating plans. In fact, many pieces of research have modeled the interaction between patrol UGVs and adversary with Stackelberg or stochastic games, in which agents pursue utility maximization. Additionally, many robust and online goal recognition approaches have been proposed, such as the self-modulating model proposed in [38] for rational and irrational agents. In the future, we will use a stochastic game model with active adversaries to model this problem.

Author Contributions: J.L. and W.Z. proposed the method; X.G., W.G., and Z.L. designed and performed the experiments; J.L. and X.J. analyzed the experimental data and wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grants No. 61702528, No. 61603406.

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this paper.

Abbreviations

The following abbreviations are used in this manuscript:

UGV	Unmanned Ground Vehicle
MAP	Multi-Agent Planning
PIL	Privacy Information Leakage
DMAP	Deterministic MAP
I-POMDPs	Interactive POMDPs
Dec-POMDPs	Decentralized POMDP
MPC	Multi-Party Computation
DCOP	Distributed Constraint Optimization Problems
MAFS	Multi-Agent Forward Search
MADLA	Multi-Agent Distributed and Local Asynchronous
MILP	Mixed Integer Linear Program
PPS	Path Plan Set
LOTP	Last Obfuscated Turning Point
ROS	Robot Operation System

Appendix A. Metrics

Appendix A.1. Plan Distance Metrics

We leverage three alternatives to measure the plan distance and one privacy leakage metric to quantify the information leakage. Three kinds of plan distance metrics have been introduced in [35,52–54]—namely, action distance, causal link distance, and state sequence distance.

Definition A1 (Action distance). *The set of unique actions in a plan π is $A(\pi) = \{a \mid a \in \pi\}$. Given the action sets $A(p_1)$ and $A(p_2)$ of two plans p_1 and p_2, respectively, the action distance can be defined:*

$$d_A(p_1, p_2) = 1 - \frac{|A(p_1) \cap A(p_2)|}{|A(p_1) \cup A(p_2)|}. \tag{A1}$$

Definition A2 (Causal link distance). *$\langle a_i, p_i, a_{i+1}\rangle$ is the tuple form of a causal link, the predicate p_i can be produced as an effect of action a_i and used as a precondition for a_{i+1}. The causal link distance for the causal link sets $C(p_1)$ and $C(p_2)$ of plans p_1 and p_2 can be defined:*

$$d_C(p_1, p_2) = 1 - \frac{|C(p_1) \cap C(p_2)|}{|C(p_1) \cup C(p_2)|}. \tag{A2}$$

Definition A3 (State sequence distance). *Given two state sequence sets $S(p_1) = (s_0^{p_1}, \ldots, s_n^{p_1})$ and $S(p_2) = (s_0^{p_2}, \ldots, s_{n'}^{p_3})$ for p_1 and p_2, respectively, where $n \geq n'$ are the lengths of the plans, $s_k^{p_1}$ is overloaded to denote the set of variables in state s_k of plan p_1, the state sequence distance can be defined:*

$$d_S(p_1, p_2) = \frac{1}{n}\Big[\sum_{k=1}^{n'}\Big(1 - \frac{|s_k^{p_1} \cap s_k^{p_2}|}{|s_k^{p_1} \cup s_k^{p_2}|}\Big) + n - n'\Big]. \tag{A3}$$

References

1. Liu, Y.; Liu, Z.; Shi, J.; Wu, G.; Chen, C. Optimization of Base Location and Patrol Routes for Unmanned Aerial Vehicles in Border Intelligence, Surveillance, and Reconnaissance. *J. Adv. Transp.* **2019**, *2019*, 9063232. [CrossRef]
2. Bell, R.A. Unmanned ground vehicles and EO-IR sensors for border patrol. In *Optics and Photonics in Global Homeland Security III*; International Society for Optics and Photonics: Bellingham, DC, USA, 2007; Volume 6540, p. 65400B.
3. Dwork, C.; Roth, A. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.* **2013**, 9, 211–407. [CrossRef]
4. Wu, F.; Zilberstein, S.; Chen, X. Privacy-Preserving Policy Iteration for Decentralized POMDPs. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
5. Sakuma, J.; Kobayashi, S.; Wright, R.N. Privacy-preserving reinforcement learning. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008. [CrossRef]
6. Liu, Q.; Ren, X.; Mo, Y. Secure and privacy preserving average consensus. In Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and PrivaCy, Dallas, TX, USA, 3 November 2017. [CrossRef]
7. Alaeddini, A.; Morgansen, K.; Mesbahi, M. Adaptive communication networks with privacy guarantees. In Proceedings of the American Control Conference, Seattle, WA, USA, 24–26 May 2017. [CrossRef]
8. Pequito, S.; Kar, S.; Sundaram, S.; Aguiar, A.P. Design of communication networks for distributed computation with privacy guarantees. In Proceedings of the IEEE Conference on Decision and Control, Los Angeles, CA, USA, 15–17 December 2014. [CrossRef]
9. Brafman, R.I. A privacy preserving algorithm for multi-agent planning and search. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Buenos Aires, Argentina, 25–31 July 2015.
10. Štolba, M. Reveal or Hide: Information Sharing in Multi-Agent Planning. Ph.D. Thesis, Czech Technical University in Prague, Prague, Czech Republic, 2017.
11. Tožička, J. Multi-Agent Planning by Plan Set Intersection. Ph.D. Thesis, Czech Technical University in Prague, Prague, Czech Republic, 2017.
12. Zhang, S.; Makedon, F. Privacy preserving learning in negotiation. In Proceedings of the Symposium on Applied Computing, Santa Fe, NM, USA, 13–17 March 2005; pp. 821–825.
13. Shokri, R.; Shmatikov, V. Privacy-preserving deep learning. In Proceedings of the 53rd Annual Allerton Conference on Communication, Control, and Computing, Allerton, IL, USA, 30 September 2015–2 October 2015. [CrossRef]
14. Léauté, T.; Faltings, B. Protecting privacy through distributed computation in multi-agent decision making. *J. Artif. Intell. Res.* **2013**, *47*, 649–695. [CrossRef]
15. Grinshpoun, T. A Privacy-Preserving Algorithm for Distributed Constraint Optimization. In Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), Paris, France, 5–9 May 2014.

16. Tassa, T.; Zivan, R.; Grinshpoun, T. Max-sum goes private. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Buenos Aires, Argentina, 25–31 July 2015.
17. Štolba, M.; Tožička, J.; Komenda, A. Secure Multi-Agent Planning. In Proceedings of the 1st International Workshop on AI for Privacy and Security - PrAISe '16, The Hague, The Netherlands, 29–30 August 2016; pp. 1–8. [CrossRef]
18. Agmon, N.; Kaminka, G.A.; Kraus, S. Multi-Robot Adversarial Patrolling: Facing a Full-Knowledge Opponent. *J. Artif. Intell. Res.* **2014**, *42*, 887–916.
19. Brafman, R.I.; Domshlak, C. From One to Many: Planning for Loosely Coupled Multi-Agent Systems. In Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS), Sydney, Australia, 14–18 September 2008.
20. Torreño, A.; Onaindia, E.; Sapena, Ó. FMAP: Distributed cooperative multi-agent planning. *Appl. Intell.* **2014**, *41*, 606–626. [CrossRef]
21. Bonisoli, A.; Gerevini, A.E.; Saetti, A.; Serina, I. A privacy-preserving model for the multi-agent propositional planning problem. In Proceedings of the 2nd ICAPS Distributed and Multi-Agent Planning workshop (ICAPS DMAP-2014), Portsmouth, NH, USA, 22 June 2014. [CrossRef]
22. Decker, K.S.; Lesser, V.R. Generalizing the partial global planning algorithm. *Int. J. Intell. Coop. Inf. Syst.* **1992**, *1*, 319–346. [CrossRef]
23. Borrajo, D. Multi-Agent Planning by Plan Reuse. In Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems, St. Paul, MN, USA, 6–10 May 2013.
24. Maliah, S.; Shani, G.; Stern, R. Stronger Privacy Preserving Projections for Multi-Agent Planning. In Proceeding of the International Conference on Automated Planning and Scheduling (ICAPS), London, UK, 12–17 June 2016.
25. Komenda, A.; Tožička, J.; Štolba, M. ϵ-strong privacy preserving multi-agent planning. In *Lecture Notes in Computer Science*; including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics; Springer Science + Business Media: Berlin, Germany, 2018._8. [CrossRef]
26. Beimel, A.; Brafman, R.I. Privacy Preserving Multi-Agent Planning with Provable Guarantees. *arXiv* **2018**, arXiv:1810.13354.
27. Goldreich, O. *Foundations of Cryptography*; Cambridge University Press: Cambridge, UK, 2010. [CrossRef]
28. Kulkarni, A.; Klenk, M.; Rane, S.; Soroush, H. Resource Bounded Secure Goal Obfuscation. In Proceeding of the AAAI Fall Symposium on Integrating Planning, Diagnosis and Causal Reasoning, Arlington, VA, USA, 18–20 October 2018.
29. Chakraborti, T.; Kulkarni, A.; Sreedharan, S.; Smith, D.E.; Kambhampati, S. Explicability? legibility? predictability? transparency? privacy? security? the emerging landscape of interpretable agent behavior. In Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS), Berkeley, CA, USA, 11–15 July 2019.
30. Nissim, R.; Brafman, R. Distributed heuristic forward search for multi-agent planning. *J. Artif. Intell. Res.* **2014**, *51*, 293–332. [CrossRef]
31. Lindell, Y.; Pinkas, B. Secure Multiparty Computation for Privacy-Preserving Data Mining. *J. Priv. Confid.* **2018**. [CrossRef]
32. Torreño, A.; Onaindia, E.; Komenda, A.; Štolba, M. Cooperative multi-agent planning: A survey. *ACM Comput. Surv. (CSUR)* **2018**, *50*, 84. [CrossRef]
33. Panella, A.; Gmytrasiewicz, P. Bayesian learning of other agents' finite controllers for interactive POMDPs. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
34. Oliehoek, F.A.; Amato, C. *A Concise Introduction to Decentralized POMDPs*; Springer International Publishing: Cham, Switzerland, 2016; Volume 1.
35. Kulkarni, A.; Srivastava, S.; Kambhampati, S. A unified framework for planning in adversarial and cooperative environments. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019.
36. Keren, S.; Gal, A.; Karpas, E. Privacy preserving plans in partially observable environments. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), New York, NY, USA, 9–15 July 2016.
37. Masters, P.; Sardina, S. Deceptive path-planning. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Melbourne, Australia, 19–25 August 2017.

38. Masters, P.; Sardina, S. Goal recognition for rational and irrational agents. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, Montreal, QC, Canada, 13–17 May 2019; pp. 440–448.
39. Root, P.J. Collaborative UAV Path Planning with Deceptive Strategies. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2005.
40. Keren, S.; Gal, A.; Karpas, E. Goal Recognition Design for Non-Optimal Agents. In Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling, Portsmouth, NH, USA, 21–26 June 2014.
41. Strouse, D.; Kleiman-Weiner, M.; Tenenbaum, J.; Botvinick, M.; Schwab, D.J. Learning to share and hide intentions using information regularization. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2018; pp. 10249–10259.
42. Le Guillarme, N. A Game-Theoretic Planning Framework for Intentional Threat Assessment. Ph.D. Thesis, Thèse de Doctorat, Université de Caen, Caen, France, 2016.
43. Shen, M.; How, J.P. Active Perception in Adversarial Scenarios using Maximum Entropy Deep Reinforcement Learning. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019.
44. Štolba, M.; Komenda, A. Relaxation heuristics for multiagent planning. In Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling, Portsmouth, NH, USA, 21–26 June 2014.
45. Smith, G. On the foundations of quantitative information flow. In *Lecture Notes in Computer Science*; including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics; Springer Science + Business Media: Berlin, Germany, 2009._21. [CrossRef]
46. Rényi, A. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*; The Regents of the University of California: Los Angeles, CA, USA, 1961.
47. Štolba, M.; Tožička, J.; Komenda, A. Quantifying privacy leakage in multi-agent planning. *ACM Trans. Internet Technol. (TOIT)* **2018**, *18*, 28. [CrossRef]
48. Štolba, M.; Fišer, D.; Komenda, A. Privacy Leakage of Search-Based Multi-Agent Planning Algorithms. In Proceedings of the International Conference on Automated Planning and Scheduling, Berkeley, CA, USA, 11–15 July 2019; Volume 29, pp. 482–490.
49. IBM CPLEX. Available online: http://www.ibm.com/us-en/marketplace/ibm-ilog-cplex (accessed on 1 March 2019).
50. Masters, P.; Sardina, S. Cost-based goal recognition in navigational domains. *J. Artif. Intell. Res.* **2019**, *64*, 197–242. [CrossRef]
51. TurtleBot3. Available online: https://www.turtlebot.com (accessed on 1 August 2019).
52. Srivastava, B.; Nguyen, T.A.; Gerevini, A.; Kambhampati, S.; Do, M.B.; Serina, I. Domain independent approaches for finding diverse plans. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Hyderabad, India, 6–12 January 2007.
53. Nguyen, T.A.; Do, M.; Gerevini, A.E.; Serina, I.; Srivastava, B.; Kambhampati, S. Generating diverse plans to handle unknown and partially known user preferences. *Artif. Intell.* **2012**, *190*, 1–31. [CrossRef]
54. Bryce, D. Landmark-based plan distance measures for diverse planning. In Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling, Portsmouth, NH, USA, 21–26 June 2014.