A Collaborative Multiagent Framework based on Online Risk-Aware Planning and Decision-Making

A Collaborative Multiagent Framework based on Online Risk-Aware Planning and Decision-Making

Iván Palomares, Ronan Killough, Kim Bauters, Weiru Liu, Jun Hong
School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast (United Kingdom)
{i.palomares,rkillough01,k.bauters,w.liu,j.hong}@qub.ac.uk

***Abstract*—Planning is an essential process in teams of multiple agents pursuing a common goal. When the effects of actions undertaken by agents are uncertain, evaluating the potential risk of such actions alongside their utility might lead to more rational decisions upon planning. This challenge has been recently tackled for single agent settings, yet domains with multiple agents that present diverse viewpoints towards risk still necessitate comprehensive decision making mechanisms that balance the utility and risk of actions. In this work, we propose a novel collaborative multi-agent planning framework that integrates (i) a team-level online planner under uncertainty that extends the classical UCT approximate algorithm, and (ii) a preference modeling and multi-criteria group decision making approach that allows agents to find accepted and rational solutions for planning problems, predicated on the attitude each agent adopts towards risk. When utilised in risk-pervaded scenarios, the proposed framework can reduce the cost of reaching the common goal sought and increase effectiveness, before making collective decisions by appropriately balancing risk and utility of actions.**

I. Introduction

Planning, where we try to come up with a series of actions to achieve a goal sought by the agent [1], is an essential component of autonomous agents. In realistic environments the size and complexity of the problem is often a challenge, one that is further aggravated when considering multiple agents which must coordinate and act in parallel. This is the domain of multi-agent planning, referring to a family of problems which require "*planning by and for multiple agents*" [2]. In collaborative multi-agent planning in particular, a team of agents combine their capabilities and beliefs to jointly complete a task leading to a common goal [3]. However, such planners typically do not consider important environmental information such as the uncertainty of actions (e.g. a door may not open), or the risk associated with actions (e.g. investing in a start-up may be rewarding or lead to bankruptcy). The authors in [4] address this issue by adopting the notion of risk defined as *the possibility of obtaining a utility (reward) lower than the expected utility, due to an undesired outcome of taking an action*. The authors then proposed an extension of existing first principles planners to provide an agent with the ability to (i) assess both risk and utility (reward) of available actions, and (ii) make rational decisions by striking a balance between utility and risk, based on the attitude of the agent towards risk. However, their work only considers the single agent setting.

This work focuses on developing a collaborative multi-agent planner, that takes into account the different "*points of view*" of each agent and their distinct risk tolerance levels. To scope our work, we furthermore set out the following three principles:

Principle 1 Agents act in parallel, with one agent designated as the team leader. While the team leader does the actual planning, it is the group that decides on the best next action to take.

Principle 2 Agents act purely collaboratively (with no form of inter-agent competition being considered), and the actions of an agent do not interfere with the actions of other agents in the team.

Principle 3 All agents have similar competencies and capabilities, i.e. they can perform the same set of actions with equal probabilities of outcomes for each action.

To tackle the huge search space often involved in multi-agent planning, we also need to rely on techniques such as online planning in which planning and execution is interleaved. Monte-Carlo Tree Search (MCTS) algorithms [5], [6] continuously explore the search space, yet can return a "*good enough*" action (rather than a complete series of actions) at any time. To date, few online approaches have been applied to multi-agent planning domains [7], [8]. Importantly, to the best of our knowledge none of these works consider the possibility of assessing risk and the utility of actions jointly.

The framework we propose and evaluate in this paper to address these challenges is defined as follows. A planner agent (the team leader) determines the best possible actions to be performed by every agent as a team. To this end we extend the classical UCT (*Upper Confidence bounds applied to Trees*) version of MCTS [6] to effectively manage information at team level, while assessing the utility and risk of possible actions during search. Unlike the single-agent, non risk-aware setting where only a single action is considered, a *set of (team) actions* deemed "*good enough*" (with their associated reward and risk estimates), are selected. Subsequently, "candidate" actions are analysed by each agent to find a common accepted solution to the planning problem [9] through a collective multi-criteria decision making stage, where each agent balances the utility and risk of each possible action at its current state, based on its own risk tolerance level. Specifically, a team preference is computed by aggregating individual assessments of (a subset of) the available actions [10], [11]. The resulting preference is finally used to return *one* team-level action deemed as the most satisfactory decision.

The main contributions of our work are therefore as follows:

1) We propose a framework in which multiple agents can

collaborate as a team. The multi-agent planning component takes the reward *as well as* the risk of each team action into account, and a final decision is reached by taking account of the individual concerns by *all* the agents in the team [12].

2) We propose the first online planning algorithm that efficiently solves multi-agent planning problems involving risk assessment and multi-criteria group decision-making.

This paper is set out as follows: Section II provides an overview of basic concepts and ideas underpinning online planning and decision making. In Section III, the scenario used to illustrate our framework is introduced. A novel, risk-aware online multi-agent planner is proposed in Section IV, and subsequently integrated with a decision making approach based on the risk tolerance of an agent in Section V. Section VI illustrates the practical use of the proposed framework, and finally, some concluding remarks are laid out in Section VII.

II. Preliminaries

This section provides an overview of online planning and the UCT algorithm, followed by basic concepts relating the decision making framework considered in our proposal.

A. Online Planning

Online planning approaches interleave planning with execution: instead of generating the whole plan a priori (as occurs with offline planners), online planners return a next "*good-enough*" action to be executed at the current state. Online planners are based on approximate *anytime* algorithms, and they provide time-sensitive results on the next actions to take under uncertainty. Our work integrates online planning with risk assessment and decision making mechanisms, in problems requiring cooperation to accomplish a common goal, and where the actions performed by agents have uncertain effects.

UCT [6], [13] is a state-of-the-art *anytime* algorithm widely utilised in planning domains pervaded by uncertainty, that combines MCTS [14] with multi-bandit selection methods [13]. The algorithm applies the following four steps: (i) *Selection*: select a child node based on a selection function. (ii) *Expansion*: randomly expand the selected node to a new unsampled one. (iii) *Rollout*: randomly simulate a playout (i.e. a sequence of selected actions and their outcomes) until reaching a terminal state. (iv) *Backpropagation*: compute a reward value associated with the terminal state reached, and propagate it back up to the root node, updating the cumulative reward values for each node in the path. A *decision node* in UCT represents an environment state. A decision node corresponding to a non-terminal state can be expanded into available actions (represented by *chance nodes*) at that state, leading in turn to child decision nodes for the outcomes of such actions. The root decision node represents the current environment state [6]. Every time a decision node is visited, the selection of the action to take is based on previous rollouts, such that actions that produced higher rewards, and actions rarely visited in previous rollouts, are both favoured. This enables algorithms such as UCT to find an elegant balance between exploitation (selecting actions with better reward statistics so far) and exploration (selecting actions that have still been rarely simulated).

B. Decision Making Framework

Decision making has long constituted an important process in human lives, consisting in the selection of the best or most suitable choice from a set of alternatives. Emergent AI techniques, including the development of deliberative multi-agent systems [15], have witnessed the necessity of incorporating rational decision making capabilities into autonomous agents. Multi-Criteria Decision Making (MCDM) refers to a family of methods to deal with decision problems under the presence of several, often conflicting criteria [16]. Agents have the potential to implement MCDM methodologies by [15]: (i) modeling consistent families of criteria, (ii) modeling preferences over alternatives to guide their decisions, and (iii) exploiting the decisions made to guide their actions.

The MCDM framework considered in this work (presented in Section V) is formulated as follows:

- There exists a decision problem, consisting of $m \geq 2$ alternatives or possible solutions, $X = \{x_1, \ldots, x_m\}$, e.g. different actions to be chosen by a team of agents.
- Alternatives are assessed according to several independent criteria, $Q = \{q_1, \ldots, q_z\}$, $z \geq 2$. For instance, criteria for assessing a team-level action in our framework include its associated reward and risk estimates.
- Each agent constructs a preference structure, in our case a numerical preference vector in the unit interval, $P_i = [p_i^1 \ \ldots \ p_i^m]$ to evaluate alternatives, with $p_i^j \in [0,1]$ the degree of preference of alternative x_l by agent i.

In MCDM, alternatives are evaluated according to each criterion separately, with $p_i^{j,l}$ the degree to which x_j satisfies criterion q_l, $j \in \{1, \ldots, m\}$, $l \in \{1, \ldots, z\}$. Therefore, an aggregation function $f : [0,1]^z \rightarrow [0,1]$ must be utilised to combine satisfaction degrees over criteria, $p_i^{j,1}, \ldots, p_i^{j,z}$, into an overall one, with p_i^j. Aggregation functions accomplish the following properties [17]:

1) *Boundary condition*: $f(0, \ldots, 0) = 0$ and $f(1, \ldots, 1) = 1$.
2) *Non-decreasing*: $(a_1, \ldots, a_z) \leq (b_1, \ldots, b_z)$, implies $f(a_1, \ldots, a_z) \leq f(b_1, \ldots, b_z)$.
3) *Identity when unary*: $f(a) = a$, $\forall a \in [0,1]$.

Examples of (families of) aggregation functions include averaging functions, conjunctive functions (t-norms), disjunctive functions (t-conorms), mixed functions e.g. uninorms, etc [18].

Another common decision framework that has attained significant research interest is that of Group Decision Making (GDM) problems, in which multiple individuals (e.g. agents) must combine their own preferences to make an accepted decision together. Classically, the resolution process for GDM approaches involves an *aggregation phase* that combines individual preferences $P_i, \ldots, P_n$ into a group or collective preference P_c, and an *exploitation phase*, that utilises the group preference to obtain an alternative or subset of alternatives as the solution for the GDM problem [12].

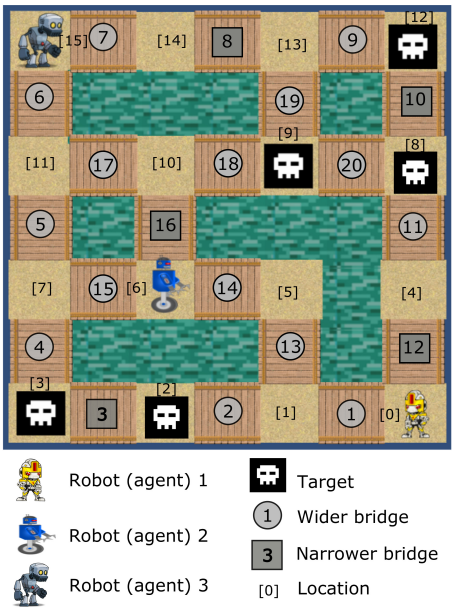

Fig. 1. Example scenario

However, the need for highly accepted solutions that minimise the possibility of disagreement between preferences of group members, has led to the appearance of consensus reaching approaches, introducing an additional phase aimed at bringing such preferences closer to each other before making a decision [19]. For the interested reader, an exhaustive overview of consensus-reaching approaches for GDM can be found in [12].

III. Scenario Overview

The nuclear navigation scenario serves to illustrate the multi-agent framework presented in this paper. A team of robots (agents) are situated in different locations of a nuclear site. A number of anomalies (targets) are detected in locations around the plant. The robots, which operate concurrently, must plan, coordinate and make decisions together to move toward the targets and address the issues efficiently. The nuclear site is organised into a number of locations and a network of bridges to move between them (see Figure 1). Some bridges are wider (and hence safer to cross) than others, and falling off a bridge will permanently disable the robot. For simplicity, we assume each robot has similar competencies in terms of their probability of successfully crossing a bridge. Furthermore, the robots are fully aware of their location within the site at all times, as well as the locations of the incomplete targets, and they also communicate any changes in their location or action outcomes to the team planner agent. Depending on their individual status, agents may have different attitudes towards *risk* and, consequently, diverse preferences over the available actions to perform. By taking account of the individual preferences of agents, an accepted collective decision should be made on the actions to be executed.

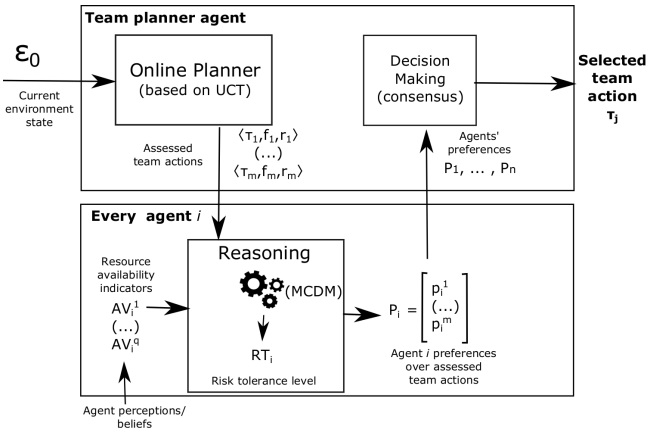

Fig. 2. Architecture of the risk-aware multi-agent planning framework

IV. Risk-Aware Online Multi-Agent Planner

This section presents a novel multi-agent planning framework for collaborative, uncertain settings. The main contribution of the underlying online planning algorithm is the extension of the single-agent approach in [4] to assess risk of actions alongside their utility at team level. The proposed framework (depicted in Figure 2) is further combined with a collective MCDM approach for action selection at group level, as explained later in Section V.

A. Notation and Basic Concepts

The following notation is introduced to refer to the elements utilised in the proposed planner. There exists a set $\mathcal{AG} = \{1, 2, \ldots, n\}$ of *agents*, and a finite *action library* shared by all agents. Each action $a_k, 1 \leq k \leq m$, is modeled as a tuple $\langle a_k, \phi_k, eff_k \rangle$. ϕ_k represents the preconditions for the action to be applicable, with $eff_k = \{(\epsilon', p)\}$ the possible (mutually exclusive) effects or outcomes ϵ' of the action at individual level, and their associated probability p. The set of all possible environment states is denoted by $\mathcal{E}$, where $\epsilon_0 \in \mathcal{E}$ is the current state (root decision node in the search tree). The subset of all goal states is denoted by $\mathcal{E}_G \subset \mathcal{E}$. In the proposed team planning approach, the team planner agent must manage information about multiple agents jointly, therefore it needs to formulate environment states at team level. Thus, a decision node is modeled upon the following two elements:

1) Agent-specific information about the state of every agent involved in the planning process, e.g. the current locations of robots in the nuclear scenario.
2) Other purely environmental information, e.g. the locations of unfixed targets (if any) in the nuclear site.

Based on this, a decision node associated to an environment state $\epsilon \in \mathcal{E}$ is formalised as a 2-tuple $D(\epsilon) = \langle s(\mathcal{AG}); s(env) \rangle$, with $s(\mathcal{AG})$ the current state of every agent and $s(env)$ the environmental information.

Example 1: Consider the nuclear navigation scenario. The state $s(i)$ of agent i is defined as a predicate of the form, $at(i, L)$, with L the location of agent i (either one of the 16 locations numbered 0-15, or "–" to indicate that i failed

in the execution of an action and is no longer available). Let $s(env) = \wedge_{at(target,L)} L$ be the locations of targets (anomalies) not addressed yet. A decision node describing the state depicted in Figure 1 can be formalised as follows:

$$D(\epsilon) = \langle \{at(1,0), at(2,6), at(3,15)\}; 2 \wedge 3 \wedge 8 \wedge 9 \wedge 12 \rangle$$

i.e. agents 1,2, and 3 are located respectively in zones 0,6,15. Furthermore, environmental information $s(env)$ indicates the existence of untreated targets in locations 2,3,8,9,12.

Example 2: Let a_k be the action of moving from location 0 to location 4 (i.e. crossing bridge 12 upwards) in the nuclear scenario. Its corresponding formalisation is:

$$\langle mv\_0\_4, at(0), \{(at(4), 0.95), (at(-), 0.05)\} \rangle$$

The precondition for an agent to execute this action, $\phi_k = at(0)$, is being situated in location 0. The action has two possible outcomes: (i) reaching location 4, with 95% probability; or (ii) failing to complete the action, with 5% probability.

The representation of states and actions in our planning framework is based on PPDDL (Probabilistic Planning Domain Definition Language) [20], which is fully compliant with implementations of approximate algorithms based on MCTS. We now introduce the concept of team action. This concept plays a central role in the proposed multi-agent planner.

Definition 1: A team action $\tau = \{(i, a_k^i), i \in pa(\tau)\}$ encompasses a number of actions $a_k^i \in \mathcal{A}$ simultaneously assigned to a team of agents $pa(\tau) \subseteq \mathcal{AG}$ (one action per agent). Team actions are formulated during planning, taking account of the current state and available actions per agent.

Example 3: Assuming the environment state represented in Figure 1, $\tau = \{(1, mv\_0\_4), (2, mv\_6\_7), (3, mv\_15\_14)\}$ indicates that the robot 1 must move from 0 to 4, robot 2 must move from 6 to 7 and robot 3 must move from 15 to 14.

B. Risk-Aware Online Team Planner

We now discuss how the online risk-aware planner introduced in [4] can be extended to deal with multiple collaborating agents. To do this, we propose an online multi-agent planner that (i) assesses both the risk and utility of actions from a team of agents, and (ii) returns a set of team actions with their associated utility and risk assessments, instead of returning a single *best* team action. We consider a MCTS-based search tree structure, with layers alternating between decision nodes, $D(\epsilon)$, representing environment states ϵ, and chance nodes, $C(\tau)$, representing team actions τ. The children of a decision node reflect the team actions available at ϵ. Conversely, the children of a chance node indicate the stochastic outcomes or resulting environment states from applying τ.

Below we describe how the outcomes of team actions and their probabilities of occurrence are determined. Having multiple agents executing a team action in parallel may involve a large number of possible outcomes, therefore we firstly discuss how the representation of such outcomes can be simplified, in order to prevent an excessive branching factor in the search tree. We assume two types of outcome for any τ: *success outcome*, ϵ_τ, when *all* participating agents succeed in completing their respective actions, and *undesired outcome*, $\epsilon_{\overline{\tau}}$, otherwise. The undesired outcome of τ encompasses all the possible eventualities ϵ_F that may lead τ into failure (i.e. one or more agents in $pa(\tau)$ failing to complete their assigned action). Therefore, $\epsilon_{\overline{\tau}} = \bigcup_{\epsilon_F \in \epsilon_{\overline{\tau}}} \epsilon_F$.

Importantly, the number of all possible undesired outcomes ϵ_F described by $\epsilon_{\overline{\tau}}$ directly depends on the number of agents participating in the team action, $|pa(\tau)|$. Concretely, it is given by the number of possible subsets of agents, $fa(\tau) \subseteq pa(\tau)$, that might succeed in completing their action, i.e. $|\epsilon_{\overline{\tau}}| = 2^{|pa(\tau)|} - 1$. Both goal states $\epsilon_G \in \mathcal{E}_G$ (which result from completing a sequence of team actions until reaching the goal established) and undesired outcomes $\epsilon_{\overline{\tau}} \in \mathcal{E}_F$ (with $\mathcal{E}_F \subset \mathcal{E}$ the set of all undesired outcomes) are *terminal states*, with $\mathcal{E}_T = \mathcal{E}_G \cup \mathcal{E}_F$ the set of all terminal states.

Remark 1: A (summarised) undesired outcome $\epsilon_{\overline{\tau}}$ is deemed as a terminal state, because if an unexpected situation is encountered, the team planner agent starts another online planning process for the remaining agents upon the resulting environment state, taken as the new ϵ_0.

Probabilistic information of individual actions must be combined to describe the effects of team actions. Let $P(\tau)$ be the probability of successfully completing τ, and $P(\overline{\tau})$ the probability of reaching any form of undesired outcome. Actions a_i^k assigned to every agent $i \in pa(\tau)$ are regarded as independent from each other, hence $P(\tau)$ can be easily calculated upon the individual action library information, as $P(\tau) = \prod_{i \in pa(\tau)} P(a_i^k)$, with $P(a_i^k) = p \in [0,1]$ being the probability of reaching the expected (successful) effect of executing the agent action $a_i^k = \langle a_k, \phi_k, \{(\epsilon', p), (\overline{\epsilon}', 1-p)\} \rangle$. Intuitively, $P(\overline{\tau}) = 1 - P(\tau)$.

Below we introduce a reward function that allows for a reduced branching of the search tree by estimating a single reward value for all possible forms of undesired outcome.

Definition 2: Let $\mathcal{E}_T = \mathcal{E}_G \cup \mathcal{E}_F$ be the set of all terminal states, as defined above. A reward function f is defined as a mapping $f : \mathcal{E}_T \rightarrow [-1, 1] \backslash \{0\}$, with the following properties:

(i) $f(\epsilon_G) > 0$, $\forall \epsilon_G \in \mathcal{E}_G$, i.e. arriving at a goal state always produces a positive reward value.

(ii) $f(\epsilon_{\overline{\tau}}) < 0$, $\forall \epsilon_{\overline{\tau}} \subset \mathcal{E}_F$, i.e. arriving at any undesired outcome always produces a negative reward value.

(iii) Let $d \in \mathbb{N}$ be the depth level at which the terminal state is encountered. Assume two identical terminal states ϵ_1, ϵ_2 can be reached at depth d_1 and d_2 respectively, with $d_1 < d_2$. Then $f(\epsilon_1) \geq f(\epsilon_2)$.

(iv) $f(\epsilon_G) > f(\epsilon_{\overline{\tau}})$ for any $\epsilon_G \in \mathcal{E}_G, \epsilon_{\overline{\tau}} \in \mathcal{E}_F$.

According to (iii), a goal state is more rewarding when encountered after a lower number of team actions. Similarly, an undesired outcome is more detrimental when more effort is previously invested, i.e. after more actions. A discount factor $\delta \in]0,1[$ is applied on f to reflect this property. The reward for an undesired outcome is calculated as follows:

$$f(\epsilon_{\overline{\tau}}) = -\delta^{d-1} \frac{\sum_{\epsilon_F \in \epsilon_{\overline{\tau}}} P_\tau(\epsilon_F) \cdot f(\epsilon_F)}{\sum_{\epsilon_F \in \epsilon_{\overline{\tau}}} P_\tau(\epsilon_F)} \quad (1)$$

Clearly, $f(\epsilon_{\overline{\tau}})$ is calculated as the (discounted) probability-weighted average of all possible forms of undesired outcome, $\epsilon_F \in \epsilon_{\overline{\tau}}$, which correspond to each of the non-empty subsets $fa(\tau)$ of agents that fail to complete their associated action, such that $fa(\tau) \in \mathcal{P}(pa(\tau))\backslash\{\emptyset\}$. Their probability of occurrence, denoted by $P_\tau(\epsilon_F)$, is easily calculated upon individual agent action information: $P_\tau(\epsilon_F) = \prod_{i \in pa(\tau)} P_i(\epsilon_F)$, where for each a_i^k assigned to i through τ,

$$P_i(\epsilon_F) = \begin{cases} 1 - P(a_i^k) & \text{if } \epsilon_F \vDash at(i,-), \\ P(a_i^k) & \text{otherwise.} \end{cases} \quad (2)$$

The reward value for each ϵ_F is computed based on the amount of failing agents in the team, i.e. $f(\epsilon_F) = \frac{|fa(\tau)|}{|pa(\tau)|}$. This non-negative value is only a partial step in the calculation of the overall negative reward for $\epsilon_{\overline{\tau}}$ (Eq. (1)).

Since we consider a collaborative setting with a common goal pursued by all agents, the reward of a goal state is defined based on the discount factor δ and its depth d, as $f(\epsilon_G) = \delta^{d-1}$, i.e. the sooner the goal is accomplished (lower cost of executing actions), the more beneficial the outcome is.

Having defined the reward function, we now describe the procedure to assess risk, which extends the one in [4].

Definition 3: The immediate risk of taking a team action τ at state ϵ is the probability-weighted variance[1] of its outcome rewards:

$$IR(\epsilon,\tau) = P(\tau)(f(\epsilon_\tau) - E(\epsilon,\tau))^2 + P(\overline{\tau})(f(\epsilon_{\overline{\tau}}) - E(\epsilon,\tau))^2 \quad (3)$$

with $E(\epsilon,\tau)$ the expected utility of taking τ at ϵ:

$$E(\epsilon,\tau) = P(\tau)f(\epsilon_\tau) + P(\overline{\tau})f(\epsilon_{\overline{\tau}}) \quad (4)$$

The success outcome of taking τ at ϵ is denoted by ϵ_τ.
Eq. (1) allows to determine $f(\epsilon_{\overline{\tau}})$, but $f(\epsilon_\tau)$ can not be directly calculated unless $\epsilon_\tau \in \mathcal{E}_G$. Instead, reward values of non-terminal states are calculated during the backpropagation phase. The immediate risk calculated by Eq. (3) is a measure associated to chance nodes (i.e. team actions). In decision nodes, however, the team of agents has a choice of which team action to execute. Therefore, we now define the immediate risk associated to a decision node.

Definition 4: Given a state ϵ and its set of immediately available team actions, $Av(\epsilon)$, the immediate risk exposure under a rational decision making perspective, is given by the immediate risk of the least risky team action available at ϵ:

$$RE(\epsilon) = min_{\tau \in Av(\epsilon)} IR(\epsilon,\tau) \quad (5)$$

The measure defined above considers the risk of *immediate* team actions only, disregarding further actions beyond these, hence we modify it to assess an average cumulative risk upon the reward and immediate risk of courses of action.

Definition 5: The cumulative risk exposure at state ϵ is defined as:

$$CRE(\epsilon) = min_{\tau \in Av(\epsilon)} CMR(\epsilon,\tau) \quad (6)$$

[1]Since we consider the use of approximate algorithms, the obtained variances are in practice suitable approximations.

with $CMR(\epsilon,\tau)$ the cumulative minimum risk of taking a team action τ at ϵ, calculated as follows:

$$CMR(\epsilon,\tau) = \frac{IR(\epsilon,\tau) + CMR_{old} \cdot visits(C(\tau))}{visits(C(\tau)) + 1} \quad (7)$$

$visits(\cdot) \in \mathbb{N}$ is the number of times a node has been visited. When a chance node is firstly visited, $CMR(\epsilon,\tau) = IR(\epsilon,\tau)$. The *selection* and *expansion* phases are applied similarly to plain UCT, and multiple risk *rollouts* are applied at each UCT iteration for resampling purposes, as explained in [4]. The *backpropagation* phase is applied by updating both reward and risk estimated in an average cumulative fashion:

- The reward of a non-terminal state $f(\epsilon)$ is updated every time $D(\epsilon)$ is visited during backpropagation, taking rewards of successive team action outcomes into consideration. Thus, $f(\epsilon)$ is interpreted as the average cumulative reward of arriving at this state:

$$f(\epsilon) = \frac{f(\epsilon^*) + visits(N(\epsilon)) \cdot f_{old}(\epsilon)}{visits(N(\epsilon)) + 1} \quad (8)$$

  Here, $f(\epsilon^*)$ is the reward of the expected (success) outcome of the least risky action in $Av(\epsilon)$.
- The updated risk estimate of a chance node in the backpropogation path is compared to that of its sibling nodes, and the risk of the sibling chance node with lowest risk estimated at that level is backpropagated.

The planner finally returns (a subset[2] of) team actions with their associated risk and reward estimates, rather than a single, most rewarding team action. These *assessed team actions* are subsequently evaluated by participating agents (Section V).

The following example illustrates the calculation and backpropagation of reward and risk estimates back to the root node.

Example 4: Consider the search tree excerpt depicted in Figure 3, where the nodes corresponding to τ_5 and its outcomes have been newly expanded. As a result of a rollout, a goal state with reward $f(\epsilon_G) = 0.7$ is encountered. This reward is backpropagated straightaway up to the last decision node generated, $N(\epsilon_5)$. The immediate risk of the predecessor chance node, $C(\tau_5)$, is calculated (Eq. (3)), resulting in $IR(\epsilon_1,\tau_5) = 0.09$. Since this is the first time $C(\tau_5)$ is visited, its cumulative minimum risk is trivially $CMR(\epsilon_1,\tau_5) = IR(\epsilon_1,\tau_5) = 0.09$. This value is compared to that of its existing sibling node so far, and it is lower than $CMR(\epsilon_1,\tau_4) = 0.59$, it is backpropagated as the cumulated risk exposure at ϵ_1, $CRE(\epsilon_1) = 0.09$. The reward at this state is updated (Eq. (8)) based on its previous reward, the reward being backpropagated, and the visit count, resulting in $f(\epsilon_1) = (0.65 \cdot 2 + 0.7 \cdot 1)/3 = 0.67$. The chance node $C(\tau_1)$ has been previously visited, hence both the immediate and cumulative minimum risk of τ_1 are updated by using Eqs. (3) and (7), respectively. By comparing $CMR(\tau_1)$ with that of its sibling nodes, $CMR(\tau_3) < CMR(\tau_1)$, therefore the estimates in the parent node, $f(\epsilon_0)$, $CRE(\epsilon_0)$, are updated by backpropagating estimates from τ_3 (instead of τ_1) in this case.

[2]If the number of immediately available team actions is large, those ones with lowest reward estimates (e.g. below a threshold) can be left out.

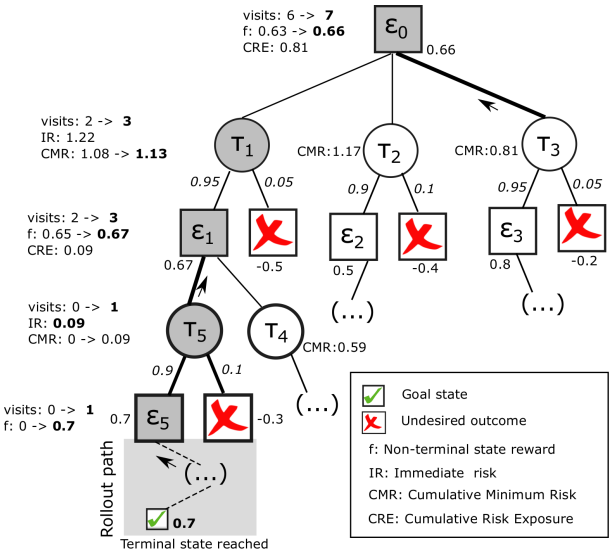

Fig. 3. Reward calculation and backpropagation (see Example 4)

V. Risk-Aware Multi-Agent Decision Making

This section describes the multi-criteria group decision making approach applied by agents to jointly select one of the candidate team actions returned by the planner. Firstly, risk attitude-based reasoning process conducted by agents to assess the available team actions. A GDM procedure is then undertaken to collectively select the best team action.

A. Preference Modeling upon Multiple Criteria

In order to rationally evaluate team actions, every agent i needs to determine its risk tolerance level $RT_i \in [0,1]$, calculated based on the availability of $q \geq 1$ resources deemed as relevant by i. Resource availability determines the attitude the agent should adopt towards risk, and the availability of each resource is viewed as the satisfaction degree of a criterion, under an MCDM perspective. RT_i is computed by using a function ρ_i that aggregates resource availability levels:

$$RT_i = \rho_i(AV_i^1, \ldots, AV_i^q) \tag{9}$$

Each argument $AV_i^k \in [0,1]$, $k \in 1 \ldots q$, is an availability indicator of the k-th resource: the larger its value, the higher the availability. Moreover, from the non-decreasing property fulfilled by ρ_i, the higher any of the AV_i^k is, the closer RT_i is to one, hence the more tolerant agent i is towards risky actions. The aggregation function ρ_i can be customised to suit each specific scenario and agent, by establishing the resource levels relevant to each agent and the way its associated indicators are determined. This allows agents to flexibly consider different (sets of) resources when calculating RT_i.

Example 5: Consider the nuclear navigation scenario. An agent i utilises the following four availability indicators to assess its risk tolerance level: AV_i^1, remaining battery life; AV_i^2: remaining time; AV_i^3, number of agents still operating; AV_i^4, number of anomalies addressed in the nuclear plant so far. In order to express these indicators as values in [0,1], they can be easily defined as percentages with respect to a full battery level, the total available time, number of agents in the team and anomalies initially detected, respectively.

Examples of aggregation functions that can be utilised by i to aggregate availability levels into RT_i include: (i) arithmetic mean; (ii) weighted mean; or (iii) the Ordered Weighted Averaging (OWA) operator [21], in which elements are firstly arranged in decreasing order, and importance weights $W = \{w_1, \ldots, w_q\}$ ($\sum_k w_k = 1$) are assigned to ordered elements. OWA operators allow to reflect different optimistic (*resp.* pessimistic) attitudes in the aggregation process, depending on importance weights being rather assigned to the highest or lowest elements to aggregate. Yager defined an measure of optimism, $\mathcal{O}_W \in [0,1]$, to categorise OWA operators [22]:

$$\mathcal{O}_W = \frac{\sum_{k=1}^{q}(q-k)w_k}{q-1} \tag{10}$$

Optimistic (OR-like) OWA operators accomplish $\mathcal{O}_W > 0.5$. Conversely, pessimistic (AND-like) operators fulfill $\mathcal{O}_W < 0.5$, and neutral operators fulfill $\mathcal{O}_W = 0.5$.

Example 6: Consider agent i from the previous example. Assume its remaining battery life is 70% ($AV_i^1 = 0.7$), half of the time limit elapsed ($AV_i^2 = 0.5$), all agents still operate ($AV_i^3 = 1$) and two out of five anomalies have been dealt with ($AV_i^4 = 0.4$). The agent utilises the OWA operator to calculate RT_i, by adopting a slightly optimistic, OR-like aggregation attitude [21] given by the vector $W = \{0.3, 0.3, 0.2, 0.2\}$:

$$RT_i = \mathbf{1} \cdot 0.3 + \mathbf{0.7} \cdot 0.3 + \mathbf{0.5} \cdot 0.2 + \mathbf{0.4} \cdot 0.2 = 0.69$$

Notice that the optimistic attitude stems from larger importance weights being associated to the two largest arguments, 1 and 0.7. Furthermore, $\mathcal{O}_W = 0.566 > 0.5$.

Given RT_i and the m assessed team actions $\langle \tau_j, f_j, r_j \rangle$ returned by the online planner, $j = 1, \ldots, m$, each agent proceeds to construct a preference vector $P_i = [p_i^1\ p_i^2\ \ldots\ p_i^m]$, where a rating $p_i^j \in [0,1]$ indicates its satisfaction degree with τ_j: the higher p_i^j, the more satisfied i is with τ_j. The assessed reward f_j, assessed risk r_j, and the agent attitude towards risk are three determinant criteria to compute a rating for each team action. Therefore, p_i^j is calculated as a function $\pi : [0,1] \times [0,1] \times \mathbb{R} \to [0,1]$ of such criteria:

$$p_i^j = \pi(RT_i, f_j, r_j)$$

with π accomplishing the following three properties:

i. If $RT_i > 0.5$ the agent adopts a *risk tolerant* attitude, tending to favor team actions with higher reward.
ii. If $RT_i < 0.5$ the agent adopts a *risk averse* attitude, tending to favor team actions with lower risk.
iii. If $RT_i = 0.5$ the agent has a *risk neutral* attitude, deeming reward and risk of team actions as equally important criteria.

These properties are fulfilled by defining π as a combining function that aggregates information related to the assessed reward and risk, and weighs these two criteria based on the

agent risk tolerance level. This results in deriving a preference degree that appropriately balances utility and risk:

$$p_i^j = RT_i \cdot \overline{f}_j + (1 - RT_i) \cdot (1 - \overline{r}_j) \tag{11}$$

Before applying Eq. (11), the assessed reward and risk of τ_j are normalised to take values in the unit interval:

$$\overline{f}_j = \frac{f_j - \min_k f_k}{\max_k f_k - \min_k f_k} \qquad \overline{r}_j = \frac{r_j - \min_k r_k}{\max_k r_k - \min_k r_k} \tag{12}$$

with $k \in \{1, \ldots, m\}$. Normalisation implies that for the assessed team action with highest (*resp.* lowest) reward, we have $\overline{f}_j = 1$ (*resp.* $\overline{f}_j = 0$). Similarly, for the most and least risky assessed team actions, $\overline{r}_j = 1$ and $\overline{r}_j = 0$, respectively.

Example 7: Consider $RT_i = 0.69$ from the previous example (i is slightly inclined towards rewarding team actions rather than low risk ones), and the following assessed team actions returned by the planner, $\langle \tau_1, 0.43, 1.08 \rangle$, $\langle \tau_2, -0.07, 0.36 \rangle$, $\langle \tau_3, 0.83, 1.45 \rangle$, $\langle \tau_4, 0.3, 1.15 \rangle$. Its associated preference vector is $P_i = [0.56 \ 0.31 \ 0.69 \ 0.51]$. The slightly risk tolerant attitude is reflected through the agent preference towards higher rewards, as occurs with τ_3 for instance.

B. GDM Approach for Team Action Selection

All active agents provide their preferences P_i to the team planner agent, which elicits them and defines a GDM problem on $\{P_1, \ldots, P_n\}$ aimed at making a common accepted solution on the next team action to undertake. Concretely, an aggregated team preference P_c that minimises the distance between individual preferences of agents and P_c, i.e. a consensus preference, is sought [12]. An automatic, iterative consensus-reaching approach is conducted, by applying the optimal preference aggregation method proposed by Lee in [23].

Let $d(P_i, P_h) \in [0, m]$ denote the dissimilarity between two preference vectors P_i,P_h, computed by using a Minkowski distance measure d, e.g. the Euclidean distance, given by:

$$d(P_i, P_h) = \sqrt{\sum_{j=1}^{m} (p_i^j - p_h^j)^2} \tag{13}$$

An approximation to an optimal team preference vector $P_c = [p_c^1, \ldots, p_c^m]$ that minimises the sum of (weighted) dissimilarities with individual preferences can be obtained by an iterative algorithm similar to Fuzzy C-means [23]. The algorithm weighs the preferences of agents assuming every agent preference P_i is initially regarded as equally important, assigning $w_i = 1/n, \forall i \in \{1, \ldots, n\}$

$$p_c^j = \frac{\sum_i (w_i)^\mu p_i^j}{(w_i)^\mu} \qquad w_i = \frac{(1/d(P_i, P_c))^{1/(\mu-1)}}{\sum_l (1/d(P_l, P_c))^{1/(\mu-1)}} \tag{14}$$

This process is iteratively applied, for each $p_c^j \in P_c$ and $w_i \in W$ respectively, until satisfying a stopping condition, e.g. when weights of agents preferences stabilise, i.e. $\|W^{(t+1)} - W^{(t)}\| \leq \kappa$, with t and $t+1$ two iterations of the consensus-reaching algorithm, and $\kappa \approx 0$, $\kappa > 0$ the threshold difference used as stopping criterion. The parameter $\mu \geq 1$ is utilised to control the influence of *noisy information*, i.e. agents preferences whose weight is low due to their preferences being situated far from consensus, compared to that of preferences with larger w_i: the larger μ, the stronger the difference between the influence made by agents positioned close and far from consensus. The convergence of the algorithm is thoroughly demonstrated in [23]. The resulting collective preference P_c is utilised to select the best[3] team action τ_* with $p_i^* = \max_j p_c^j$, as the "best" (most preferred) team action for its execution. This team action is finally executed by agents in $pa(\tau^*)$.

VI. Experiments and Results

In this section we demonstrate the performance of the proposed multi-agent planning framework. Throughout experiments, we refer to the nuclear navigation scenario (Section III, Figure 1) with three robots and five target anomalies.

We start by evaluating the overall performance of the framework by comparing it against three *baseline* approaches:

- B1: *Risk-aware planning, individual decision making*: Instead of making a collective decision, only the team planner agent evaluates candidate team actions by balancing reward and risk (based on its own attitude towards risk).
- B2: *Risk-aware planning, lowest-risk team action selection*: The team planner agent directly selects the lowest-risk available team action, i.e. without balancing reward and risk.
- B3: *Reward-driven planning*: Only the reward of team actions is assessed during planning (risk is not assessed), therefore the immediate team action with highest reward estimate is returned by the planner straightaway, with no need for subsequent decision making process.

10 different settings are considered for the (success) probabilities of agent actions, ranging between $p_w \in [0.86, 0.95]$ for crossing wider bridges, and with $p_n = p_w - 0.05$ for narrower bridges. The experiments were run 100 times, gathering the following two metrics:

- *Success Rate* (%): Number of executions where the goal of completing the five targets is achieved before *all* three robots fall off a bridge.
- *Average Reward when Successful*: Average reward f associated to the result of successful executions. To this end we calculate the reward of an outcome reached upon execution of actions, after which some agents might have failed. According to this, the $f(\epsilon_G)$ calculated during planning is adjusted by multiplying it by the proportion of "surviving" agents, $\#surv/3$. This metric provides an estimate of the cost invested in reaching the goal.

Figure 4 shows the results obtained for each metric and action probability setting. In general, balancing the reward and potential risk of team actions leads to higher chances of success. This becomes more noticeable as the success probabilities of agent actions decrease. The difference in

[3] *Best* is understood in this context as the most collectively accepted team action based on their individual preferences.

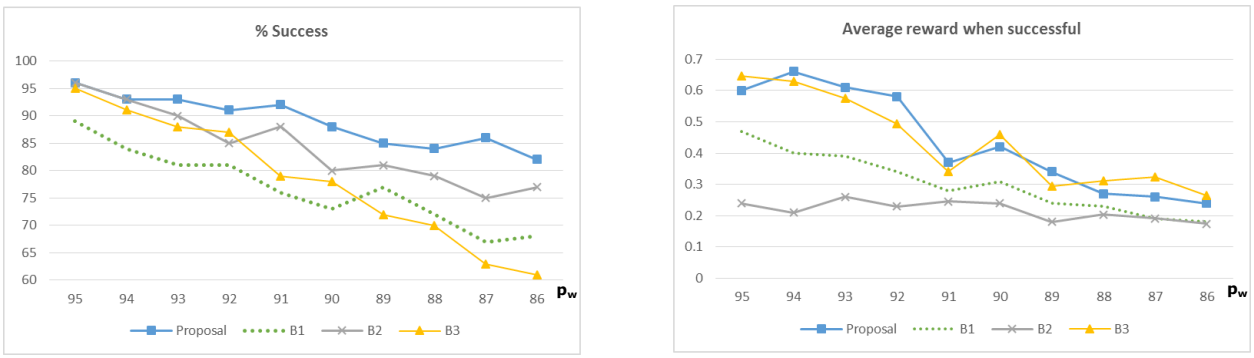

Fig. 4. Success rate (left) and average reward when successful (right) of the proposed risk-aware framework against three baseline approaches.

performance accentuates when compared with (B3), i.e. when the team planning process is purely reward-driven and risk is completely ignored. Results demonstrate that assessing risk alongside reward of team actions, and making more informed, context-aware decisions based on all agents' viewpoints towards risk, becomes increasingly beneficial in cooperative settings, particularly as the uncertainty of action effects increases. On the contrary, relying on only one agent to make decisions (B1) does not provide optimal results in multi-agent settings, thus the need for rational collective decision making mechanisms among agents is justified.

A larger average reward when successful is generally observed with the proposed risk-aware framework regardless of the agent action probabilities. Only in some cases the reward-driven baseline (B3) slightly outperforms in average reward, however this occurs at the cost of a much lower % success with respect to the other approaches being compared (as outlined above). The lowest-risk approach (B2) shows the most accentuated difference with our framework in terms of average reward when successful. Overall, results show that the cost of reaching the goal can be reduced when planning and making decisions predicated on risk assessment.

VII. Concluding Remarks

This work presented a collaborative multi-agent planning framework for domains where agent actions have uncertain effects, that integrates an online multi-agent planner capable of assessing risk alongside utility of actions, with a multi-criteria group decision making approach that enables the collective selection of an accepted solution for the planing problem. As a result, agents make rational decisions by balancing the risk and utility of actions based on their attitude towards risk. Future work aims at extensions to larger-scale scenarios through subgroup delegation and uncertain information fusion.

Acknowledgments

This work has been funded by EPSRC PACES project (Ref: EP/J012149/1).

References

[1] D. Weld, "Recent advances in AI planning," *AI Magazine*, vol. 20, pp. 93–123, 1999.

[2] M. de Weerdt and B. Clement, "Introduction to planning in multiagent systems," *Multiagent Grid Syst.*, vol. 5, no. 4, pp. 345–355, 2009.

[3] A. Torreño, O. Sapena, and E. Onaindia, "Global heuristics for distributed cooperative multi-agent planning," in *Proc. ICAPS 2015*. AAAI Press, 2015, pp. 225–233.

[4] R. Killough, K. Bauters, K. McAreavey, W. Liu, and J. Hong, "Risk-aware planning in BDI agents," in *Proc. ICAART'16*, 2016.

[5] B. Abramson, "The expected-outcome model of two-player games," Columbia University, Tech. Rep. CUCS-315-87, 1987, ph.D. Thesis.

[6] T. Keller and P. Eyerich, "PROST: Probabilistic planning based on UCT," in *Proc. ICAPS'12*, 2012.

[7] F. Wu, S. Zilberstein, and X. Chen, "Online planning for ad hoc autonomous agent teams," in *Proc. IJCAI 2011*, 2011, pp. 439–445.

[8] ——, "Online planning for multi-agent systems with bounded communication," *Artificial Intelligence*, vol. 175, no. 2, pp. 487 – 511, 2011.

[9] F. S. Melo and A. Sardinha, "Ad hoc teamwork by learning teammates' task," *Autonomous Agents and Multi-Agent Systems*, vol. 30, no. 2, pp. 175–219, 2015.

[10] M. Doumpos and E. Grigoroudis, *Multicriteria Decision Aid and Artificial Intelligence*. Wiley, 2013.

[11] J. Lu, G. Zhang, D. Ruan, and F. Wu, *Multi-Objective Group Decision Making*. Imperial College Press, 2006.

[12] I. Palomares, F. Estrella, L. Martínez, and F. Herrera, "Consensus under a fuzzy context: Taxonomy, analysis framework AFRYCA and experimental case of study," *Information Fusion*, vol. 20, no. November 2014, pp. 252–271, 2014.

[13] L. Kocsis and C. Szepesvári, "Bandit based monte-carlo planning," in *Proc. ECML'06*, 2006, pp. 282–293.

[14] C. Browne, E. Powley, D. Whitehouse, S. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A Survey of Monte Carlo Tree Search Methods," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, pp. 1–43, March 2012.

[15] P. Delias and N. Matsatsinis, *Multicriteria Decision Aid and Artificial Intelligence*. Wiley, 2013, ch. Multiple criteria decision aid and agents: Supporting effective resource federation in virtual organizations.

[16] B. Roy, *Multicriteria Methodology for Decision Aiding*. Dordrecht: Kluwer, 1996.

[17] G. Beliakov, A. Pradera, and T. Calvo, *Aggregation Functions: A Guide for Practitioners*. Springer, 2007.

[18] M. Detyniecki, "Fundamentals on aggregation operators," *LIP6 Research Report:2001-2002, University of California, Berkeley*, 2001.

[19] S. Saint and J. R. Lawson, *Rules for Reaching Consensus. A Modern Approach to Decision Making*. Jossey-Bass, 1994.

[20] H. Younes and M. Littman, "PPDDL1.0: An extension to PDDL for expressing planning domains with probabilistic effects," in *Proc. ICAPS'03*, 2003.

[21] R. Yager, "On ordered weighted averaging aggregation operators in multi-criteria decision making," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 18, no. 1, pp. 183–190, 1988.

[22] A. Kishor, A. K. Singh, and N. R. Pal, "Orness measure of owa operators: A new approach," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 4, pp. 1039–1045, 2014.

[23] H.-S. Lee, "Optimal consensus of fuzzy opinions under group decision making environment," *Fuzzy Sets and Systems*, vol. 132, no. 3, pp. 303 – 315, 2002.