Decision Making for Multi-Objective Multi-Agent Search and Rescue Missions

Hend AlTair[1], Tarek Taha[2], Jorge Dias[3] and Mahmoud Al-Qutayri[4]

***Abstract*— In this paper we introduce a novel rewarding scheme for the classical POMDP formulation. The proposed scheme aims to reinforce preference of objectives. It ensures that a high-priority preferences get high accumulative rewards, solve ambiguities and it can be conducted before the low-priority-preference. In order to show conflicting of objectives, context of search and rescue has been selected for this paper. It involves heterogeneous team with potential conflicting multi-objective situations. Our rewarding scheme has been tested in simulated scenarios using multiple POMDP solvers. Results obtained from the simulated experiments show that the system is able to represent an impact on the human (first responder) through enforcing priority of objectives and iteratively optimize policies to better suit the first responder and the search and rescue mission goals.**

I. Introduction

Deploying robots in hazardous situations reduces human exposure to dangerous environments and increases the efficiency of responding to these incidents. Examples of incidents might vary from small fires [1], oil spillage [2], earthquake [3], to nuclear incidents [4]. In emergencies, particularity the large scale ones, systems that allow seamless collaboration between teams of robots and humans are highly desirable [5]. A seamless collaboration requires a number of components: coordination (includes mission planning, task allocation and role assignment); context-awareness; and decision-making; are some of the main collaboration components.

In search and rescue decision-problem there are multiple objectives that can be easily conflicting with each other. A simple example of decisions is to decide between taking a longer path to rescue a victim instead of optimal path that has a danger. An agent in this case wants to stay away from danger and at the same time he wants to rescue the victim as soon as possible. The problem model is expressed with multiple objectives and therefore multiple rewards. A methodology based on rewarding procedure has recently been the focus that led to form other types of reward functions from state-based *R(s,a)* [6] to belief-based $\rho(b,a)$ [7] and hybrid reward function [8]. Multi-objective rewards has been introduced by Soh and Demiris [9]. The rewards are formed in a vector that consists rewards of all objectives. In order to solve the conflicting of the actions performed there should be a mechanism that ensures priority of objectives to be maintained so that specific policy will be applied before another. Wray and Zilberstein [10] introduced a new approach of POMDP utilizes the multi-objectives rewards (time and autonomy) to find a path for a single agent using lexicographic ordering. Roijers et. all proposed using Optimistic Linear Support (OLS) to calculate policy's multi-objective values which he also applied for single agent problems [11].

In this article we propose a method to priorities objectives in a reward function of Multi-Objectives Partially Observable Markov Decision Process (MOPOMDP). The method proposed in this paper is an alternative to the existing solutions with an easier implementation. It ensures that high-priority objectives are given higher accumulative rewards, so that a higher influence on the decision process than low-priority objectives. We applied it for multi-agent multi-objective problem which we did not see in the literature. The proposed method was evaluated in a Search and Rescue scenario that involves a heterogeneous team composed of a robot and human with different complimentary skills. The search and rescue operations incorporate search and rescue parameters such as *risk*, *energy* and *time* into the decision-making through a re-definition of the reward function.

The paper is organized as follows: first we briefly mention the POMDP model parameters. Then, we explain the multi-objectives reward functions, and introduce a reference search and rescue scenario to be used for evaluation. Finally, the results of 50 test cases are discussed followed by a summary of the findings and a discussing the future work.

II. Background

POMDP is an extension of Markov Decision Process (MDP) to partial observability. A POMDP for n agents is defined as $\langle n, \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma, O, \mathcal{O}, h, b^0 \rangle$ where $\mathcal{S}$ is a finite set of states. In addition, $\mathcal{A} = \times_i \mathcal{A}_i$ is the set $\{a^1,...,a^j\}$of J joint actions. $\mathcal{A}_i$ is the set of actions available for agent i. At each time step one a=$\langle a_1,...,a_n \rangle$ is taken. Furthermore, $\mathcal{T}$ the transition function which defined the probability of going to state s' when in state s under action a ,$p(s'|s,a)$. $\mathcal{R}(s,a)$ is the reward function. A reward is given for taking an action a when in state s. There is a γ that discounts the future reward. The partially observable part is added by O, the set of observations, and $\mathcal{O}$ which is the observation model $p(o|a,s')$. Finally, h is the horizon

[1]Hend AlTair, Robotics Institute, Khalifa University hend.altair@kustar.ac.ae

[2]Tarek Taha, Robotics Institute, Khalifa University tarek.taha@kustar.ac.ae

[3]Jorge Dias, Robotics Institute Khalifa University and Institute of Systems and Robotics, University of Coimbra, jorge.dias@kustar.ac.ae/jorge@deec.uc.pt

[4]Mahmoud Al-Qutayri, ECE, Khalifa University mqutayri@kustar.ac.ae

978-1-5386-0872-2/17/$31.00 ©2017 IEEE

and $b^0 \in \mathcal{P}(\mathcal{S})$, is the initial state distribution at time t. $\mathcal{P}(\mathcal{S})$ donates the set probability distributions over $\mathcal{S}$.

In POMDP's an agent's state is not directly observable, or observable with a noise. Therefore, a probability distribution, known as belief state *b(s)*, over all states should be maintained. At each time step the agent will have a belief state *b(s)*. The agent then takes an action a while observing o from the environment which will change its state from s to *s'*. In consequence, the belief state *b(s)* will be updated *b(s')*.

For multiple objectives problems it was first introduced in [9] the vector of reward functions instead of classical reward function. An accumulative discounted reward is counted for each objective. An approach to priorities a set of objectives was proposed using the goal programming lexicographic method in the reward preference [10]. It was applied for one agent. In this paper, we propose to use weighting factors in the reward function in a multi-agent multi-objective scenario.

III. Reward Function

At each horizon of the problem solving a joint reward is calculated for the multiple objectives. It is represented in equation 1

$$\sum_{n=1}^{N} R_n(S, A) * \alpha_n \qquad (1)$$

where N is number of objectives and α is a weighting factor which can be generated dynamically using an optimization algorithm. However, this is beyond the scope of this paper. In this paper we focus on the idea of using the weight method in order to reinforce preferences in multi objective reward. We have found specific weighting factors for the scenarios' objectives through a trial-and-error methodology. This paper discusses primarily results using these weights. In the following section we discuss our proposed algorithm to generate joint rewards of the multiple objectives. The algorithm enforce the rewards preferences. The algorithm is an initial attempt to use weighting factors in generating accumulative rewards.

This algorithm receives rewards that has been assigned for the objectives but without labelled priorities. The priority then passed to this algorithm which assign weights to priorities. After that the sum of all rewards is calculated where every weight is multiplied to the corresponding reward.

```
Data: Preferred priority of objectives
Result: Weights
for $x \leftarrow 1$ to $numOfObjectives$ do
    if $x \leftarrow 1$ then
        $weightPriority$(x) ←x;
    else if (x<=numOfObjectives / 2)and(x>1) then
        $weightPriority$(x) ← (x*0.9)/x;
    else if $x > (numOfObjectives/2)$ then
        $weightPriority$(x) ← (x/0.9)*0.3;
end
```

Algorithm 1: Pseudo code of Algorithm.

IV. Experiments and Validation

A. Priorities Rewards for Search and Rescue Operations

In this section we introduce a reference search and rescue scenario that we will use to evaluate the proposed changes to the reward function.

B. Scenario Description

In response to a search and rescue mission, a team has been dispatched in order to locate and extract a victim present in the rescue area. This team consists of one robot and one human (first-responder), each with different capabilities, but collectively are capable of dealing with the rescue situation. The robot is capable of scanning the area, locating source of danger, and clearing it. Although human has countless capabilities, he cannot sense nuclear or dangerous gases that has no smell. Our aim of this scenario is to keep the human first responder and the victim away from any danger source as far as possible. Therefore, in this scenario we limit the role of the first responder to evacuate the victim with less damage possible. The human has the capability to locate and extract the victim to an safe area. The problem has been modelled with a known map topology as depicted in Figure 1. Assuming that there is no collapse or change in environment. The topology of this map represents the connectivity between major intersection areas "nodes", and it's assumed to be known ahead. Only one victim is expected to be in the search and rescue area. For proof of concept reason, the problem is minimized so that the victim is in one node and it is assumed that she/he will not be moving. Similarly, there is one source of danger in one node. The multi-agent team will start from different locations in an attempt to locate the victim, and extract him/her to the safe evacuation location.

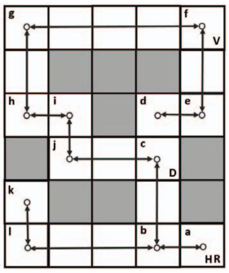

Fig. 1. Representation of 6x5 topological map where node *c* has the danger D, and node *f* has the victim V.

C. Model

The problem is modelled in a two slice Dynamic Bayesian network (DBN) shown in Figure 3.

D. States

The state space described in Table I is the cross product between the location of each of the two agents (one robots and one human), as depicted earlier in the topological map in Figure 1, and the possible locations of the victim and the danger.

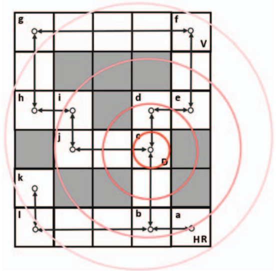

Fig. 2. Larger distance from the danger source means less risk which is illustrated with the circles on the 6x5 map

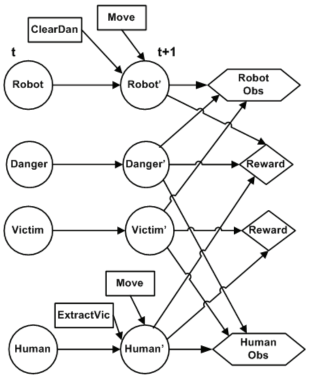

Fig. 3. DBN POMDP problem representation

E. Actions

This represent the set of joint actions that the agents can perform to achieve the common objectives. In our model, the robotic agent can perform the following set of actions: $A_r = \{Up, Down, Right, Left, Stop, ClearDanger\}$. The human can perform the following actions: $A_h = \{Up, Down, Right, Left, Stop, ExtractVictim\}$.

F. Observations

Once in a certain state/node, the agents can observe the presence of either a victim or a danger in the current node/location. This set is the cross product of whether the robot or the human sees a victim or a danger in a certain location: $O = \{yesVicNoDan, noVicYesDan, noVicNoDan\}$.

G. Reward Functions

Figure 4 illustrates how the two objectives compute the Reward output, based on different goals such as ”Risk” and ”Time”. Precisely, the reward function can be defined as tuple of four rewards $R :< R_{exv}, R_{cld}, R_r, R_t >$ where the reward for extracting victim R_{exv} and the reward for clearing danger R_{cld} are the positive rewards, defining the ultimate goals of this particular search and rescue scenario. On the other hand, the human rescuer is penalised for moving toward danger source through a negative reward function R_r. The danger zones are shown in Figure 2. Similarly, agents are penalised for travelling un-necessary steps (consuming extra energy and wasting precious rescue time) through the negative time reward function R_t .

TABLE I

STATE SPACE OF THE PROBLEM MODEL

Variable Type	$(Quantity)$ Name	Domain
State	(1) Robot	12 nodes
State	(1) Human	12 nodes
State	(1) Victim	1 node, nV
State	(1) Danger	1 node,nD

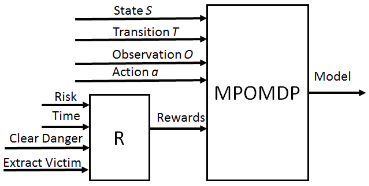

Fig. 4. Discounted Reward takes into account the different Search and Rescue parameters such as: time and risk

The typical extension of the reward function assigns reward knowing the action and start state $R(s, a)$. Our reward function is defined in Equation 2, where any action taken in a particular state will have rewards linked to each of the objectives, and α is a weighting factors used to tune the contribution of each objective reward in the total reward.

$$\begin{aligned} R(S, A) = R_{exv}(S, A) * \alpha_{exv} + R_{cld}(S, A) * \alpha_{cld} \\ + R_r(S, A) * \alpha_r + R_t(S, A) * \alpha_t \end{aligned} \tag{2}$$

In our modelling process, the map is known in advance, and we exploit that knowledge to assign the reward values. The risk reward function R_r is represented in Equation 3

$$R_r = \begin{cases} -50/D_d & , D_d > 0 \\ -50 & , D_d = 0 \end{cases} \tag{3}$$

where D_d is the distance from the human location to danger location. The time reward R_t considers the time of travelling assuming that the speed is constant

$$R_t = -2 * D_n \tag{4}$$

where D_n is distance from current location to next location. This might be an over simplification of the problem, but the intention is to generalize the problem once the reward modelling is finalized.

V. Simulated Experiments and Test Cases

In this scenario we used the minimum number of agents which are two with 4 objectives to be conducted. Simplifying the problem is the least we need to prove the concept we are proposing which we can solve multi-agent multi-objective POMDP with specific priority of objectives through using weighting factors. We have implemented a code to simulate critical 50 test cases from the states space. Since in the scenario there are 12 nodes (locations) a robot and human can be in any of those nodes. The priority of objectives that is passed to the models are ordered as follows: clear danger, extract victim, distance of human from danger, and finally time of traveling. In order to show the result we have conducted the same 50 test cases on the model proposed in this paper and another model which is a straight-forward model to enforce the priority without weighting factors. We refer to the model without weighting factor as *Model 1* and the one with weighting factor as *Model 2*.

VI. Results and Discussion

In this paper we show part of the results that we got from the experiments that has been conducted. Number of steps (time or actions) before conducting an objective with high priority is one of the ways to show the efficiency of the proposed algorithm. Figure 5 shows the results of the 50 test cases on the two models *Model 1* and *Model2*. The figure illustrates the total number of steps for the first two objectives(clear danger or extract victim) for both models. The number of steps to clear danger for both models are almost the same. However, for extracting victim which is the second objective the number of steps to execute this task in Model 2 is less than Model 1. This shows that the victim can be extracted faster using the proposed algorithm. This means that we can ensure to have less number of steps not only to achieve the objective with the highest priority but also to achieve the other objectives with the less steps possible. We have solved the model 2 (with the proposed algorithm) using two POMDP solvers; Perseus and Symbolic Perseus. Figure 6 illustrates the number of steps to conduct the same first two objectives (clear danger and extract victim). As it is shown that using Symbolic Perseus solver with the same proposed algorithm for weighting factors to generate rewards can have less number of steps for both objectives.

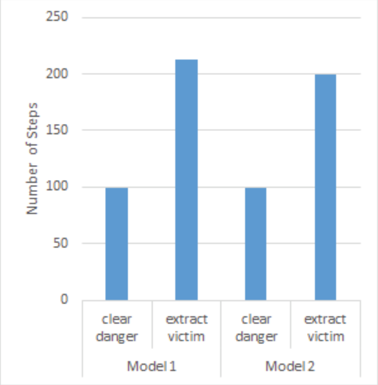

Fig. 5. Comparison between the number of steps it took the two models to achieve the first two objectives: clear danger and extract victim

VII. Conclusions and Future Work

In this paper we proposed an algorithm using weighting factors to apply multi-objective reward functions to better optimise common objectives in search and rescue scenarios. The results of the 50 test case shows that model generated using the proposed algorithm follows the preferred priority of the objectives and can conduct the objectives with high priority with less steps. This is a key promising solution and for future work we intend to apply an optimization algorithm that generalize the weighting factors. We are, also, planning to test the robustness and the scalability through conducting experiments on different problems and with different number of objectives.

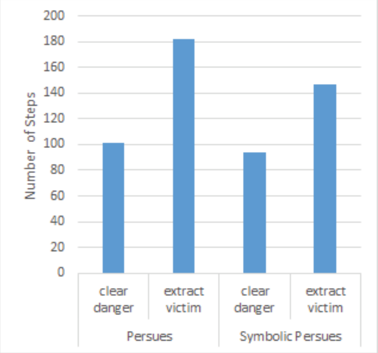

Fig. 6. Comparison between the number of steps it took models solved by Persues and Symbolic Persues to achieve the first two objectives: clear danger and extract victim

References

[1] G. Randelli, L. Iocchi, and D. Nardi, "User-friendly security robots," in *IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2011, pp. 308–313.

[2] V. Callaghan, G. Clarke, M. Colley, H. Hagras, J. S. Y. Chin, and F. Doctor, "Inhabited intelligent environments," *BT Technology Journal*, vol. 22, no. 3, pp. 233–247, 2004.

[3] G.-J. M. Kruijff, V. Tretyakov, T. Linder, F. Pirri, M. Gianni, E. Pianese, S. Corrao, P. Papadakis, M. Pizzoli, and A. Sinha, "Rescue robots at earthquake-hit mirandola, italy: a field report," in *IEEE International Symposium on Safety, Security, and Rescue Robotics*, vol. 1, 2012. [Online]. Available: http://www.dis.uniroma1.it/gianni/pubblications/SSRR2012.pdf

[4] E. Guizzo, "Fukushima robot operator writes tell-all blog," Aug. 2011. [Online]. Available: http://spectrum.ieee.org/automaton/robotics/industrial-robots/fukushima-robot-operator-diaries

[5] V. Robotics, "A roadmap for us robotics: From internet to robotics, 2013 edition," *Accessed online October*, vol. 23, 2013.

[6] M. T. Spaan, "Cooperative active perception using pomdps," in *AAAI 2008 workshop on advancements in POMDP solvers*, 2008.

[7] M. Araya, O. Buffet, V. Thomas, and F. Charpillet, "A pomdp extension with belief-dependent rewards," in *Advances in Neural Information Processing Systems*, 2010, pp. 64–72.

[8] A. Eck and L.-K. Soh, "Evaluating pomdp rewards for active perception," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*. International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 1221–1222.

[9] H. Soh and Y. Demiris, "Evolving policies for multi-reward partially observable markov decision processes (mr-pomdps)," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM, 2011, pp. 713–720.

[10] K. H. Wray and S. Zilberstein, "Multi-Objective POMDPs with Lexicographic Reward Preferences," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[11] D. M. Roijers, S. Whiteson, and F. A. Oliehoek, "Point-based planning for multi-objective pomdps," in *IJCAI 2015: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015, pp. 1666–1672.