Hierarchical Multi-agent Reinforcement Learning in Spatial Domain Tasks using Empowerment Rewards

***Abstract*—In complex multi-agent tasks, various agents must cooperate to distribute relevant subtasks among each other in order to efficiently achieve the joint task objectives. Multi-agent Hierarchical Reinforcement Learning (MAHRL) provides an approach for learning to select the subtasks in response to the task environment states in a sequential decision manner. Standard MAHRL relies on a shared task reward to train various agents. This approach, however, has mostly been demonstrated on homogeneous agents or agents without inter-dependencies. When the joint task involves multiple agents of heterogeneous capabilities, only a few agents might reach the rewarding states in the task environment, while a certain subset of agents must play intermediate roles to enable former agents. The task reward becomes delayed or sparse for such intermediate agents which slows down learning without the use of intermediate guiding rewards. In this paper, we introduce a novel approach of MAHRL called Inter-Subtask Empowerment based Multi-agent Options (ISEMO) in which an Inter-Subtask Empowerment Reward (ISER) is given to an agent for enabling the execution of another agent's subtask. This effect of one agent enabling the subtask of another is named *empowerment* and used as a basis for deriving the intermediate guiding rewards. ISER is given in addition to the domain task reward in order to improve the inter-agent coordination. ISEMO also incorporates options model with parameterized *subtask termination* functions to learn the dynamic termination of various subtasks and relax the limitations posed by hand-crafted termination conditions. Experiments in a spatial Search and Rescue domain show that ISEMO can learn the inter-dependencies among agents during intermediate stages of a task which are distant from the main task's rewarding states, and perform better than the standard MAHRL technique called Cooperative HRL (CoHRL). It is also shown that the dynamic termination further improves the performance of ISEMO compared to CoHRL which uses non-optimizable fixed termination rules.**

***Keywords*—Multi-agent Coordination; Search & Rescue; Reinforcement Learning**

I. Introduction

Multi-agent Hierarchical Reinforcement Learning (MAHRL) [1], [2] provides a framework for the division of a complex joint task into simpler subtasks which can be distributed among different agents. Subtasks are basically temporally abstract actions in HRL, with a pre-defined plan or learned policy to achieve a sub-goal. The core problem in MAHRL is to learn an optimal high-level meta policy which selects a subtask for an agent in response to an environment state under the objective of the joint task coordination. In this paper, we consider a coordination problem which involves both homogeneous and heterogeneous agents with a set of pre-defined subtask plans and the associated preconditions. In a *heterogeneous* multi-agent system, an optimal decomposition can improve the *reachability* of the main task goal when a single agent is incapable of performing the entire task [3].

The primary challenge concerning MAHRL involving heterogeneous agents is to learn the inter-agent dependencies which affect the coordinated performance. In a complex heterogeneous system, it is likely that only a subset of agents reach the environment states which generate the task reward. Other subset of agents might simply take intermediate roles for creating the preconditions necessary to reach the rewarding states. A simple example is of sequential dependency where the subtask of an agent A creates preconditions necessary to execute the subtasks of another agent B, while only agent B can reach the rewarding states. The task reward, therefore, might be too delayed for the former agent to learn appropriate behaviour in a short learning period. The standard techniques in MAHRL [1], [4] rely only on shared task reward to learn coordination among homogeneous agents or among agents without complex inter-dependencies. For a heterogeneous system as discussed above, the task reward may be delayed or sparse for certain agents or agent-specific reward engineering may be required, both of which are undesirable.

A secondary issue concerns subtask commitment in a continuously changing environment. Subtask commitment refers to the time duration for which a subtask is continued after its selection/initiation. A subtask can only be initiated if its precondition is satisfied. In a multi-agent dynamic environment, the preconditions of the subtasks of an agent may change due to concurrent operation of another agent. Assume that an agent can choose between subtasks τ_1 and τ_2 but the preconditions of τ_2 are not satisfied initially. Hence, the agent chooses τ_1. During the execution of τ_1, however, another agent performs certain actions which enable the preconditions for τ_2. Now, whether the former agent continues with τ_2 or terminates τ_2 to make a subtask choice again itself becomes a decision problem. In the previous MAHRL work [5], a dynamic termination is performed by interrupting a subtask before reaching its (sub)goal state, but this termination decision is driven by hand-crafted events. The key challenge here is that an early termination exposes an agent to make stochastic choices from the entire subtask set and increases the exploratory behaviour, while a late termination restricts the choices and reduces the exploration. Highly exploratory behaviour slows down the learning process by delaying the completion of subtasks, while low exploration may lead to biased and sub-optimal policies. Therefore, hand-crafting the termination conditions demands explicit knowledge of the joint task environment and how it changes during the execution of the task.

To address these issues, we propose a new MAHRL approach named Inter-Subtask Empowerment based Multi-agent Options (ISEMO). Our main contributions are as follows,

- We introduce Inter-Subtask Empowerment Reward (ISER) as a guiding feedback for coordination when the joint task reward may be too delayed to train certain heterogeneous agents in a team. ISER is given as a positive reward to an agent that *empowers* another agent by enabling the preconditions of the latter agent's subtasks. Moreover, an agent can also receive ISER for performing a subtask which enables other subtask of its own. The intuition is that in the absence of an immediate task reward signal, an agent can use the criteria of expanding the space of possible subtasks as a motivation. This is loosely inspired by the concept of empowerment in RL [6] through which an agent rewards itself for the actions that lead to reachability of more future states. We explore this in the context of subtasks and preconditions, with an assumption that more subtask choices during the exploration phase lead to better reachability to the states with external task reward.
- We treat the termination decision as part of the learning problem to investigate its effect on the learning performance of the agents. To achieve this, we incorporate learned termination conditions into MAHRL by modelling the subtasks as *Options* with gradient-based optimization of the subtask termination functions. The optimization method is based on the single-agent option-critic algorithm [7]. Learned termination mitigates the restrictive need of the hand-crafted termination events or rules.

ISEMO is tested on a customized Search & Rescue (S&R) task in a two-dimensional spatial domain, involving multiple heterogeneous (as well as homogeneous) agents cooperating to rescue victims. It is compared against the standard MAHRL method called *Cooperative HRL* (CoHRL) [1], [5]. ISEMO learns the inter-dependencies among the subtasks of different agents and improves their credit assignment when the task reward is delayed or sparse. It results in better performance compared to CoHRL in terms of lower *victim death count*. The ISEMO method, the S&R setup, and the experiments are discussed in section 4, section 5, and section 6, respectively.

II. Preliminaries and Notations

In this section, we discuss the basic concepts and notations used in this paper.

A. Reinforcement Learning (RL)

RL is a learning mechanism for sequential decisions [8] modeled using a Markov Decision Process (MDP) consisting of an agent Ag with *primitive* (or single step) action set Λ and environment observation states S. The MDP has an underlying transition dynamics given by the probability distribution $P(s'|s,a) \rightarrow [0,1]$ where s' is the state observed after taking action a in a state s. The agent takes actions using an action policy $\pi : S \times \Lambda \rightarrow [0,1]$ and receives reward $r : S \times \Lambda \rightarrow \mathbf{R}$. The value of taking $a \in \Lambda$ in state $s \in S$ is $Q_\pi(s,a)$: $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t | s_0{=}s,\ a_0{=}a]$ where $\gamma \in [0,1)$ is a *discount factor*. Usually, $Q(s,a)$ is either a parametric or non-parametric function which predicts the expected sum of future rewards. In this paper, we use temporal difference (or TD) learning in which the Q function is updated as $Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a'))$ where s' is the next state after (s,a) and α is a learning rate. For parametric Q functions, the optimization is performed by minimizing the loss $\mathcal{L}(\theta) = (r + \gamma \max_{a'} Q_\theta(s',a') - Q_\theta(s,a))^2$.

B. Semi-Markov Decision Process (SMDP)

SMDP extends the standard MDP model used in RL to decisions over temporally abstract actions. A temporally abstract action, denoted in this paper as τ, differs from the primitive actions mainly in the sense that it is itself a plan or a policy over primitive actions. The transition probabilities in a SMDP are defined as $P(s'|s,\tau,n) \times P(n|s,\tau)$ where n is the number of time steps for which the temporally extended action τ lasts. The temporally extended action is formally defined as $\tau :< I_\tau, \pi_\tau, \beta_\tau >$ [9]. Here, π_τ is either a learned or pre-defined policy for τ, $\beta_\tau : S \rightarrow [0,1]$ is a probabilistic termination function for τ, and I_τ is the precondition for initiating τ. This is the basic setting of Hierarchical RL (HRL) in which a meta policy $\mu : S \times \Upsilon \rightarrow [0,1]$ learns to select a temporally abstract action $\tau \in \Upsilon$. In this paper, we refer to these temporally abstract actions as subtasks and learn the policy μ using *Options* framework [9], [7] discussed later. The preconditions and the policy of each subtask are pre-defined but the meta policy to select each subtask and the terminations of the subtasks are learned.

C. Multi-agent Hierarchical RL (MAHRL)

MAHRL [5] extends the basic Hierarchical RL mechanism discussed above to multi-agent coordination problems. Each agent Ag_i in a team of N agents has a corresponding set of subtasks Υ^i. The core problem is to learn a set of meta policies $\mu^1, \mu^2, ..., \mu^N$ for each agent such that coordination is achieved to satisfy the joint task. The meta policy of an individual agent Ag_i is defined as $\mu^i : S^i \times \Upsilon^i \rightarrow [0,1]$. A state observation $s^i \in S^i$ consists of global observation features ϕ^i from the perspective of the agent Ag_i and the subtask choices of other agents, collectively denoted as τ_{other}. The meta policy is guided by Q-value function $Q^i(\phi^i, \tau_{other}, \tau^i)$. Each agent seeks to maximize the total expected return $\mathbb{E}[R^i = \sum_{t=0}^{T} \gamma^t r_t^i]$ where the reward r^i may contain only the joint task reward or additional local rewards.

D. Options

The Option framework [9] is a general theoretical framework for SMDPs and HRL. A fundamental difference in the learning algorithm from the non-hierarchical RL discussed above is that the target to update the Q function is modified into:

$$y = r + \gamma((1-\beta_\tau(s'))Q_\theta(s',\tau) + \beta_\tau(s') \max_{\tau'} Q_\theta(s',\tau')) \quad (1)$$

where $\beta_\tau(s')$ is the probability of termination of τ in the next state. The loss function is $\mathcal{L}(\theta) = (y - Q_\theta(s,a))^2$. Therefore, if an option does not terminate, its own future value is used in the target to update its current value. Otherwise, the future value of the best subtask in the next state (i.e. the maximum value) is used. In this paper, the subtasks are technically *options* with defined policies and preconditions. Therefore, we use the terms subtask and option interchangeably.

III. Related Work

Makar et al. [5], and in a successor work Ghavamzadeh et al. [1], proposed a standard approach to MAHRL for cooperative agents with each agent using a pre-defined subtask hierarchy. Each agent learns its own Q-function which is conditioned on the global state information and the subtasks chosen by other agents. Inclusion of global information reduces the non-stationarity [10] in the Q-value estimates faced by the fully independent learning agents due to the effects of the behaviour of other agents [11], [12]. We use similar setting in our work, where each agent can observe the global environment and the agents communicate their subtask choices with each other. This MAHRL technique has previously been applied to Search & Rescue domain [13] but specifically for semi-autonomous control where human operator decisions are involved as elements in the subtask hierarchy. Macro-actions or subtasks also facilitate multi-agent coordination in decentralized partial observability settings [4]. In more recent work, a feudal multi-agent hierarchy is proposed [14] which functions with a centralized manager choosing joint subtasks which are then passed to corresponding agents. All of these methods, however, rely solely on a shared reward structure among agents which is suitable for homogeneous or non-dependent agents, but not for heterogeneous agents which may have complex inter-dependencies making the task reward delayed or sparse for certain agents. Inter-dependencies may be captured by incorporating coordination knowledge. Coordination information may be coded as *task pre-requisite* rules with winner-take-all subtask selection heuristics [15]. Coordination graphs [16], [17] is another popular approach used to define relations among agents. Intrinsic social motivation is used in [18], [19] to capture inter-agent influences, but with domain-specific intrinsic reward design. Use of domain-constrained knowledge allows reward engineering specific to different agents at the cost of generality. In contrast, the inter-subtask empowerment reward proposed in this paper applies generally to different agents without agent-specific reward engineering.

Apart from coordination, we also focus on a secondary issue of *termination conditions* of various subtasks. Subtasks can be terminated by comparing their Q-values to those of other subtasks, as formulated in Option interruption rules in [9]. This interruption mechanism is improved in [20]. In [21], the authors propose gradient-based update of a *termination function* within the Option framework. This adaptive termination has been integrated into an HRL framework in Option-Critic (OC) architecture [7]. These methods are applicable to single-agent HRL and so far we are not aware of the use or investigation of learned termination in MAHRL scenarios.

IV. Inter Subtask Empowerment Based Multi-agent Options (ISEMO)

In this section, we describe the Inter Subtask Empowerment Reward (ISER) which can be used as an internal (and intermediate) guiding signal for agents which do not directly reach the rewarding states in the environment but enable the subtasks for other agents. ISER can also be used by an agent for enabling its own subtasks.

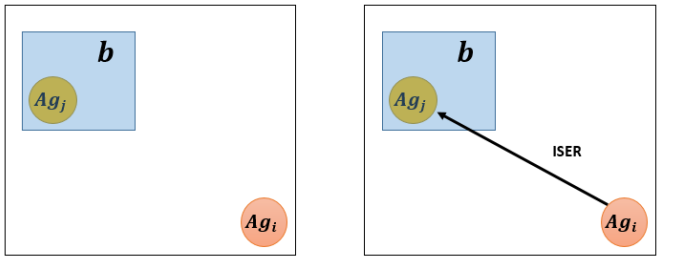
A. Subtask Preconditions

The calculation of ISER is based on the effect of one subtask on the preconditions of another subtask. We define the preconditions of a subtask using the state features. Suppose any state s_t consists of a D dimensional feature vector $[x_{1,t}, x_{2,t}, ..., x_{d,t}, ..., x_{D,t}]$. For a subtask τ, a precondition is defined corresponding to each feature as $pre(\tau, x_{d,t}, \hat{x}_{d,t})$. The precondition can be defined as $x_{d,t}$ being either greater than, lesser than, or equal to $\hat{x}_{d,t}$. The inequality or equality condition is specific to a $\tau, x_{d,t}$ pair. If $\hat{x}_{d,t}$ is *none*, then no precondition exists for τ corresponding to that feature. The preconditions are assumed to be already defined as part of each subtask and for each agent. The collective precondition for a particular subtask of agent Ag_i is defined as $PRE(\tau^i) : \{pre(\tau^i, x_{1,t}, \hat{x}_{1,t}), pre(\tau^i, x_{2,t}, \hat{x}_{2,t}), ..., pre(\tau^i, x_{D,t}, \hat{x}_{D,t})\}$. The subtask τ^i can be selected/initiated only if the preconditions corresponding to all the features are true. Otherwise, the subtask is disabled and the meta policy μ^i cannot select the subtask.

B. Inter Subtask Empowerment Reward (ISER)

ISER is used as a motivation signal for enabling any precondition $pre(\tau^i, x_{d,t}, \hat{x}_{d,t}) \in PRE(\tau^i)$. The intuition is that if one subtask enables a precondition of another subtask, the chance of the latter subtask being executed increases. We call this phenomenon *empowerment* in the context of the effect of one subtask on another. In a cooperative setting in which various subtasks contribute towards reaching the states where a task reward can be achieved, the empowerment of a subtask is a positive motivation as it might improve the reachability of the rewarding task states in the future. When a precondition $pre(\tau^i, x_{d,t}, \hat{x}_{d,t}) \in PRE(\tau^i)$ of an agent Ag_i is enabled, it communicates ISER to the agent Ag_j which causes this change in the precondition. In a spatial task such as Search & Rescue which is considered for the experimentation, the state features of the agent Ag_i are associated with different regions on the spatial environment map. Moreover, the effects of the actions of an agent are localized in the space. ISER, therefore, is directly communicated to the agent Ag_j which is in the immediate proximity of the region where the feature $x_{d,t}$ is modified at the time instance when $pre(\tau^i, x_{d,t}, \hat{x}_{d,t})$ is enabled. The agent Ag_j, to which ISER is communicated, allocates ISER to the state-subtask pair (s_t^j, τ_t^j). More than one preconditions in $PRE(\tau^i)$ may be satisfied simultaneously.

In such case, the agent Ag_i communicates only one ISER signal to each enabler agent (i.e., if one agent Ag_j enables multiple precondition of Ag_i at time t, it still receives single ISER signal). If Ag_j receives more than one ISER signals, the ISER is normalized to 1. ISER is incorporated into the reward function by weighted summation as $R^i = w_1 \times R^i_G + w_2 \times ISER$, where R^i_G is the main task reward observed by the agent Ag_i. In a case of multiple agents being proximal to the region where a feature change occurs, a false reward assignment might happen. The probability of the repetition of such assignment, however, is low if we consider that the same situation may not appear frequently over the multiple episodes of training used in RL/HRL. The ISER concept described above is depicted concisely in Figure 1.



Agent Ag_j modifies certain environment feature in a region b

Agent Ag_i detects a precondition enabled due to change in the environment feature in region b and communicates ISER to the proximal agent Ag_j

Fig. 1: The basic concept of ISER generation and sharing.

The ISER generation and communication described above works under the following assumptions: (i) that the agent Ag_i can observe the global spatial environment to detect changes in its features and corresponding preconditions, (ii) the agents can communicate with each other, and (iii) the effect of the actions of an agent are spatially localized. Under the second assumption of communication, agents can share selective local observation (including their own positions) with each other. When an agent receives local observations from other agents, it consolidates these observations into the global features $[x_{1,t}, x_{2,t}, ..., x_{d,t}, ..., x_{D,t}]$ discussed above which represent the joint situation. The assumption (iii) stated above may not hold true for all kinds of tasks. For such tasks, the determination of which subtask or agent enables another subtask or agent has to be based on some form of counterfactual reasoning [22], [23]. Such reasoning, however, might require complex simulation models to evaluate multiple counterfactual choices in parallel for a large number of agents. We leave this investigation for the future work.

While the primary motivation behind ISER is to enable credit assignment across different agents, it is also assigned to a subtask performed by an agent Ag_i which enables the precondition of another subtask of Ag_i itself. In this case, communication of ISER is not necessary as it is local to the agent Ag_i.

C. Multi-agent Options with learned terminations

The ISER signal discussed above can be incorporated into the standard MAHRL methods [5], [1] which are originally based on the MAXQ framework [24] but can be easily generalized to other HRL techniques such as Options [9], Feudal learning [25], [26] etc. In this work, we choose the Options framework as the basis for ISEMO. This choice is mainly motivated by the provision of the gradient based optimization of the subtask termination functions in the Option-Critic (OC) framework [7] which is originally proposed for single agent domains. In the OC framework, the termination probability of a subtask τ in a state s is represented as $\beta_{\tau,\nu}(s)$. This is a parameterized sigmoid function, with parameters ν which are updated using the gradient rule: $\nu \leftarrow \nu - \alpha_\nu \frac{\partial \beta_{\tau,\nu}(s)}{\partial \nu}(Q(s,\tau) - V(s) + \eta)$, where α_ν is the learning rate. Basically, the gradient is scaled by the advantage $(Q(s,\tau) - V(s))$ of performing τ in state s over the value of the state $V(s)$ which can be taken as the maximum $max_{\hat{\tau}} Q(s,\hat{\tau})$ over all possible choices. Therefore, if the advantage is positive, ν is shifted in the direction which reduces the termination probability β (and vice versa). Here, η is the *switching penalty* to prevent frequent termination of *options* when the Q values are near zero in the initial phases of learning [7], [27].

To adapt the OC framework to our multi-agent approach, the Q-function of each agent Ag_i is represented as $Q^i_\theta(\phi^i, \tau_{other}, \tau^i)$ where ϕ_i is environment feature observation and τ_{other} is a vector containing the subtask choices of other agents to inform the agent about the 'intentions' of other agents (similar to [1]). θ are the parameters of Q^i. Similarly, the termination functions take into account the global information containing τ_{other} and are represented as $\beta^i_{\tau^i,\nu}(\phi^i, \tau_{other})$, thereby terminations can also occur in response to the subtask choices of other agents. Both Q and β functions are linear approximators trained via Stochastic Gradient Descent (SGD). The β functions are trained using the gradient rule discussed above, while the Q functions are trained by the loss minimization as discussed under subsection II(D). The reward function of each agent is $R^i = w_1 \times R^i_G + w_2 \times ISER$, where R^i_G is the task reward observed by the agent Ag_i, which may be same across different agents.

V. Search and Rescue (S&R) Task

The setup of the Search & Rescue task is as follows: The environment is represented as a two-dimensional situation map $M : \{m_{pq} | p = 1, 2, ..., P; q = 1, 2, ..., Q\}$ (Figure 2). Here, m_{pq} is the attribute value of a map cell at location (p,q) which takes one of the following scalar values: unknown cell; victim found, critical (default); victim stable, not relocated; Base Station (contains Life Supply); vacant cell; debris (covering victim); path blockage; wall; if an agent is at cell (p, q), then the *id* of the agent. The initial global state is $\forall(p,q), (m_{pq} = unknown)$. There are V randomly scattered but concealed victims. Each victim $v \in \{1, 2, ..., V\}$ has a *health value* H_v, and location $vloc_v$. The Base Station location is BS. H^{BS}_v denotes the health of the victim v when the victim is relocated to the Base Station. Until a victim is relocated, $H^{BS}_v = -\infty$.

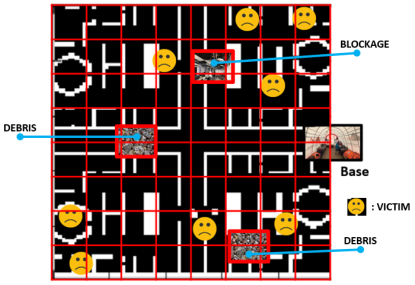


Fig. 2: Map of indoor environment. White cells are obstacles, black cells are *vacant*. Two victims are underneath *debris*. Red grids represent regional blocks over which the main task can be *parallel* decomposed.

At the start of S&R, a hypothetical health value H_0 = +1 is set, which decays over time at a rate of ψ_1. The *key events* in the task are as follows:

Discovery: victim v is discovered, $H_v = H_0$, m_{vloc_v} = critical victim, future decay rate = ψ_1. **Delivery of Life Supply to** v: m_{vloc_v} = stable victim, $H_v \leftarrow H_v$ + 1, future decay rate = ψ_2. **Carry victim** v **for relocation**: $H_v \leftarrow H_v$ + 1, future decay rate = ψ_3. **Relocation done**: $vloc_v = BS$, $H_v^{BS} = H_v$, future decay rate = 0. **A victim dies if** $H_v \leq \mathbf{0}$. If so, $H_v \leftarrow H_v$ + 1 is impossible.

A. Feature Space

The map M is divided into $k \times k$ (e.g. k = 10) blocks. Each block b approximates a small region of the map (Figure 2) and the features in that region. An agent observes following global and local/internal features:

$[x_1].b$ = fraction of block area occupied by unknown cells,
$[x_2].b$ = count of critical victims in the block,
$[x_3].b$ = average health of critical victims in the block,
$[x_4].b$ = minimum health among critical victims in the block,
$[x_5].b$ = count of stable victims in the block,
$[x_6].b$ = average health of stable victims in the block,
$[x_7].b$ = minimum health among stable victims in the block,
$[x_8].b$ = is the agent (self) at the search frontier in the block? *'internal'*.
$[x_9].b$ = is debris present in the block?,
$[x_{10}].b$ = is blockage present in the block?,
x_{11} = self location,
x_{12} = quantity of carried Life Supplies *'internal'*,
x_{13} = number of carried victims *'internal'*

The features marked *internal* are not communicated with other agents. Other features are shared and collectively grouped into a vector $\phi^i = \{[x_1, ..., x_{10}]._1, [x_1, ..., x_{10}]._2, ..., [x_1, ..., x_{10}]._{k \times k}, [x_{11}, x_{12}, x_{13}]\}$ for agent Ag_i. Finally, a vector τ^i_{other} containing other agents' locations and subtasks is added. Concisely, the state of an agent Ag_i is $s^i = \{\phi^i, \tau^i_{other}\}$.

B. Agents and Subtasks

The subtasks available to different types of agents are shown in Figure **??**. In addition to the shown subtasks, each agent can also choose a *NONE* subtask without any precondition. The agent *types* are A_1: Search, A_2: Aid, A_3: Relocate, and A_4: Helper. There can be multiple agents of the same type (*homogeneous*). But more importantly, there is heterogeneity in the nature of agents across different types, with certain agents playing intermediate roles in helping other agents reach the main task goals. The goals of the main task are expressed in the form of task rewards. The agents of Search type and the Helper agent share a terminal reward $R_G = \frac{1}{T_{search}} \times \mathbb{1}_{area\,searched=totalarea}$. Here, T_{search} is total time taken to complete search, and the second term is zero if the search is not complete by an episode termination. The agents of type Aid, Relocate, and again, Helper share a task reward $R_G = 1$ upon the relocation of a victim to the Base Station if the victim v is alive ($H_v > 0$) and $R_G = -20$ if the victim is dead. If multiple victims are relocated simultaneously, then the rewards are summed. Both types of rewards are scaled to 1. The evaluation of the task is based on the count of dead victims. A rapid search combined with timely relocation of all victims should result in minimum deaths. The dependencies or enabling relations among various subtasks are also depicted in Figure **??**. A subtask is eligible to receive ISER if it effects the enabling of certain precondition of another subtask by modifying certain feature value.

VI. Experiments and Results

For the experiments, the task configuration is: H_0 = 1; $\psi_1 = \frac{1}{1500}$, $\psi_2 = \frac{1}{2} \times \psi_1$, and $\psi_3 = \frac{1}{10} \times \psi_1$. These values are set by trials to ensure that the main task is neither too simple nor unsolvable within 500 training episodes. There are ten victims to be discovered and rescued, with two victims under debris. The victims locations are randomly set. We use four agents of types A_1, A_2, and A_3 each and one of type A_4 (totally 13 agents). The agent types are shown in Figure **??**. The map dimension is 100×100 and it is divided into 10×10 blocks (i.e., k=10). The reward function of each agent Ag_i is $R^i(s^i, \tau^i) = 1000 \times R^i_G(s^i, \tau^i) + 1 \times ISER(s^i, \tau^i)$. The discount factor is γ = 0.99. Each experiment is performed for five runs of 500 episodes each, with 25000 decision steps (or time steps) per episode. The performance measure for all the experiments is the *victim death count*. ISEMO is compared against the *Cooperative HRL* method [1] which is a standard MAHRL approach. We refer to this method as **CoHRL**. Since ISEMO differs from CoHRL in terms of both ISER and the learned termination functions, we also implement a version of CoHRL in which ISER is added (**CoHRL+ISER**) for focused comparisons. The results are shown in Figure 3. It can be observed that ISER alone significantly improves the performance of CoHRL. ISEMO achieves the lowest death count by combining ISER with dynamic subtask termination. For CoHRL, the termination boundary of each subtask is 150 time steps after the start of the subtask. Discussion on the effect of the termination boundaries is provided in the subsection VI(C).

Search (A_1)	SCAN	Scan over the line-of-sight with radial scan-line of length = 10 cells	PRE: $[x_8]_{\cdot b} = 1$ (i.e., the agent is at the frontier)
Search (A_1)	Move To Frontier (MTF)	Move to the nearest unknown cell within *block b*	PRE: $[x_1]_{\cdot b} \neq 0$, $[x_8]_{\cdot b} = 0$ (i.e. there is unknown frontier in the block and the agent is not at the block frontier)
Search (A_1)	SCAN	Scan over the line-of-sight with radial scan-line of length = 10 cells	PRE: $[x_8]_{\cdot b} = 1$ (i.e., the agent is at the frontier)
Search (A_1)	Move To Frontier (MTF)	Move to the nearest unknown cell within *block b*	PRE: $[x_1]_{\cdot b} \neq 0$, $[x_8]_{\cdot b} = 0$ (i.e. there is unknown frontier in the block and the agent is not at the block frontier)
Search (A_1)	SCAN	Scan over the line-of-sight with radial scan-line of length = 10 cells	PRE: $[x_8]_{\cdot b} = 1$ (i.e., the agent is at the frontier)
Search (A_1)	Move To Frontier (MTF)	Move to the nearest unknown cell within *block b*	PRE: $[x_1]_{\cdot b} \neq 0$, $[x_8]_{\cdot b} = 0$ (i.e. there is unknown frontier in the block and the agent is not at the block frontier)
Search (A_1)	SCAN	Scan over the line-of-sight with radial scan-line of length = 10 cells	PRE: $[x_8]_{\cdot b} = 1$ (i.e., the agent is at the frontier)
Search (A_1)	Move To Frontier (MTF)	Move to the nearest unknown cell within *block b*	PRE: $[x_1]_{\cdot b} \neq 0$, $[x_8]_{\cdot b} = 0$ (i.e. there is unknown frontier in the block and the agent is not at the block frontier)

TABLE I: Blabla

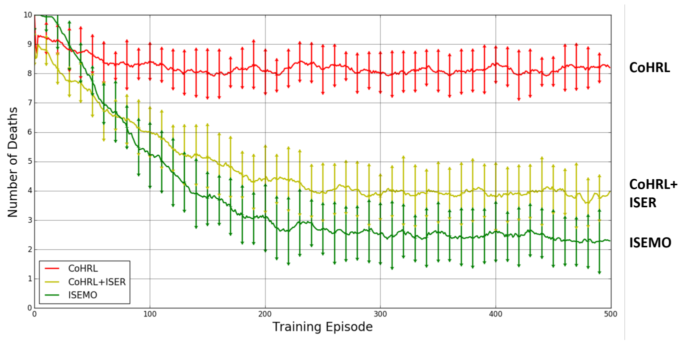


Fig. 3: ISEMO is compared against a standard MAHRL technique named CoHRL here. We observe improvement by just adding ISER to CoHRL itself. However, ISEMO with both ISER and learned terminations provides the best performance. The results are shown for five runs with averaging over 10 episode window.

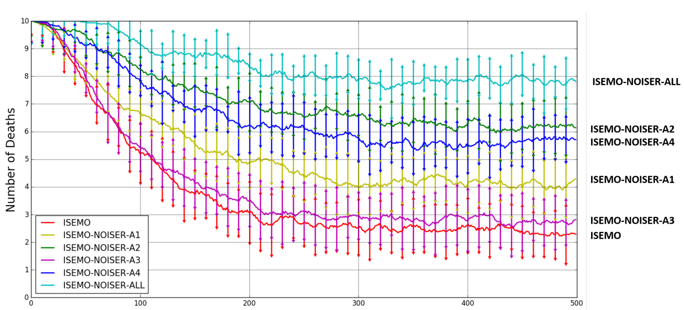


Fig. 4: This plot shows the effect of ISER on individual agents. The worst performance bound is set by removing ISER for all agents and the best bound is by original ISEMO. The results are shown for five runs with averaging over 10 episode window.

A. Ablation experiments 1

Further ablation experiments are performed to determine which agents are the most affected by ISER. For these experiments, ISER is disabled for one *type* of agents while all other types receive ISER. Suppose ISER is disabled for the agents of type A_1, then the method variant is named **ISEMO-NOISER-A1**. Similarly, this applies for all the other types of agents. It is observed that the effect of ISER as an intermediate reward is not significant for the agents of type A_3 (Relocate). This is possibly because the *Relocate* type agents immediately receive the task reward R_G upon the relocation of one or more victims. Therefore, even in the absence of ISER, these agents receive correct reward for the RELOCATE subtask, while the value can simply propagate to the CARRY subtask via Q-updates. For the other types of agents, the effect of ISER is more significant. The *Search* agents (type A_1) do receive a task reward for scanning (as discussed under subsection V(B)) but only terminally, when the entire map has been scanned. Therefore, ISER received during the intermediate stages (Figure **??**) still helps the scanning process by learning to avoid the NONE action. The *Aid* agents (type A_2) make indirect contribution to the task reward generated upon the relocation of the victims by enabling the subtasks of the agents of type A_3 (*Relocate*). Similarly, the *Helper* agent A_4 takes intermediate role by enabling other agents to perform subtasks which ultimately lead to a task reward. The task reward, therefore, is delayed or even sparse from the perspective of the *Aid* and the *Helper* agents. ISER captures the inter-subtask enabling roles of such agents and provides an intermediate guiding signal to accelerate learning. This is observed in Figure 4 as the agents of type A_3 are the least affected by

ISER, with the performances of ISEMO-NOISER-A3 being close to ISEMO. In contrast, more significant performance deterioration is seen by the removal of ISER for other types of agents.

B. Ablation experiments 2

We analyze the behavior of the *Helper* agent (A_4) by running tests on one of the training maps. In the first test set (Figure 5), three different scenarios are shown in which a search agent is close to the *blockage* and a helper agent A_4 is situated at three different locations. The probabilities of the selection of different subtasks (calculated as the softmax of $Q(s,\tau)$) are also shown. The desired subtask is CLEAR BLOCKAGE, because it enables A_1 to perform SCAN. It is observed that in all the three scenarios, the probability distribution of the Helper agent trained using CoHRL is more or less uniform over CLEAR BLOCKAGE and NONE (other subtasks are masked as their preconditions are not satisfied in this scenario). This is expected since the Helper agent gets no reward for CLEAR BLOCKAGE during training as the task reward is only terminally received. On the other hand, using ISER as a motivation to enable the Search agent, the Helper agent is able to figure out the correct choice when in proximity to the blockage. This is observed in the case of ISEMO.

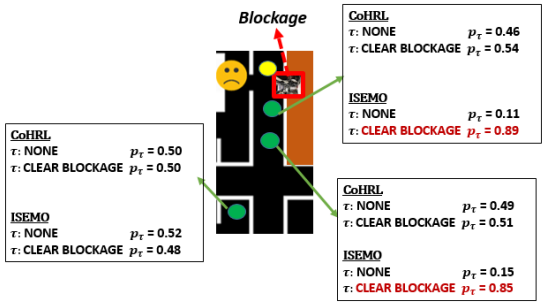


Fig. 5: Comparing the behavior of the Helper agent (Green) during test runs in three different scenarios when a Search agent (Yellow) is near a blockage. There is only one Helper agent but different scenarios are shown in the same figure to save space. Refer subsection VI(B) for details.

In the second test set (Figure 6), one *Helper*, one *Search*, and one *Aid* agents are shown. Three scenarios are considered in which *Debris* is discovered while the Aid agent is far. The desired subtask is CLEAR DEBRIS to reveal a victim which enables a subtask for the Aid agent. Once again, the Helper agent trained with CoHRL fails to allocate higher probability to the CLEAR DEBRIS subtask because the only relevant task reward during training for this choice is that received upon the relocation of the victim (hidden under the debris), an event which is much delayed in the future because first the Helper agent must clear debris, then the Aid agent delivers Life Supply, followed by a Relocate agent performing relocation which generates the task reward. Using ISER as the motivation to enable the Aid agent, the Helper agent is able to figure out the correct choice when neares to the debris than the blockage, as observed in the case of ISEMO.

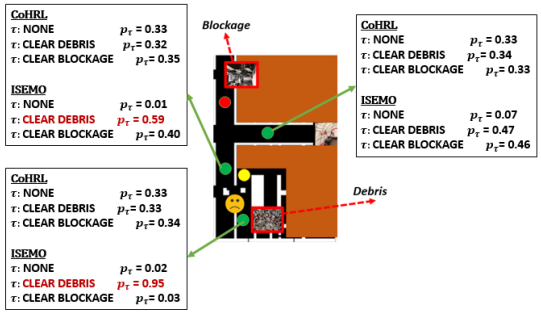


Fig. 6: Comparing the behavior of Helper agent (Green) during test runs in three different scenarios when a Search agent (Yellow) and Aid agent (Red) are present and Debris is found. There is only one Helper agent but different scenarios are shown in the same figure to save space. Refer subsection VI(B) for details.

C. Effect of Termination Condition

Figure 7 depicts the performance of ISEMO against CoHRL+ISER with different termination boundaries. For CoHRL+ISER itself the performance varies with the subtask execution duration TM. For small duration of TM = 1 to 10, the agents fail to reach rewarding states. For higher TM (around 150), the agents remain *committed* to the chosen subtasks long enough to observe the rewards, and hence, the performance is better. However, further increasing TM degrades the performance due to the over-commitment to the chosen subtasks (reduced exploration). The best performance of CoHRL+ISER is for TM = 150. It is impractical to try all the possible TM values exhaustively or to train a policy to choose TM considering the output space complexity of such a policy. ISEMO, in contrast, learns a continuous probability function $\beta_\tau(s)$ such that explicit duration need not be fixed. For switching penalty η = 1, ISEMO performs worse than CoHRL+ISER initially when β functions are being learned. However, it converges to the lowest death count as training proceeds. For this experiment, $\beta_{\tau^i}(s^i)$ is initialized to 0.3 $\forall Ag_i, \forall \tau^i, \forall s^i$. A lower probability value ensures that ISEMO executes subtasks for longer duration during initial training phases so that sufficient rewards are visible.

VII. Conclusion

In this paper, we propose a Multi-agent HRL (MAHRL) method for complex spatial domain tasks involving heterogeneous agents which might take intermediate roles by enabling other agents to reach the rewarding states of the task and themselves observe the task reward as delayed or sparse. The reliance on a shared task reward, such as in the existing MAHRL methods [5], [1], [4] may not be enough for fast learning is such domains. Our method named Inter-Subtask Empowerment based Multi-agent Options (ISEMO) consists of an Inter-Subtask Empowerment Reward (ISER) which captures the inter-dependencies among heterogeneous agents. ISER is provided in addition to the global task reward based on the criteria of empowering other agents by enabling the preconditions of others' subtasks. ISER is also allocated to a

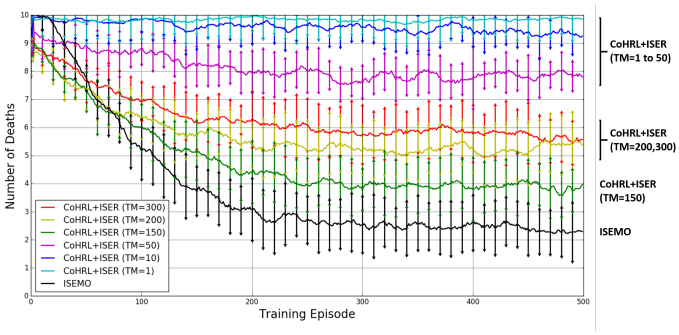


Fig. 7: Performance of baseline CoHRL for different termination boundaries v/s ISEMO with learned termination. For CoHRL, TM is the number of time steps for which a subtask is held for execution after starting. The results are shown for five runs with averaging over 10 episode window.

subtask which enables the preconditions of another subtask of the same agent. The criteria of empowerment provides a general basis for allocating intermediate rewards without the agent-specific reward engineering. Furthermore, ISEMO extends the *learned subtask termination* concept of single-agent Option-Critic [7] to multi-agent setting using *options*, and thereby, relaxes the limitation of the fixed pre-defined termination boundaries. Experiments in a spatial Search & Rescue domain show that agents trained by including ISER perform better than a standard Cooperative HRL (CoHRL) [1] technique as ISER facilitates better credit assignment to agents than the delayed task reward. We also observe further improvement in the performance with ISEMO using the learned subtask-termination conditions. In the future, we plan to develop ISEMO along the following directions: (i) investigate decentralized approaches with limited communication in which case the ISER can only be communicated sparsely, (ii) explore counterfactual reasoning to relax the assumption of spatially localized effects of the agents' actions used to determine which agents should receive ISER, and (iii) explore more end-to-end approaches in which the subtask policies and preconditions are also learned.

REFERENCES

[1] M. Ghavamzadeh, S. Mahadevan, and R. Makar, "Hierarchical multi-agent reinforcement learning," *Autonomous Agents and Multi-Agent Systems*, vol. 13, no. 2, pp. 197–229, 2006.

[2] A. G. Barto and S. Mahadevan, "Recent advances in hierarchical reinforcement learning," *Discrete event dynamic systems*, vol. 13, no. 1-2, pp. 41–77, 2003.

[3] C. Boutilier, T. Dean, and S. Hanks, "Decision-theoretic planning: Structural assumptions and computational leverage," *Journal of Artificial Intelligence Research*, vol. 11, pp. 1–94, 1999.

[4] M. Liu, C. Amato, E. P. Anesta, J. D. Griffith, and J. P. How, "Learning for decentralized control of multiagent systems in large, partially-observable stochastic environments." in *AAAI*, 2016, pp. 2523–2529.

[5] R. Makar, S. Mahadevan, and M. Ghavamzadeh, "Hierarchical multi-agent reinforcement learning," in *Proceedings of the fifth international conference on Autonomous agents*. ACM, 2001, pp. 246–253.

[6] A. S. Klyubin, D. Polani, and C. L. Nehaniv, "All else being equal be empowered," in *European Conference on Artificial Life*. Springer, 2005, pp. 744–753.

[7] P.-L. Bacon, J. Harb, and D. Precup, "The option-critic architecture." in *AAAI*, 2017, pp. pp.1726–1734.

[8] R. S. Sutton, A. G. Barto *et al.*, *Reinforcement learning: An introduction*, 1998.

[9] R. S. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial intelligence*, vol. 112, no. 1-2, pp. 181–211, 1999.

[10] L. Matignon, G. J. Laurent, and N. Le Fort-Piat, "Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems," *The Knowledge Engineering Review*, vol. 27, no. 1, pp. 1–31, 2012.

[11] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls *et al.*, "Value-decomposition networks for cooperative multi-agent learning," *arXiv preprint arXiv:1706.05296*, 2017.

[12] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in Neural Information Processing Systems*, 2017, pp. 6379–6390.

[13] Y. Liu, G. Nejat, and J. Vilela, "Learning to cooperate together: A semi-autonomous control architecture for multi-robot teams in urban search and rescue," in *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 1–6.

[14] S. Ahilan and P. Dayan, "Feudal multi-agent hierarchies for cooperative reinforcement learning," in *Workshop on Structure & Priors in Reinforcement Learning (SPiRL 2019) at ICLR 2019*, 2019, pp. 1–11.

[15] T.-H. Teng, A.-H. Tan, J. A. Starzyk, Y.-S. Tan, and L.-N. Teow, "Integrating self-organizing neural network and motivated learning for coordinated multi-agent reinforcement learning in multi-stage stochastic game," in *Neural Networks (IJCNN), 2014 International Joint Conference on*. IEEE, 2014, pp. 4229–4236.

[16] C. Guestrin, M. Lagoudakis, and R. Parr, "Coordinated reinforcement learning." Citeseer.

[17] J. R. Kok and N. Vlassis, "Collaborative multiagent reinforcement learning by payoff propagation," *Journal of Machine Learning Research*, vol. 7, no. Sep, pp. 1789–1828, 2006.

[18] P. Sequeira, F. S. Melo, R. Prada, and A. Paiva, "Emerging social awareness: Exploring intrinsic motivation in multiagent learning," in *Development and Learning (ICDL), 2011 IEEE International Conference on*, vol. 2. IEEE, 2011, pp. 1–6.

[19] E. Hughes, J. Z. Leibo, M. G. Philips, K. Tuyls, E. A. Duéñez-Guzmán, A. García Castañeda, I. Dunning, T. Zhu, K. R. McKee, R. Koster *et al.*, "Inequity aversion resolves intertemporal social dilemmas," *arXiv preprint arXiv:1803.08884*, 2018.

[20] D. J. Mankowitz, T. A. Mann, and S. Mannor, "Time regularized interrupting options," in *Internation Conference on Machine Learning*, 2014.

[21] G. Comanici and D. Precup, "Optimal policy switching algorithms for reinforcement learning," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 2010, pp. 709–714.

[22] D. H. Wolpert and K. Tumer, "Optimal payoff functions for members of collectives," in *Modeling complexity in economic and social systems*. World Scientific, 2002, pp. 355–369.

[23] J. N. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[24] T. G. Dietterich, "Hierarchical reinforcement learning with the maxq value function decomposition," *Journal of Artificial Intelligence Research*, vol. 13, pp. 227–303, 2000.

[25] P. Dayan and G. E. Hinton, "Feudal reinforcement learning," in *Advances in neural information processing systems*, 1993, pp. 271–278.

[26] A. S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, and K. Kavukcuoglu, "Feudal networks for hierarchical reinforcement learning," *arXiv preprint arXiv:1703.01161*, 2017.

[27] J. Harb, P.-L. Bacon, M. Klissarov, and D. Precup, "When waiting is not an option: Learning options with a deliberation cost," *arXiv preprint arXiv:1709.04571*, 2017.