MV-FAC: Mean–Variance Value Function Factorization for Multi-Robot Mean–Standard Deviation Moving Target Search

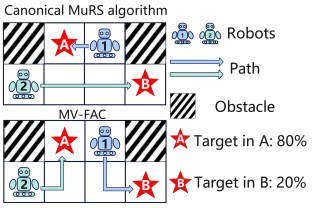
Anonymous Submission

Abstract

This paper studies a risk-sensitive formulation of the multi-robot search problem, termed multi-robot mean–standard deviation search (MuRMSS), in which a team of robots cooperatively search for a moving target by minimizing a linear combination of the mean and standard deviation of search time. However, the standard deviation term is inherently non-additive, making it difficult to estimate, incompatible with canonical multi-robot search algorithms, and preventing consistent decomposition into individual robot utilities, which is essential for scalable multi-robot cooperation. In view of these challenges, we propose MV-FAC, which comprises a mean–variance temporal-difference module that jointly learns the mean and variance of search time, a factorization module that decomposes them into individual utilities, and a decentralized policy optimization module that minimizes each robot's individual mean–std objective. We further establish and prove the mean–std individual-global minimization (MS-IGM) theorem, thereby ensuring consistency between individual- and team-level objectives. Extensive simulation studies on standard multi-robot search benchmarks demonstrate that MV-FAC achieves the best overall mean–std search-time performance. We also validate MV-FAC's practicality by deploying it on a physical multi-robot system for moving target search in a real-world building environment.

1 Introduction

Multi-robot search (MuRS) has long been an active research field, attracting sustained attention from both academia and industry. In practice, MuRS underpins a wide range of real-world applications, such as search and rescue in hazardous environments, persistent surveillance, and exploration in unknown environments. Beyond its practical relevance, MuRS also serves as a challenging and representative testbed for both algorithmic development and theoretical analysis in multi-agent systems, enabling the systematic study of core problems across multi-agent reinforcement learning, distributed control, and game-theoretic decision making.



(a) A Simple Scenario

Algorithms	Output	Mean+Std
Canonical MuRS Alg.	1→A 2→B	1.4+0.8 =2.2
MV-FAC	1→B 2→A	2+0 =**2.0**

(b) Performance Comparison

Figure 1: (a) A simple yet illustrative example of MuRMSS: two robots cooperate to search for one stationary target residing in Grid A with probability of 80% or in Grid B with probability of 20%, with the objective of minimizing the mean–std of search time. (b) Performance comparison: (1) Canonical MuRS algorithms dispatch Robot 1 to Grid A and Robot 2 to Grid B. The resulting mean search time is 1.4, with the variance 0.64, yielding a mean–std search time at $1.4 + \sqrt{0.64} = 2.2$. (2) MV-FAC dispatches Robot 1 to Grid B and Robot 2 to Grid A. The mean search time is 2, and the variance is 0, resulting in the mean–std search time at $2 + \sqrt{0} = 2$.

In the existing literature, multi-robot search has been studied under a variety of problem settings. Among them, a prevalent line of work focuses on minimizing the expected search time. While such expectation-based objectives are intuitive and convenient, they fail to capture the variability of search outcomes. In many risk-sensitive applications, it is preferable to sacrifice a small amount of expected search time for the reduced search-time variance. In view of these considerations, we introduce a risk-sensitive multi-robot search problem, termed multi-robot mean–standard deviation search (MuRMSS), which aims at minimizing the linear combination of mean and standard deviation (mean–std) search time.

However, MuRMSS alters the underlying problem structure, causing the canonical multi-robot search algorithms to yield suboptimal solutions. As illustrated in Fig. 1a, even in a simple scenario, algorithms designed to minimize the expected search time are no longer optimal under the mean–std search objective. Moreover, the standard-deviation term is inherently nonlinear and non-additive, which complicates its modeling and estimation procedure, and prevents the effective decomposition into individual robot utilities for decentralized decision making.

To address the aforementioned challenges, we propose mean–variance value function factorization (MV-FAC), a fac-

torized multi-agent reinforcement learning (MARL) method for the MuRMSS problem. MV-FAC consists of (1) a mean–variance temporal difference (TD) learning module, which jointly estimates the team-level mean and variance of search time, thereby circumventing the need to directly model the non-additive standard-deviation term in MuRMSS; (2) a mean–variance factorization module, which simultaneously decomposes the team-level mean and variance into corresponding individual-level utilities for each robot; and (3) a decentralized policy optimization module, which enables each robot to update its decision-making policy towards minimizing the individual-level mean–std objective. Furthermore, we establish the mean–std individual-global minimization (MS-IGM) theorem, which guarantees consistency between individual- and team-level mean–std objectives. We evaluate MV-FAC against state-of-the-art methods on standard multi-robot search benchmarks, and further validate its effectiveness through deployment on a physical multi-robot system for moving-target search in a real-world building environment.

The paper's main contributions can be summarized as follows: (1) We formulate a risk-sensitive multi-robot search problem, termed MuRMSS, which explicitly incorporates both the mean and standard deviation into the search-time objective. (2) We propose MV-FAC for the MuRMSS problem, which jointly learns and factorizes the team-level mean and variance of search time into corresponding individual-level utilities, thereby enabling decentralized policy optimization. (3) We establish and prove the MS-IGM theorem, which guarantees local-to-global consistency between individual- and team-level mean–std objectives.

2 Literature Review

This section reviews related work on multi-robot search from the perspectives of problem settings and solution methodologies, and further discusses representative studies on risk-sensitive multi-agent decision making.

2.1 Problem Settings in Multi-Robot Search

Multi-robot search has been studied under various problem settings, among which multi-robot efficient search (MuRES), multi-robot adversarial search (MuRAS), and multi-robot guaranteed search (MuRGS) are the most representative.

1) Multi-robot efficient search (MuRES): MuRES is the most classical and widely studied problem setting in multi-robot search. In this setting, the target is assumed to be non-adversarial, *i.e.*, its motion dynamics are independent of the robot team's search strategy, and the objective is to minimize the expected search time. Within the MuRES setting, existing studies have explored a variety of modeling variations along several dimensions, including environment representations such as grid-based [Lee and Lee, 2024], graph-based [Patil *et al.*, 2023], and continuous environments [Kwa *et al.*, 2020]; sensor characteristics such as perfect detection [Sheng *et al.*, 2022] and imperfect sensing with false negatives [Asfora *et al.*, 2020] and/or false positives [Sun *et al.*, 2020]; sensing ranges including same-node detection [Guo *et al.*, 2023a], neighbor-node detection [Chen *et al.*, 2024], and line-of-sight sensing [Cui *et al.*, 2021]; and robot team compositions such as homogeneous robot teams [Andreychuk *et al.*, 2025] and heterogeneous robot teams [Jo and Son, 2025].

2) Multi-robot adversarial search (MuRAS): MuRAS extends the classical multi-robot search problem setting by explicitly modeling the target as an adversarial one, which acts adversarially against the robot team's search strategy [Rahman *et al.*, 2022; Peng *et al.*, 2025a]. In this setting, the robot team still aims to minimize the expected search time, however, the target actively attempts to evade the robots in order to delay detection. Due to the adversarial nature of target motion, MuRAS is often studied under the pursuit–evasion framework [Fang *et al.*, 2022; Olsen *et al.*, 2022].

3) Multi-robot guaranteed search (MuRGS): MuRGS considers a more conservative problem setting in which the robot team seeks to guarantee target detection without making assumptions about the target's motion characteristics [Kolling and Kleiner, 2013]. Unlike MuRES and MuRAS, MuRGS typically adopts the worst-case performance criterion, aiming to minimize the maximum possible search time regardless of the target's motion behavior and initial location. As a result, MuRGS is commonly studied in settings where safety-critical requirements necessitate strict upper bounds on search time.

This paper introduces MuRMSS, which defines a different problem setting from MuRES, MuRAS and MuRGS. MuRMSS follows the non-adversarial moving target assumption in MuRES, but departs from the existing formulations by adopting a risk-sensitive objective that minimizes the mean–standard deviation of search time. The risk-sensitive objective introduces additional challenges due to the non-additivity of the standard-deviation term, making classical multi-robot search methods insufficient for the problem setting.

2.2 Methodologies for Multi-Robot Search

Existing methodologies for multi-robot search can be broadly categorized into four groups: planning-based methods, swarm intelligence methods, game-theoretic methods, and multi-agent reinforcement learning (MARL)-based methods.

1) Planning-based methods represent the most canonical solutions for the MuRES problem. They typically formulate multi-robot search as a mathematical optimization problem by explicitly modeling target motion, initial location uncertainty, and robot dynamics, and solve it using off-the-shelf optimization solvers [Asfora *et al.*, 2020; Gonzalez and Jaillet, 2025; Shree *et al.*, 2021]. However, these methods suffer from poor scalability, as the computational complexity grows rapidly with the number of robots, target uncertainty, and environment size.

2) Swarm intelligence methods address the multi-robot search problem by prescribing simple local behavior or interaction rules for individual robots, from which team-level search behaviors emerge through decentralized execution [Yuan *et al.*, 2024; Sharma *et al.*, 2025; Morin *et al.*, 2022]. While intuitive, easy to implement, and highly scalable, they lack an explicit objective-driven problem formulation, making it difficult to relate local rules to the global objective and systematically optimize team-level performance.

3) Game-theoretic methods model multi-robot search as an interactive decision-making process between the robot team and an adversarially moving target, and typically formu-

late the problem within the zero-sum or pursuit–evasion game framework to seek equilibrium-based solutions [Zhou *et al.*, 2024b; Esfahani *et al.*, 2026] or derive sufficient conditions for guaranteed search success [Bone *et al.*, 2023]. However, such methods often rely on restrictive assumptions about the target's capabilities and/or environment properties, and face scalability challenges in complex settings.

4) MARL-based methods have recently emerged as a promising approach for the MuRES problem by formulating multi-robot search as a decentralized sequential decision-making process [Calzolari *et al.*, 2025; Tan *et al.*, 2021; Hou *et al.*, 2024; Zhou *et al.*, 2024a]. These methods typically cast MuRES within the decentralized partially observable Markov decision process (Dec-POMDP) framework and employ distributed policy optimization algorithms, such as multi-agent policy gradient [Chen *et al.*, 2025; Guo *et al.*, 2023a] or value-function factorization [Guo *et al.*, 2025], to learn cooperative search strategies from the interaction data.

2.3 Risk-Sensitive Multi-Agent Decision Making

Risk-sensitive multi-agent decision making has recently attracted increasing attention, aiming to extend beyond canonical expectation-based objectives by explicitly accounting for uncertainty and variability in long-term returns. Representative studies focus on distribution-based risk measures, such as value-at-risk (VaR), conditional value-at-risk (CVaR), and distortion risk measures (DRMs), and develop corresponding value factorization frameworks to enable decentralized execution under these risk-sensitive objectives [Slumbers *et al.*, 2022; Gao *et al.*, 2021; Shen *et al.*, 2023; Jiang *et al.*, 2025]. In this context, it has been shown that classical expectation-based IGM principles are generally insufficient, motivating generalized consistency conditions and quantile-based factorization architectures for non-additive risk operators.

Despite the progress, existing risk-sensitive multi-agent decision-making methods primarily target distributional risk objectives, which require learning and manipulating the *full* return distribution and tend to incur substantial computational complexity. In contrast, the MuRMSS problem adopts a mean–standard deviation objective, where risk sensitivity is characterized by first- and second-order statistics rather than full distributional modeling, motivating a more lightweight algorithmic design that only estimates and factorizes the mean and variance of search time. A detailed review of risk-sensitive MARL is presented in the supplementary material.

3 Problem Formulation

This section presents the problem formulation of MuRMSS and its modeling within the decentralized partially observable Markov decision process (Dec-POMDP) framework. For clarity, a summary of major notations used throughout the paper is provided in the supplementary material.

3.1 Task-Level Problem Definition

The task-level definition of MuRMSS specifies the operating environment, the target motion dynamics, the robots' sensing and motion models, and the target detection condition.

1) Operating Environment: The environment is modeled as an undirected and connected unit-cost graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of nodes, and $\mathcal{E}$ denotes the set of edges. The unit-cost assumption means that traversing any edge or remaining at the same node incurs one time step for both the robots and the target. This assumption has been commonly adopted in the multi-robot search literature for discrete environments, see [Guo *et al.*, 2025; Asfora *et al.*, 2020; Sheng *et al.*, 2022] as examples. In practice, non-unit-cost graphs can be converted into unit-cost ones by subdividing long edges into several unit-cost sub-edges.

2) Target Motion Dynamics: The target's position at time step t is denoted by $e_t \in \mathcal{V}$, and is unknown to the robot team. The target's motion is assumed to be non-adversarial, meaning that the motion dynamics are *independent* of the robot team's search strategy. The target motion is modeled as a discrete-time Markov process governed by the stochastic transition matrix $\mathbf{\Gamma}$, *i.e.*, $\mathbb{P}[e_{t+1}|e_t] = \mathbf{\Gamma}(e_t, e_{t+1})$. Here, $\mathbf{\Gamma}$ respects the underlying graph $\mathcal{G}$, such that $e_{t+1} = e_t$ or $(e_t, e_{t+1}) \in \mathcal{E}$, and is also unknown to the robot team.

3) Robot Model: We consider a team of N robots indexed by $i \in \{1, \dots, N\}$. Robot i's position at time step t is denoted by $p_t^{(i)} \in \mathcal{V}$, and at each time step, robot i can either move to an adjacent node in $\mathcal{G}$, *i.e.*, $\big(p_t^{(i)}, p_{t+1}^{(i)}\big) \in \mathcal{E}$ or remain stationary at the current node, *i.e.*, $p_{t+1}^{(i)} = p_t^{(i)}$. Each robot is equipped with a local sensor and receives a binary observation $o_t^{(i)} \in \{0, 1\}$, where $o_t^{(i)} = 1$ indicates that the target is detected at node $p_t^{(i)}$, while $o_t^{(i)} = 0$ for no target detection at node $p_t^{(i)}$. The decision-making policy of robot i, denoted by $\pi^{(i)}$, depends on its own history of positions and observations, *i.e.*, $\pi^{(i)} = \pi^{(i)}\big(p_{\le t}^{(i)}, o_{\le t}^{(i)}\big)$.

4) Target Detection Condition: The target is deemed to be detected if, at any time step t, there exists at least one robot i such that the robot and the target reside at the same node, *i.e.*, $p_t^{(i)} = e_t$. The time step at which this event first occurs is defined as the search time and is denoted by t_{cap}. Here, t_{cap} is a random variable due to the stochastic target motion and the uncertainty in the target's initial location.

3.2 Dec-POMDP Modeling for MuRMSS

Based on the task-level definition in Section 3.1, we formulate MuRMSS as a decentralized partially observable Markov decision process (Dec-POMDP). A Dec-POMDP is defined by the tuple $\langle \mathcal{I}, \mathcal{S}, \{\mathcal{A}^{(i)}\}, \{\mathcal{O}^{(i)}\}, \mathbb{P}, R, \gamma \rangle$, and each element is specified within the multi-robot search context.

The set of agents is given by $\mathcal{I} = \{1, \dots, N\}$, where N is the number of searching robots. The global state $s_t \in \mathcal{S}$ is defined as the joint configuration of the target and all robots at time t, *i.e.*, $\boldsymbol{s_t} = (e_t, p_t^{(1)}, \dots, p_t^{(N)})$. Each robot $i \in \mathcal{I}$ selects an action $a_t^{(i)} \in \mathcal{A}^{(i)}$, which corresponds to moving to one of the neighboring nodes or remaining stationary, yielding the joint action $\boldsymbol{a}_t = (a_t^{(1)}, \dots, a_t^{(N)})$. Each robot i receives a local observation $o_t^{(i)} \in \{0, 1\}$, indicating whether the target is detected at its current node. Each robot follows a decentralized policy $\pi^{(i)}$ based on its local position-observation history.

The state transition function $\mathbb{P}$ is induced by the robots'

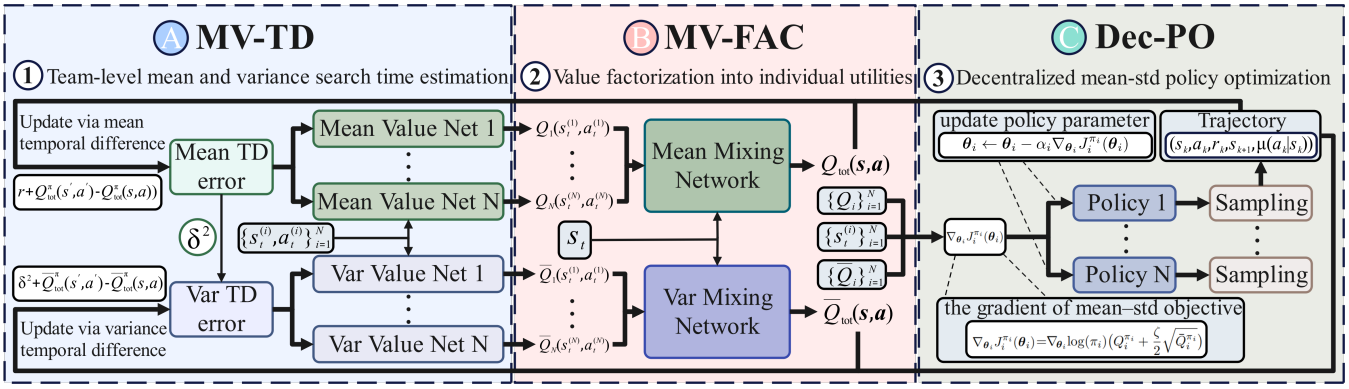


Figure 2: The MV-FAC Framework: Ⓐ MV-TD estimates the team-level mean and variance value functions of search time; (2) Ⓑ MV-FAC decomposes the team-level mean and variance value functions into individual-level utilities with the monotonic mixing networks; and (3) Ⓒ Dec-PO updates each robot's decision-making policy based on the mean–std policy gradient.

joint actions and the target's stochastic motion dynamics. Robot motion is deterministic given the action and graph connectivity, whereas the target motion is stochastic and independent of the robots' behavior. The instantaneous reward r_t is set to 1 at each time step until termination, corresponding to the unit time cost. We consider an undiscounted episodic setting with $\gamma = 1$, under which the cumulative return equals t_{cap}. Unlike standard expectation-based formulations, MuRMSS adopts a risk-sensitive objective that minimizes the mean–standard deviation of t_{cap} under the joint policy $\boldsymbol{\pi}$, *i.e.*, $\min_{\boldsymbol{\pi}} \mu(t_{\text{cap}}) + \zeta\sigma(t_{\text{cap}})$, where $\mu(t_{\text{cap}})$ and $\sigma(t_{\text{cap}})$ denote the mean and standard deviation of t_{cap}, respectively, and $\zeta > 0$ is a risk-sensitivity coefficient.

4 The MV-FAC Framework

This section presents the mean–variance value function factorization (MV-FAC) framework to solve the MuRMSS problem. Unlike the distributional risk-sensitive MARL methods designed for VaR-, CVaR-, or distortion-based objectives, MV-FAC directly targets the mean–std objective by estimating and factorizing only the mean and variance of search time, resulting in a lightweight and scalable design.

The MV-FAC framework consists of three coupled components that form an 'estimation-factorization-optimization' loop: (1) a mean–variance temporal difference (MV-TD) module that *estimates* the mean and variance of search time under a given joint policy; (2) a mean–variance value function factorization (MV-FAC) module that *factorizes* the team-level mean and variance into individual utilities; and (3) a decentralized policy optimization (Dec-PO) module that *optimizes* each robot's policy based on the factorized mean–variance objective. We further provide MV-FAC's pseudocode along with its computational complexity analysis. An overview of the MV-FAC framework is presented in Fig. 2.

4.1 Mean–Variance Temporal Difference (MV-TD)

The MV-TD module estimates the mean and variance of search time under a given joint policy $\boldsymbol{\pi}$ by extending standard TD learning to the use case of second-order statistics. We first introduce the mean–variance Bellman equation set, and then present the corresponding TD methods for online model-free estimation of these quantities.

The team-level mean action-value function $Q^{\pi}_{\text{tot}}(s,a)$, which corresponds to the expected search time when taking the joint action a at global state s, under policy $\boldsymbol{\pi}$ satisfies the following *mean* Bellman equation:

$$Q^{\pi}_{\text{tot}}(s,a) = \sum_{s' \in \mathcal{S}} \mathbb{P}(s'|s,a) \Big[r(s,a,s') + \mathbb{E}_{\pi} Q^{\pi}_{\text{tot}}(s',a') \Big]. \quad (1)$$

Correspondingly, the team-level variance action-value function $\bar{Q}^{\pi}_{\text{tot}}(s,a)$, satisfies the *variance* Bellman equation:

$$\bar{Q}^{\pi}_{\text{tot}}(s,a) = \sum_{s' \in \mathcal{S}} \mathbb{P}(s'|s,a) \sum_{a' \in \mathcal{A}} \pi(a'|s') \big[\delta^2 + \bar{Q}^{\pi}_{\text{tot}}(s',a') \big], \quad (2)$$

where $\delta = r(s,a,s') + Q^{\pi}_{\text{tot}}(s',a') - Q^{\pi}_{\text{tot}}(s,a)$ refers to the sampled TD error. Note that the derivation process of the variance Bellman equation (Eq. (2)) makes use of the law of total variance in probability theory, and is presented in the supplementary material.

The mean–variance Bellman equation set, *i.e.*, Eq. (1) and Eq. (2), provides the theoretical foundation for estimating the mean and variance of the return. Building on this foundation, we develop the mean–variance temporal difference (MV-TD) method to estimate these quantities in an online, model-free manner. Given a transition tuple $\langle s, a, r, s', a' \rangle$ sampled according to the joint policy π, the estimates of $Q^{\pi}_{\text{tot}}(s,a)$ and $\bar{Q}^{\pi}_{\text{tot}}(s,a)$ are recursively updated as follows:

$$Q^{\pi}_{\text{tot}}(s,a) \leftarrow Q^{\pi}_{\text{tot}}(s,a) + \alpha \big(r + Q^{\pi}_{\text{tot}}(s',a') - Q^{\pi}_{\text{tot}}(s,a) \big) \quad (3)$$

$$\bar{Q}^{\pi}_{\text{tot}}(s,a) \leftarrow \bar{Q}^{\pi}_{\text{tot}}(s,a) + \bar{\alpha} \big(\delta^2 + \bar{Q}^{\pi}_{\text{tot}}(s',a') - \bar{Q}^{\pi}_{\text{tot}}(s,a) \big), \quad (4)$$

where α and $\bar{\alpha}$ are the step-size parameters for the mean and variance value functions, respectively.

Note that both the mean–variance Bellman equations (Eqs. (1)–(2)) and the corresponding mean–variance TD updates (Eqs. (3)–(4)) are formulated primarily for estimating the team-level *action*-value functions, *i.e.*, $Q^{\pi}_{\text{tot}}(s,a)$ and $\bar{Q}^{\pi}_{\text{tot}}(s,a)$. In practice, these formulations can be naturally adapted with minor modifications to estimate the corresponding *state*-value functions, *i.e.*, $V^{\pi}_{\text{tot}}(s)$ and $\bar{V}^{\pi}_{\text{tot}}(s)$, as detailed in the supplementary material.

4.2 Mean–Variance Factorization (MV-FAC)

MV-TD estimates the team-level mean and variance value functions, which characterize the expected search time and its variation. However, scalable multi-robot coordination requires further decomposing these quantities into individual utilities to support decentralized policy optimization. We next introduce the mean–variance value function factorization (MV-FAC) module.

MV-FAC employs two parallel factorization architectures to decompose team-level value functions into individual utilities. Specifically, a *mean factorization network* decomposes the team-level mean action-value function $\big(Q^{\pi}_{\text{tot}}(s,a)\big)$ into individual-level mean action-value functions $\big(Q^{\pi_i}_i(s_i,a_i)\big)$, while a *variance factorization network* decomposes the team-level variance action-value function $\big(\bar{Q}^{\pi}_{\text{tot}}(s,a)\big)$ into individual-level variance action-value functions $\big(\bar{Q}^{\pi_i}_i(s_i,a_i)\big)$. Each factorization network consists of agent-specific utility networks, and a mixing network whose parameters are generated by state-conditioned hypernetworks. The hyper-networks enforce non-negative mixing weights, guaranteeing the following monotonicity conditions:

$$\frac{\partial Q^{\pi}_{\text{tot}}(s,\boldsymbol{a})}{\partial Q^{\pi_i}_i(s_i,a_i)} \geq 0, \quad \frac{\partial \bar{Q}^{\pi}_{\text{tot}}(s,\boldsymbol{a})}{\partial \bar{Q}^{\pi_i}_i(s_i,a_i)} \geq 0, \quad \forall i \in \mathcal{I}. \tag{5}$$

Based on these monotonicity conditions, we establish the mean–std individual-global minimization (MS-IGM) theorem that ensures consistency between individual- and team-level mean-std objectives.

Theorem 1 (The MS-IGM Theorem)**.** *Suppose that the team-level mean and variance action-value functions admit monotonic factorizations with respect to corresponding individual utilities. Then, $\forall \zeta > 0$, the greedy minimization of the team-level mean–std objective is consistent with that of the individual-level objectives, i.e.,*

$$\begin{aligned}
&\arg\min_{\mathbf{a}}\Big(Q^{\pi}_{\text{tot}}(s,\mathbf{a}) + \zeta\sqrt{\bar{Q}^{\pi}_{\text{tot}}(s,\mathbf{a})}\Big) \\
&= \begin{pmatrix} \arg\min_{a_1} Q^{\pi_1}_1(s_1,a_1) + \zeta\sqrt{\bar{Q}^{\pi_1}_1(s_1,a_1)} \\ \vdots \\ \arg\min_{a_N} Q^{\pi_N}_N(s_N,a_N) + \zeta\sqrt{\bar{Q}^{\pi_N}_N(s_N,a_N)} \end{pmatrix}.
\end{aligned} \tag{6}$$

Proof.

$$\begin{aligned}
&\frac{\partial(Q_{\text{tot}} + \zeta\sqrt{\bar{Q}_{\text{tot}}})}{\partial(Q_i + \zeta\sqrt{\bar{Q}_i})} \\
=&\frac{\partial Q_{\text{tot}}}{\partial Q_i}\cdot\frac{\partial Q_i}{\partial(Q_i+\zeta\sqrt{\bar{Q}_i})} + \frac{\partial(\zeta\sqrt{\bar{Q}_{\text{tot}}})}{\partial \bar{Q}_i}\cdot\frac{\partial \bar{Q}_i}{\partial(Q_i+\zeta\sqrt{\bar{Q}_i})} \\
=&\frac{\partial Q_{\text{tot}}}{\partial Q_i}\cdot\frac{\partial Q_i}{\partial(Q_i+\zeta\sqrt{\bar{Q}_i})} + \frac{\zeta}{2\sqrt{\bar{Q}_{\text{tot}}}}\cdot\frac{\partial \bar{Q}_{\text{tot}}}{\partial \bar{Q}_i}\cdot\frac{\partial \bar{Q}_i}{\partial(Q_i+\zeta\sqrt{\bar{Q}_i})} \\
=&\frac{\partial Q_{\text{tot}}}{\partial Q_i}\cdot\frac{1}{1+\zeta\cdot\frac{\partial\sqrt{\bar{Q}_i}}{\partial Q_i}} + \frac{\zeta}{2\sqrt{\bar{Q}_{\text{tot}}}}\cdot\frac{\partial \bar{Q}_{\text{tot}}}{\partial \bar{Q}_i}\cdot\frac{1}{\zeta\cdot\frac{\partial\sqrt{\bar{Q}_i}}{\partial Q_i}} \\
=&\frac{\partial Q_{\text{tot}}}{\partial Q_i} + \frac{\sqrt{\bar{Q}_i}}{\sqrt{\bar{Q}_{\text{tot}}}}\cdot\frac{\partial \bar{Q}_{\text{tot}}}{\partial \bar{Q}_i} \geq 0.
\end{aligned}$$

□

Note that in the above proof process, we omit the superscripts π and π_i, and (s_i,a_i) for notational brevity, and make use of the independence between the mean and variance factorization processes, *i.e.*, $\partial Q_{\text{tot}}/\partial\bar{Q}_i = 0$, $\partial\bar{Q}_{\text{tot}}/\partial Q_i = 0$.

4.3 Decentralized Policy Optimization (Dec-PO)

The MS-IGM theorem established in Section 4.2 ensures that minimizing individual mean–standard deviation utilities is consistent with optimizing the team-level objective. In this subsection, we present the decentralized policy optimization (Dec-PO) module, which updates each robot's policy using only its own factorized mean–variance value functions.

Defining the individual-level mean–std objective as $J^{\pi_i}_i = Q^{\pi_i}_i + \zeta\sqrt{\bar{Q}^{\pi_i}_i}$, we derive its gradient with respect to the policy parameters $\boldsymbol{\theta}_i$, which is used for decentralized policy optimization.

$$\begin{aligned}
&\nabla_{\boldsymbol{\theta}_i} J^{\pi_i}_i(\boldsymbol{\theta}_i) \\
=&\nabla_{\boldsymbol{\theta}_i}(Q^{\pi_i}_i + \zeta\sqrt{\bar{Q}^{\pi_i}_i}) \\
=&\nabla_{\boldsymbol{\theta}_i} Q^{\pi_i}_i + \zeta\nabla_{\boldsymbol{\theta}_i}\sqrt{\bar{Q}^{\pi_i}_i} \\
=&\nabla_{\boldsymbol{\theta}_i} Q^{\pi_i}_i + \frac{\zeta}{2\sqrt{\bar{Q}^{\pi_i}_i}}\nabla_{\boldsymbol{\theta}_i}\bar{Q}^{\pi_i}_i \\
\overset{(*)}{\propto}&(\nabla_{\boldsymbol{\theta}_i}\log\pi_i)\times Q^{\pi_i}_i + \frac{\zeta}{2\sqrt{\bar{Q}^{\pi_i}_i}}(\nabla_{\boldsymbol{\theta}_i}\log\pi_i)\times\bar{Q}^{\pi_i}_i \\
=&\nabla_{\boldsymbol{\theta}_i}\log(\pi_i)\Big(Q^{\pi_i}_i + \frac{\zeta}{2}\sqrt{\bar{Q}^{\pi_i}_i}\Big),
\end{aligned} \tag{7}$$

where $(*)$ follows from the canonical policy gradient theorem whose proof is provided in [Sutton and Barto, 2018]. Note that the symbol '$\propto$' means *approximately proportional to*, and the final result in Eq. (7) is a stochastic approximation of the *true* gradient of $J^{\pi_i}_i(\boldsymbol{\theta}_i)$. The above derivation process omits (s_i,a_i) for notational brevity, *e.g.*, $Q^{\pi_i}_i$ abbreviates $Q^{\pi_i}_i(s_i,a_i)$, and $\log(\pi_i)$ abbreviates $\log\pi_i(s_i,a_i)$.

With the derived gradient of $J^{\pi_i}_i(\boldsymbol{\theta}_i)$ in Eq. (7), the Dec-PO module updates the individual robot's policy network parameter vector $\boldsymbol{\theta}_i$ as:

$$\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_i - \alpha_i\nabla_{\boldsymbol{\theta}_i} J^{\pi_i}_i(\boldsymbol{\theta}_i), \tag{8}$$

where $0 < \alpha_i < 1$ is the learning rate for the decentralized policy optimization process of Robot i.

4.4 Computational Complexity Analysis

In this subsection, we summarize the training procedure of the MV-FAC framework and analyze its computational complexity. Before presenting the pseudocode, we briefly introduce the notations used to describe the learnable components and optimization variables in MV-FAC. The parameter vectors of the individual mean and variance action-value networks, Q^{π}_i and $\bar{Q}^{\pi}_i$, are denoted by $\boldsymbol{w}_i$ and $\bar{\boldsymbol{w}}_i$, respectively. Robot i's decision-making policy π_i is parameterized by $\boldsymbol{\theta}_i$. The parameters of the mean and variance hyper-networks are denoted by $\boldsymbol{\phi}$ and $\bar{\boldsymbol{\phi}}$, respectively.

The pseudocode of MV-FAC's training procedure is presented in Algorithm 1, followed by the computational complexity analysis under the Big-$\mathcal{O}$ convention [Knuth, 1976].

Algorithm 1: MV-FAC's Training Procedure

Input: (1) Graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$; (2) Risk Coefficient ζ; (3) # of Robots N; (4) Max. # of Episodes: $E_{\max}$.
Output: Decentralized policies $\{\pi_i(\cdot \mid s_i; \boldsymbol{\theta}_i)\}_{i\in\mathcal{I}}$.

Initialize:
(1) Individual mean and variance action-value network parameters: $\{\boldsymbol{w}_i, \bar{\boldsymbol{w}}_i\}_{i\in\mathcal{I}}$;
(2) Individual policy network parameters $\{\boldsymbol{\theta}_i\}_{i\in\mathcal{I}}$;
(3) Hyper-network parameters $\boldsymbol{\phi}$ and $\bar{\boldsymbol{\phi}}$.
(4) Replay buffer $\mathcal{D} = \{\}$; counter $\leftarrow 0$;
while *counter* $\leq E_{\max}$ **do**
  Reset environment; $t \leftarrow 0$;
  $\forall i \in \mathcal{I}$: $s_i(t) \leftarrow \mathbf{0}$, $o_0^{(i)} \leftarrow 0$;
  while $\forall i \in \mathcal{I} : o_t^{(i)} = 0$ **do**
    for *each robot* $i \in \mathcal{I}$ **do**
      Observe $(p_t^{(i)}, o_t^{(i)})$;
      $s_t^{(i)} \leftarrow \mathrm{GRU}(s_{t-1}^{(i)}, p_t^{(i)}, o_t^{(i)})$;
      Sample $a_t^{(i)} \sim \pi_i(a|s_t^{(i)})$;
    Execute joint action $\boldsymbol{a}_t = \{a_t^{(i)}\}_{i\in\mathcal{I}}$;
    $r_t \leftarrow 1$;
    for *each robot* $i \in \mathcal{I}$ **do**
      Observe $(p_{t+1}^{(i)}, o_{t+1}^{(i)})$;
      $s_{t+1}^{(i)} \leftarrow \mathrm{GRU}\big(s_t^{(i)}, p_{t+1}^{(i)}, o_{t+1}^{(i)}\big)$;
    Store $\big\langle \{s_t^{(i)}\}, \{a_t^{(i)}\}, r_t, \{s_{t+1}^{(i)}\} \big\rangle$ into $\mathcal{D}$;
    $t \leftarrow t + 1$.
  // **MV-TD with MV-FAC**
  Sample a mini-batch of transitions from $\mathcal{D}$;
  for *each robot* $i \in \mathcal{I}$ **do**
    Compute $Q_i(s_i, a_i; \boldsymbol{w}_i)$ and $\bar{Q}_i(s_i, a_i; \bar{\boldsymbol{w}}_i)$;
  Mix individual utilities with hyper-networks to obtain $Q_{\mathrm{tot}}(s, a)$ and $\bar{Q}_{\mathrm{tot}}(s, a)$;
  Calculate mean and variance TD errors: δ and $\bar{\delta}$;
  Update $\{\boldsymbol{w}_i, \bar{\boldsymbol{w}}_i, \boldsymbol{\phi}, \bar{\boldsymbol{\phi}}\}$ to min. $\sum \delta^2$ and $\sum \bar{\delta}^2$.
  // **Dec-PO**
  for *each robot* $i \in \mathcal{I}$ **do**
    Update policy parameter $\boldsymbol{\theta}_i$ using Eq. (8) with $Q_i(s_i, a_i)$ and $\bar{Q}_i(s_i, a_i)$.
  counter $\leftarrow$ counter $+1$;

Let B be the mini-batch size, and denote C_Q, $C_{\bar{Q}}$, C_π, C_{mix} as the unit cost of one forward/backward pass for individual mean/variance network (Q_i and $\bar{Q}_i$), individual policy network (π_i), and the mixing network, respectively. Per episode, MV-TD evaluates N individual utilities, and applies two mixing networks, yielding $\mathcal{O}(BN)$ forward/backward passes, *i.e.*, $\mathcal{O}(BN(C_Q + C_{\bar{Q}}) + 2C_{\mathrm{mix}})$. For Dec-PO, each policy update costs $\mathcal{O}(BNC_\pi)$. Therefore, the overall complexity over $E_{\max}$ episodes is $\mathcal{O}(E_{\max}(BN(C_Q + C_{\bar{Q}} + C_\pi) + 2C_{\mathrm{mix}}))$. Note that C_Q, $C_{\bar{Q}}$, C_π and C_{mix} depend on the network configuration and the graph size, *i.e.*, $|\mathcal{V}|$ and $|\mathcal{E}|$. Detailed analysis is provided in the supplementary material.

5 Simulation Results and Analysis

This section evaluates the proposed MV-FAC framework on standard multi-robot search (MuRS) benchmarks by comparing its mean–std performance with state-of-the-art baselines. The benchmark algorithms, summarized in Table I, span three representative categories: prevailing MuRS methods, canonical MARL algorithms, and risk-sensitive decision-making approaches. All baselines are adapted to the MuRMSS setting with identical observation and action spaces to ensure fair comparison. We further conduct an ablation study to systematically isolate and quantify the contribution of each module in MV-FAC. All meta-parameter configurations and additional experimental results are provided in the supplementary material, and the source code is publicly available[1].

Table I: Summary of Selected Baseline Algorithms

Category	Methodology
MuRS Methods	PF-MAAC [Peng *et al.*, 2025b] HMA-SAR [Cao *et al.*, 2024] CE-PG [Guo *et al.*, 2023a] DRL-Searcher [Guo *et al.*, 2023b]
MARL Algorithms	P-MAT [Hu *et al.*, 2025] HAPPO [Kuba *et al.*, 2022] QMIX [Rashid *et al.*, 2020]
Risk-Sensitive Approaches	RiskQ [Shen *et al.*, 2023] RMIX [Qiu *et al.*, 2021] D-FAC [Sun *et al.*, 2021]

5.1 Benchmark Comparison

This subsection evaluates MV-FAC against state-of-the-art baselines in terms of mean–std search-time performance on two standard multi-robot search (MuRS) benchmarks, MUSEUM and OFFICE (Fig. 3). In both environments, robots start from fixed initial nodes, *i.e.*, Node 1 in MUSEUM and Node 43 in OFFICE, while the target is randomly initialized over the environment. At each time step, the target moves non-adversarially to one of its adjacent nodes with equal probability, and is considered 'detected' when any robot occupies the same node as the target.

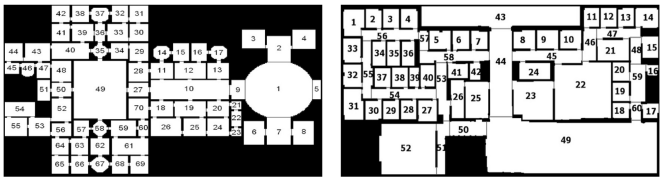


Figure 3: Standard MuRS benchmarks: MUSEUM and OFFICE.

Table II reports the mean–std performance comparison between MV-FAC and state-of-the-art baselines across different experimental configurations, including varying team sizes ($N \in \{3, 4, 5\}$) and risk-sensitivity coefficients ($\zeta \in \{0.1, 1, 10\}$). For each setting, results are obtained from 1000 independent simulation runs, with the sample mean and std to evaluate each algorithm's final mean–std performance.

[1] https://anonymous.4open.science/r/MV-FAC-3D4F.

Table II: Performance comparison between MV-FAC and baseline algorithms on standard multi-robot search benchmarks. **Bold** numbers indicate the best mean–std performance, and underlined numbers indicate the second best performance.

Env.	N	ζ	MV-FAC	QMIX	D-FAC	PF-MAAC	HMA-SAR	CE-PG	DRL-Searcher	P-MAT	HAPPO	RiskQ	RMIX
OFFICE	3	0.1	**5.18**	6.09	7.13	7.55	7.27	9.85	7.43	6.45	6.92	5.85	6.12
		1.0	**8.06**	11.11	14.46	12.80	10.33	17.20	14.97	10.85	12.15	10.45	10.92
		10.0	**36.86**	61.33	87.72	75.40	40.93	92.15	90.39	55.20	68.30	39.50	41.20
	4	0.1	**4.84**	5.81	6.85	7.15	6.90	8.90	6.85	6.10	6.55	5.25	5.60
		1.0	**7.16**	9.97	13.20	11.50	9.80	16.50	12.66	10.15	11.80	8.82	9.45
		10.0	**30.38**	51.55	78.62	65.10	38.50	88.40	75.72	48.30	60.25	35.15	37.80
	5	0.1	**4.68**	5.34	6.42	6.90	6.55	8.15	6.42	5.60	6.20	5.15	5.28
		1.0	**6.58**	9.19	11.92	10.50	9.10	14.20	11.13	9.45	11.10	8.25	8.65
		10.0	**25.57**	47.71	69.89	58.40	35.20	75.60	66.21	42.50	55.80	32.60	34.25
MUSEUM	3	0.1	6.55	8.07	14.74	10.50	**6.24**	12.80	7.39	7.10	9.20	6.57	6.95
		1.0	**9.44**	14.30	23.50	18.40	9.36	21.50	12.83	11.20	15.60	9.55	9.88
		10.0	**39.10**	76.58	111.07	95.20	40.59	105.30	67.19	58.40	82.10	38.15	40.95
	4	0.1	**5.95**	7.50	12.25	9.10	6.05	11.40	7.13	6.95	8.50	6.30	6.65
		1.0	**8.97**	12.18	19.05	16.80	9.10	18.20	12.30	10.90	13.50	9.25	9.60
		10.0	**37.97**	65.28	95.09	82.50	38.41	95.60	64.05	55.20	68.40	35.60	38.30
	5	0.1	**5.45**	6.83	10.25	7.90	5.80	9.80	7.06	6.75	7.50	6.10	6.45
		1.0	**8.39**	11.04	16.10	14.50	8.85	15.40	11.81	10.55	12.40	8.95	9.15
		10.0	**29.86**	58.15	80.62	72.20	36.69	72.10	59.33	52.10	62.50	33.86	35.80

In Table II, we observe that (1) MV-FAC consistently achieves the best or second-best mean–std performance across all configurations, demonstrating strong robustness to variations in both team size and risk sensitivity coefficient; (2) the performance gap between MV-FAC and baseline algorithms becomes increasingly pronounced as ζ grows, highlighting MV-FAC's ability to trade off expectation and variability. In contrast, canonical MuRS and MARL algorithms exhibit limited adaptability to high risk-sensitive scenarios. (3) While some risk-sensitive baselines, *e.g.*, RiskQ and RMIX, also improve performance for larger ζ values, their gains are generally inconsistent across different team sizes[2], whereas MV-FAC maintains stable and scalable performance as the team size grows.

5.2 Ablation Study

This subsection conducts an ablation study to isolate the contribution of each module in MV-FAC. Specifically, we compare four variants: (1) I-MV-TD (ϵ), where each robot independently learns its own mean–variance value functions via MV-TD and selects actions using the ϵ-greedy policy; (2) I-MV-TD (Dec), which replaces ϵ-greedy with the Dec-PO policy update process; (3) MV-TD+MV-FAC (ϵ), which learns team-level mean and variance values and applies MV-FAC to decompose them into individual utilities, but uses ϵ-greedy for action selection; (4) MV-TD+MV-FAC+Dec-PO, the full MV-FAC framework that couples MV-TD estimation, MV-FAC factorization, and Dec-PO optimization.

Table III presents the comparative mean–std performance of the four MV-FAC variants in the OFFICE environment. We observe that I-MV-TD(ϵ) shows limited improvement with increasing team size, as independent ϵ-greedy policies lead to redundant behaviors. I-MV-TD (Dec) slightly improves cooperation via stochastic policy updates, but lacks explicit inter-robot coordination. Introducing MV-FAC significantly enhances cooperation, while the ϵ-greedy policy still causes instability. Finally, MV-TD+MV-FAC+Dec-PO consistently achieves the best mean–std performance, combining effective coordination with stable policy optimization.

[2]RiskQ and RMIX need to factorize the return's *full* distribution into individual utilities, which tends to become unstable as the number of robots increases.

Table III: MV-FAC's Ablation Study.

Env.	N	ζ	I-MV-TD (ϵ)	I-MV-TD (Dec)	MV-TD + MV-FAC (ϵ)	MV-TD + MV-FAC + Dec-PO
OFFICE	3	0.1	11.24	9.56	6.12	**5.18**
		1.0	22.15	18.30	9.45	**8.06**
		10.0	135.40	98.20	48.12	**36.86**
	4	0.1	10.85	8.92	5.65	**4.84**
		1.0	24.60	19.45	8.80	**7.16**
		10.0	152.10	112.50	42.60	**30.38**
	5	0.1	10.50	8.55	5.40	**4.68**
		1.0	26.80	21.10	8.15	**6.58**
		10.0	168.30	125.40	35.80	**25.57**

6 Conclusion and Future Work

This paper formulates a risk-sensitive multi-robot search problem, termed MuRMSS, characterized by an inherently non-additive mean–standard deviation objective on search time. To address this challenge, we propose MV-FAC, an RL-based framework for MuRMSS, enabling joint mean–variance estimation, consistent factorization, and decentralized policy optimization. Extensive experiments on standard benchmarks demonstrate that MV-FAC consistently achieves superior mean–standard deviation performance over state-of-the-art baselines. In the future, we plan to incorporate multi-robot deadlock resolution mechanisms into MV-FAC under risk-sensitive objectives, and to explore integrating graph foundation models to improve the generalization ability of MV-FAC across diverse and previously unseen environments.

References

[Andreychuk *et al.*, 2025] Anton Andreychuk, Konstantin S. Yakovlev, Aleksandr Panov, and Alexey Skrynnik. Advancing learnable multi-agent pathfinding solvers with active fine-tuning. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10564–10571, 2025.

[Asfora *et al.*, 2020] Beatriz A. Asfora, Jacopo Banfi, and Mark E. Campbell. Mixed-integer linear programming models for multi-robot non-adversarial search. *IEEE Robotics and Automation Letters (RA-L)*, 5:6805–6812, 2020.

[Bone *et al.*, 2023] Sean Bone, Luca Bartolomei, Florian Kennel-Maushart, and Margarita Chli. Decentralised multi-robot exploration using Monte Carlo Tree Search. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7354–7361, 2023.

[Calzolari *et al.*, 2025] Gabriele Calzolari, Vidya Sumathy, Christoforos Kanellakis, and George Nikolakopoulos. Reinforcement learning driven multi-robot exploration via explicit communication and density-based frontier search. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11406–11412, 2025.

[Cao *et al.*, 2024] Xiao Cao, Mingyang Li, Yuting Tao, and Peng-Chen Lu. HMA-SAR: Multi-agent search and rescue for unknown located dynamic targets in completely unknown environments. *IEEE Robotics and Automation Letters (RA-L)*, 9:5567–5574, 2024.

[Chen *et al.*, 2024] Jiayu Chen, Chao Yu, Guosheng Li, Wenhao Tang, Shilong Ji, Xinyi Yang, Botian Xu, Huazhong Yang, and Yu Wang. Online planning for multi-uav pursuit-evasion in unknown environments using deep reinforcement learning. *IEEE Robotics and Automation Letters (RA-L)*, 10:8196–8203, 2024.

[Chen *et al.*, 2025] Yong Chen, Yu Shi, Xunhua Dai, Qing Meng, and Tao Yu. Pursuit-evasion game with online planning using deep reinforcement learning. *Applied Intelligence*, 55, 2025.

[Cui *et al.*, 2021] Jianfeng Cui, Dongchang Li, Peng Liu, Jia Qin, Yuedong Ma, and Zhigang Lu. Game-model prediction hybrid path planning algorithm for multiple mobile robots in pursuit evasion game. In *2021 IEEE International Conference on Unmanned Systems (ICUS)*, pages 925–930, 2021.

[Esfahani *et al.*, 2026] Messiah Abolfazli Esfahani, Ayşe Başar, and Sajad Saeedi. FG-PE: Factor-graph approach for multi-robot pursuit–evasion. *Robotics and Autonomous Systems (RAS)*, 195:105216, 2026.

[Fang *et al.*, 2022] Xu Fang, Chen Wang, Lihua Xie, and Jie Chen. Cooperative pursuit with multi-pursuer and one faster free-moving evader. *IEEE Transactions on Cybernetics (ToC)*, 52(3):1405–1414, 2022.

[Gao *et al.*, 2021] Yue Gao, Kry Yik Chau Lui, and Pablo Hernandez-Leal. Robust risk-sensitive reinforcement learning agents for trading markets. In *Reinforcement Learning for Real Life (RL4RealLife) Workshop at ICML*, 2021.

[Gonzalez and Jaillet, 2025] Victor Gonzalez and Patrick Jaillet. Multi-drone rescue search in a large network. *European Journal of Operational Research (EJOR)*, 324:787–798, 2025.

[Guo *et al.*, 2023a] Hongliang Guo, Zhaokai Liu, Rui Shi, Wei-Yun Yau, and Daniela Rus. Cross-entropy regularized policy gradient for multirobot nonadversarial moving target search. *IEEE Transactions on Robotics (T-RO)*, 39:2569–2584, 2023.

[Guo *et al.*, 2023b] Hongliang Guo, Qihang Peng, Zhiguang Cao, and Yaochu Jin. DRL-Searcher: A unified approach to multirobot efficient search for a moving target. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 35:3215–3228, 2023.

[Guo *et al.*, 2025] Hongliang Guo, Qi Kang, Wei-Yun Yau, Chee-Meng Chew, and Daniela Rus. R-FAC: Resilient value function factorization for multirobot efficient search with individual failure probabilities. *IEEE Transactions on Robotics (T-RO)*, 41:3385–3401, 2025.

[Hou *et al.*, 2024] Yukai Hou, Jin Zhao, Rongqing Zhang, Xiang Cheng, and Liuqing Yang. UAV swarm cooperative target search: A multi-agent reinforcement learning approach. *IEEE Transactions on Intelligent Vehicles (T-IV)*, 9:568–578, 2024.

[Hu *et al.*, 2025] Kun Hu, Muning Wen, Xihuai Wang, Shao Zhang, Yiwei Shi, Minne Li, Minglong Li, and Ying Wen. PMAT: Optimizing action generation order in multi-agent reinforcement learning. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 997–1005, 2025.

[Jiang *et al.*, 2025] Jiahui Jiang, Wang He, and Wenwu Yu. Risk-averse multi-agent reinforcement learning with distributional mean–variance formulation. In *International Conference on Artificial Intelligence and Machine Learning Research (CAIMLR)*, volume 13635. SPIE, 2025.

[Jo and Son, 2025] Yuseung Jo and Hyoung Il Son. CBS-HT: Prioritized safe interval path-planning algorithm for heterogeneous agricultural robot team. *IEEE Access*, 13:146630–146648, 2025.

[Knuth, 1976] Donald E. Knuth. Big Omicron and big Omega and big Theta. *SIGACT News*, 8(2):18–24, April 1976.

[Kolling and Kleiner, 2013] Andreas Kolling and Alexander Kleiner. Multi-UAV motion planning for guaranteed search. In *Proceedings of the 12th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, AAMAS '13, page 79–86, Richland, SC, 2013. International Foundation for Autonomous Agents and Multiagent Systems.

[Kuba *et al.*, 2022] JG Kuba, R Chen, M Wen, Y Wen, F Sun, J Wang, and Y Yang. Trust region policy optimisation in multi-agent reinforcement learning. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, pages 1046–1072, 2022.

[Kwa *et al.*, 2020] Hian Lee Kwa, Jabez Leong Kit, and Roland Bouffanais. Optimal swarm strategy for dynamic target search and tracking. In *Proceedings of the 19th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 672–680, 2020.

[Lee and Lee, 2024] Seung-Mok Lee and Jeong-Uk Lee. Multi-robot formation planning in maze-like environments consisting of narrow passages using graph search. *IEEE Access*, 12:167694–167704, 2024.

[Morin *et al.*, 2022] Michael Morin, Irène Abi-Zeid, and Claude-Guy Quimper. Ant colony optimization for path planning in search and rescue operations. *European Journal of Operational Research (EJOR)*, 305:53–63, 2022.

[Olsen *et al.*, 2022] Trevor Olsen, Nicholas M. Stiffler, and Jason M. O'Kane. Robust-by-design plans for multi-robot pursuit-evasion. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10716–10722, 2022.

[Patil *et al.*, 2023] Indraneel Patil, Rachel Zheng, Charvi Gupta, Jaekyun Song, Narendar Sriram, and Katia P. Sycara. Graph-based simultaneous coverage and exploration planning for fast multi-robot search. *ArXiv*, abs/2303.02259, 2023.

[Peng *et al.*, 2025a] Qihang Peng, Hongliang Guo, Boyang Li, Chih-Yung Wen, and Yaochu Jin. SMC-Searcher: Signal mediated coordination for decentralized multi-robot adversarial moving target search. *IEEE Transactions on Emerging Topics in Computational Intelligence (TETCI)*, 9(5):3399–3412, 2025.

[Peng *et al.*, 2025b] Qihang Peng, Hongliang Guo, Zhengyan Zhang, Chih-Yung Wen, and Yaochu Jin. PF-MAAC: A learning-based method for probabilistic optimization in time-constrained non-adversarial moving target search. *Swarm and Evolutionary Computation*, 92:101785, 2025.

[Qiu *et al.*, 2021] Wei Qiu, Xinrun Wang, Runsheng Yu, Rundong Wang, Xu He, Bo An, Svetlana Obraztsova, and Zinovi Rabinovich. RMIX: Learning risk-sensitive policies for cooperative reinforcement learning agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 23049–23062, 2021.

[Rahman *et al.*, 2022] A. Azizur Rahman, Arnab Bhattacharya, Thiagarajan Ramachandran, Sayak Mukherjee, Himanshu Sharma, Ted Fujimoto, and Samrat Chatterjee. AdverSAR: Adversarial search and rescue via multi-agent reinforcement learning. In *2022 IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–7, 2022.

[Rashid *et al.*, 2020] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research (JMLR)*, 21(1):7234–7284, 2020.

[Sharma *et al.*, 2025] Vikas Sharma, Rahul Gupta, and K.R. Sharma. Swarm intelligence in robotics: Optimizing PSO parameters for target search in single and multiple scenarios. In *International Conference on Mechatronics and Robotics Engineering (ICMRE)*, pages 140–144, 2025.

[Shen *et al.*, 2023] Siqi Shen, Chennan Ma, Chao Li, Weiquan Liu, Yongquan Fu, Songzhu Mei, Xinwang Liu, and Cheng Wang. RiskQ: risk-sensitive multi-agent reinforcement learning value factorization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 34791–34825, 2023.

[Sheng *et al.*, 2022] Wenda Sheng, Hongliang Guo, Wei-Yun Yau, and Yingjie Zhou. PD-FAC: Probability density factorized multi-agent distributional reinforcement learning for multi-robot reliable search. *IEEE Robotics and Automation Letters (RA-L)*, 7(4):8869–8876, 2022.

[Shree *et al.*, 2021] Vikram Shree, Beatriz Asfora, Rachel Zheng, Samantha Hong, Jacopo Banfi, and Mark Campbell. Exploiting natural language for efficient risk-aware multi-robot SaR planning. *IEEE Robotics and Automation Letters (RA-L)*, 6(2):3152–3159, 2021.

[Slumbers *et al.*, 2022] Oliver Slumbers, David Henry Mguni, Stefano B. Blumberg, Stephen Marcus McAleer, Yaodong Yang, and Jun Wang. A game-theoretic framework for managing risk in multi-agent systems. In *International Conference on Machine Learning (ICML)*, pages 32059–32087, 2022.

[Sun *et al.*, 2020] Yinjiang Sun, Rui Zhang, Wenbao Liang, and Cheng Xu. Multi-agent cooperative search based on reinforcement learning. In *2020 3rd International Conference on Unmanned Systems (ICUS)*, pages 891–896, 2020.

[Sun *et al.*, 2021] Wei-Fang Sun, Cheng-Kuang Lee, and Chun-Yi Lee. DFAC Framework: Factorizing the value function via quantile mixture for multi-agent distributional Q-learning. In *International Conference on Machine Learning (ICML)*, pages 9945–9954, 2021.

[Sutton and Barto, 2018] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[Tan *et al.*, 2021] Aaron Hao Tan, Federico Pizarro Bejarano, Yuhan Zhu, Richard Ren, and Goldie Nejat. Deep reinforcement learning for decentralized multi-robot exploration with macro actions. *IEEE Robotics and Automation Letters (RA-L)*, 8:272–279, 2021.

[Yuan *et al.*, 2024] Shilong Yuan, Zhou He, Pengyang He, and Wei Tang. An improved ant colony optimization algorithm for multi-robot path planning under complex tasks. In *International Conference on Intelligent Robotics and Control Engineering (IRCE)*, pages 95–99, 2024.

[Zhou *et al.*, 2024a] Meng Zhou, Xinheng Wang, Chang Wang, and Jing Wang. Multi-robot cooperative target search based on distributed reinforcement learning method in 3D dynamic environments. *Drones and Autonomous Vehicles*, 1:10012, 2024.

[Zhou *et al.*, 2024b] Yinglu Zhou, Yinya Li, Andong Sheng, Guoqing Qi, and Jinliang Cong. Optimal control strategies and target selection in multi-pursuer multi-evader differential games. *Neurocomputing*, 588:127701, 2024.