Game-Theory based Multi-Robot Searching Approach

Yan Meng and Ke Cao
Department of Electrical and Computer Engineering
Stevens Institute of Technology, Hoboken, New Jersey, USA
yan.meng@stevens.edu, kcao@stevens.edu

Abstract — **Two strategic searching approaches for a multi-robot system are proposed in this paper: utility greedy approach and game theory approach. It is assumed that a-priori probability of the target distribution is provided in a partially known dynamic environment. The proposed two approaches aim to optimize the searching task using a dynamic-programming based utility function. The pure Nash Equilibrium solution and the mixed-strategy equilibrium solutions for game-theory based approach are provided. Extensive simulation results demonstrate that the proposed searching approaches have better searching performance and robustness compared to the other heuristic searching strategies.**

I. Introduction

In an Urban Search and Rescue (USAR) scenario, one of key objectives is to efficiently find the survivors, drop off some foods/water, and inform the First Responders the exact locations of the survivors. If the locations of the survivors are completely unknown, the searching problem becomes the area exploration, which usually leads to extravagant time waste by going through those areas that the survivors wouldn't stay.

To be more efficient for the searching, it is reasonable to assume that some partial information are available either through distributed sensors installed in the area, or based on some heuristics from human beings in the emergency situation. One natural way to capture the available information is to represent it as the likelihood of the survivor presence in the search space. In USAR scenarios, Murphy [1] proposed a biomimetic search strategy where a robot partitions the search space based on the semantic understanding of the expected distribution of survivors, then systematically searches each of the volumes in ranked order. In the case of finding targets in free space, such as boats lost at sea, Bourgault et al. [2] employed a Bayesian approach where the target probability density function (PDF) is used as prior information. As rescue air vehicles cover the ocean, the target PDF is updated using the model of the sensor and expected target motion. Optimal trajectories for the search are those that maximize the cumulative probability of detection over a limited time horizon. Lau et al. [3] proposed a search strategy that minimizes the expected average time for detection of each target, which deals with an environment where multiple targets may be present and allows travel and search costs of individual regions to be arbitrarily specified. The available information about the targets is expressed by the expected proportion of targets present in each region.

All of these strategies are applied only to a single individual robot. As we know, multi-robot systems are more desirable in some scenarios, such as underwater and space exploration, urbane search and rescue, and hazardous environments, due to the robustness, stability, adaptability, and scalability, which is difficult, if not impossible, provided by individual robots.

Some researches have been conducted on the multi-robot searching. In [4], an optimal searching strategy for a target on m concurrent rays in parallel using p robots was presented based on a solution of *cow path problem* that can be treated as a special condition with $p = 1$ and has been solved in [5][6]. The interaction between the robots is relatively simple in this case due to the special configurations. An alternative approach that proved to be more efficient consists of discretizing time and partitioning the continuous space into a finite collection of cells. The search problem is then reduced to deciding which cell to visit at each time interval. Eagle [7] noted that a discrete search can be formulated as an optimization on a partially observable Markov decision process (POMDP) and proposed a dynamic programming solution to it. However, since the optimization of POMDPs is extremely computationally extensive, this approach is often not practical. Instead, Eagle and Yee [8], and Stewart [9] formulated the discrete search as nonlinear integer programming problem and proposed branch-and-bound procedures to solve it, which in the case of [8] are optimal. Kok et. al. [10] described a framework to coordinate multiple robots using coordinate graphs (CG). First a discretization of the state by assigning roles to the agents was conducted. Then a CG-based method was applied to the derived set of roles. The capability of dynamic update of the topology of the CG and the role-based coordination rules made this approach to be practical for large scale teams.

Most multi-robot searching approaches assume that robots will maintain wireless (explicit) communication with each other during the searching. However in USAR environment, the wireless networks may attenuate or even unavailable due to the mobile connectivity topology or destroyed buildings. Furthermore, extensive communication between the robots also causes the extensive power consumption, which may lead to the unexpected halt of robot motion in a very short period. Therefore, by reducing the explicit communication between the robots, the overall system robustness can be improved. Meanwhile, some strategic approaches are necessary for the coordination.

For a multi-robot system, the mutual interactions between individual robots sharing a common workspace could be much more complex in general cases, the game theory seems to be a convenient tool for modeling and solving multi-robot interaction problems.

Skrzypczyk [11] proposed the architecture of a control system for a real-time collision-free movement coordination in a multi-robot environment, performing their navigational tasks by using the normal form games. In [12], an approach for analyzing and selecting time-optimal coordination strategies for n robots whose configurations were constrained to lie on C-space road map was proposed, where the maximal Nash Equilibrium concept was used to find the optimal strategies for each robot. A pursuit-evasion problem as a Markov game was described in [13], which was the generalization of a Markov decision process to the case when the system evolution is governed by a transition probability function depending on two or more player's actions. This probabilistic setting made it possible to model the uncertainty affecting the player's motion. Combining exploration and pursuit in a single problem was then translated into learning the transition probability function while playing the game. However, this approach required that the pursuit-evasion policies be learned for each new obstacle configuration. To reduce the complexity of large scale teams, the architecture model need to be dynamically simplified.

Very few works have been conducted in multi-robot target searching using the game theory due to the expensive computation time. In this paper, we investigate this problem in a partially known dynamic searching area. The searching area is partitioned into different regions, where the initial probability of the target distribution in each region is given. When the searching task starts, the probability for each region will be updated dynamically based on the new information obtained by the robots in different regions. A dynamic programming is applied to formalize the utility function for each robot. Based on this utility function, the decision making approach and a non-zero-sum game theory are proposed to coordinate a team of robots for the searching task.

To reduce the computation time of the proposed game strategic approach, instead of calculating all the recursive steps, one-step optimal approach is proposed, which turns out to be good enough for the searching task. In addition, the game will only be reinitiated upon new events rather than fixed-time iteration, which also decrease the computation complexity. Furthermore, the proposed algorithm can easily be extended to the dynamic environments in USAR.

Compared to other multi-robot searching algorithm, the proposed decision-making and game strategies algorithms can detect the targets within much less searching time than other heuristic algorithms. Furthermore, the proposed algorithm has much higher robustness and fault-tolerance compared to those multi-robot systems which heavily rely on extensive explicit communication for coordination, especially under USAR environment where the wireless network is tend to be unreliable.

II. PROBLEM STATEMENT

Assume there are N robots searching for a single target in an indoor environment with J different regions, as shown in Fig. 1. A prior map of the searching area is given as well as the initial probability of the target distribution for each region. The searching area is discretized and partitioned into a finite collection of cells. The robots can only move from one cell to the adjacent ones. Initially, N robots start from the entrance of the searching area. The team of robots can be homogeneous or heterogeneous in terms of their searching capabilities.

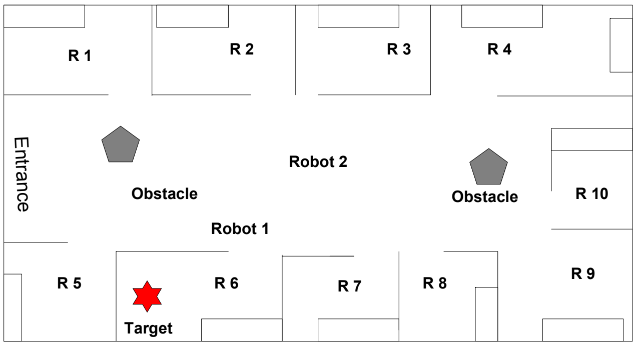

Fig. 1: The searching environment example

In USAR scenarios, buildings may collapse, facilities may scatter, and some of the obstacles may block the ways to enter each region. Therefore, the real world map may be different from the given map. Therefore, it is reasonable to expect that some unexpected obstacles are distributed randomly in the environment which can not be known in advance. When a robot detects an obstacle, the robot has to avoid the obstacle and plan the global path again dynamically. The searching task stops when the target is found.

The searching problem can be stated as follows: given a coarse map of the searching area and a priori knowledge of the target distribution, find an efficient and robust searching strategy for multiple robots so that the expected average searching time can be minimized.

III. SEARCHING STRATEGIES

A. Event-Triggered System Discretization

Once the searching task starts, the finite state machine of a robot consists of two states: busy and free. The robot state is defined as busy when it is searching inside a region. Otherwise, its state is defined as free. Initially all robots are set as free, and the robots' states are updated dynamically during the searching. A new event happens when a robot enters a region or finishes searching its current region, which can trigger the update of the robot state.

To decide when the next event will occur, a trigger is used to record the pair of times when each robot will change its state. The next event will happen at the closer time in the trigger, which is cumulated with the total searching time and is subtracted from the farther one. With this event-triggered discretization, the searching time is updated at each event until the end of searching. Since the robots only communicate with each other upon new event it can obviate the communication overhead significantly compared with the fixed-time-interval discretization method.

B. Utility Function Definition

Generally, the utility can be defined as the searching payoff value by selecting which region to search on the next discrete time. For a multi-robot system, to improve the collective searching efficiency, the utility value of each robot does not only depend on its own payoff value, but also on other robots' decisions.

Obviously, the utility associated with each robot depends on the probability of the target at each region. The higher the probability, the higher the utility value should be. When the searching task starts, the probabilities of all regions are updated dynamically based on the current searching results. For example, if one robot finishes searching in region 1 without detecting the target, then the initial probability of the target in region 1 is evenly distributed by all of unsearched regions on next discrete time. In other words, the probabilities of the target in the unsearched regions are increased, which can be expressed in the following equation.

$$p_i^{n+1} = 0;\ \ p_j^{n+1} = \frac{p_j^n}{1-p_i^n},\ j \neq i,\ j=1,\ 2,\ \dots,\ J. \quad (1)$$

where p_i represents the priori probability of the target in each region, J is the maximum region number and n is the current discrete time.

However, this priority-only based approach may try to achieve the most valuable goal (i.e. highest priority of target detection) irrespective of the difficulty of that goal. For example, the agent may skip the nearest region which has lower probability, and go for a very far region which has higher probability but may take much longer time to reach. To build more sensible agents, in this paper, we combine travel cost and utility calculations for each action and then calculate the expected utility.

The set of decisions made by the robot from 1 to N is denoted by $\mathbf{D} = [d_1, \dots, d_n, \dots, d_N]$. The set of probabilities of the target from region 1 to region J is denoted by $\mathbf{P} = [p_1, p_2, \dots, p_J]$, where p_i represents the priori probability of the target in each region, and i represents the region number. The travel cost can be divided into two time-based vectors, $\mathbf{T}_n$ and $\mathbf{T}_c$, where $\mathbf{T}_n = [T_{n1}, T_{n2} \dots, T_{nJ}]$ denotes the set of time that robot n takes to navigate from its current position to different regions from 1 to J. Each robot may take different time to cover different regions depending on the region size and the facility density inside each region. Therefore, it is necessary to put this factor into the utility value estimation. The set of time required for the robot to cover each region from 1 to J is denoted by $\mathbf{T}_c = [T_1, T_2 \dots, T_J]$ for a homogeneous multi-robot system, where T_j denotes the time required to cover the region j for each robot. For a heterogeneous system, the set of time for robot n to cover each region is denoted by $\mathbf{T}_{nc} = [T_{nc1}, T_{nc2} \dots, T_{ncJ}]$.

To simplify the concept description, it is assumed that the system is homogeneous in this paper although it can be easily extended to the heterogeneous systems. The utility function of the robot n can be defined as a function of $(\mathbf{D}, \mathbf{P}, \mathbf{T}_n, \mathbf{T}_c)$.

To obtain the optimal solution, a Dynamic Programming Equation (DPE) is applied to define the utility function for robot n as follows.

$$U_n(d_1, d_2, \dots, d_N) = f_n(\mathbf{D}, \mathbf{P}, \mathbf{T}_n, \mathbf{T}_c) = \begin{cases} 0, & \text{if } p_{d_n} = 0 \\ g(\mathbf{D}, p_{d_n}, T_{nd_n}, T_{d_n}) & \text{if } p_{d_n} = 1 \\ h(\mathbf{D})[g(\mathbf{D}, p_{d_n}, T_{nd_n}, T_{d_n}) + (1-p_{d_n}) \max_{\hat{d}_n} \{ f_n(\hat{\mathbf{D}}, \hat{\mathbf{P}}, \hat{\mathbf{T}}_n, \mathbf{T}_c) \}] & \textit{otherwise} \end{cases} \quad (2)$$

where U_n represents the utility function of robot n, $d_1, d_2, \dots, d_N$ represent the decisions made by robot 1 to robot N. p_{d_n} represents the probability of target detection in region d_n by robot n. $g(\mathbf{D}, p_{d_n}, \mathbf{T}_{nd_n}, \mathbf{T}_{d_n})$ represents the payoff gain of robot n searching the region d_n, which is defined as follows:

$$g(\mathbf{D}, p_{d_n}, \mathbf{T}_{nd_n}, \mathbf{T}_{d_n}) = \frac{p_{d_n}}{k_1 T_{nd_n} + k_2 T_{d_n}}, \quad (3)$$

where T_{nd_n} and T_{d_n} represent the time required for robot n to navigate from its current position to region d_n, and the time required for a robot to cover region d_n, respectively. k_1 and k_2 are scale factors which can be adjusted based on different environmental structures. For example, the robot may move faster in the empty corridors and move slower in the rooms with higher density of obstacles. In this case, k_1 is set greater than k_2.

$(1-p_{d_n}) \max_{\hat{d}_n} \{ f_n(\hat{\mathbf{D}}, \hat{\mathbf{P}}, \hat{\mathbf{T}}_n, \mathbf{T}_c) \}$ represents the maximum expected utility of robot n by selecting different $\hat{d}_n$ for the rest of the unsearched regions after finishing the region d_n with the assumption that other robots keep their current decision during this recursive procedure, where

$$\hat{\mathbf{D}} = [d_1, \dots, \hat{d}_n, \dots, d_N]$$

$$\hat{\mathbf{P}} = \left[\frac{p_1}{1-p_n}, \dots \frac{p_{n-1}}{1-p_n}, 0, \frac{p_{n+1}}{1-p_n} \dots \frac{p_J}{1-p_n} \right],$$

$$\hat{\mathbf{T}}_n = \left\lfloor \hat{T}_{n1}, \hat{T}_{n2} .., T_{nn} ., \hat{T}_{nJ} \right\rfloor, \quad (4)$$

where the variables with hat sign represent the estimated value after the robot *n* finishes its current searching step.

The dynamic programming, however, is somewhat intractable for large-scale region numbers. It is possible to approximate the dynamic programming without iterating for all possible states. To reduce the computational time, one-step dynamic programming solution is applied in Equation (2). The average expected teammate contribution is computed as the contribution that the teammate would make from its current pose. This approximation is reasonable when each step is relatively small.

In general, the utility is zero if the probability of target detection in region d_n is zero. If the probability of the target detection in region d_n is 1, which means this is the last room need to be searched. In this case, the utility function is only related to the payoff value by searching region d_n. Otherwise, the utility is a recursive function defined in (2).

Considering the situation that several robots may choose the same region simultaneously based on their own utility functions, which may decrease the overall searching performance. We define a factor $h(\mathbf{D})$ as follows:

$$h(\mathbf{D}) = \begin{cases} \dfrac{T_{ni}}{T_{ni} + \sum T_{mi}}, & if \quad d_m = d_n, m \neq n, m = 1,2,...,N. \\ 1 & otherwise \end{cases}, \quad (5)$$

where T_{ni} is the travel time for robot *n* from its current position to the selected region *i,* ΣT_{mi} is the total travel time for robot *m* (*m* can be multiple robots from 1 to *N*, except *n*) from their current positions to region *i*.

The definition of $h(\mathbf{D})$ actually embeds the coordination between the multiple robots. In other words, by cutting down the utility value, it helps to prevent multiple robots picking up the same region simultaneously, which will eventually improve the overall searching efficiency.

The coefficients $k_1, k_2,$ and k_3 are gains to adjust the influence of individual factors of the utility function. Various kinds of searching behaviors can be achieved by tuning these parameters.

C. Utility Greedy Strategy

Based on the utility function defined in the previous section, an intuitive approach for the searching task is utility greedy strategy, where each robot chooses the region with the highest utility value based on Equation (2). If more than one region has the same highest utility values, the robot randomly picks one from them.

D. Game-Theory based Searching Strategy

The multiple robots searching for a target is modeled as a multi-player cooperative non-zero-sum game since the coordination is embedded into the utility function through (5). The players choose their strategies simultaneously at the beginning of the game. Although the overall process of the searching is dynamic, we can treat it as a sequence of static game at each discrete time. At each discrete time moment, the players must solve a static game that is nonzero-sum because the probability in question is conditioned to the distinct observations that the corresponding team is available at that time.

The probabilities of all the regions are updated whenever a region has been searched. With the updated probabilities, all the free robots recalculate their utilities. The game stops when the target is found.

According to the current positions of robots and current probability of the target in each region, the utility matrix can be calculated. Based on the calculated utility matrix, the Nash Equilibrium (NE) is applied for this nonzero-sum game. Playing at a NE ensures a minimum performance level to each team. On the other hand, no player can gain from a unilateral deviation with respect to the NE policy.

It is easy for robots to make decision if only one NE point is available. However, when the room number becomes larger, there may exist more than one NE point. In this case, a max-min method is applied to calculate the mixed-strategy equilibrium. Let $p_n(j)$ denotes the probability of robot *n* choosing region *j*, and $\mathbf{P}_n = [p_n(1) \quad p_n(2) \quad \dots \quad p_n(J)]^T$, we have

$$\sum_{j=1}^{J} p_n(j) = 1, \qquad n = 1,2,...,N$$

The pseudo code for game-theory based strategic algorithm is described as followings:

```
WHILE (target is not found)
    IF (all robots are free or at least one robot is free)
        Calculate the utility values of each robot using equation (2);
        Build the utility matrix U_n and find the maximum point of utilities in the matrix for each robot;
        IF (the max point for all the robots is located at the same point)
            Pure NE point. Take this NE point as the next decision;
        ELSE
            Find the mixed strategy using the max-min method to estimate P_n by solving the equation U_n × P_n = [0 0 0 1]^T with linear programming algorithm.
        END;
    END;
    Estimate the time when the next trigger event happens based on the time traveled to new destination and the time searching the current region;
    Move the robots to the destination region;
    Update the searching status;
END;
```

Utility matrix $\mathbf{U}_n$ is a *N*-dimensional matrix for *N* robots, where there are J (region number) units at each dimension, and each cell of the matrix consists of *N* utility values for each robot at the corresponding position. Since the game is a finite strategic-form game, the existence of the mixed-strategy equilibrium can be guaranteed. Generally speaking, how to choose the proper solution from the multiple equilibria is a complex problem [14]. The criteria are that the solution that provides the maximum payoffs for all the players and "fair" distribution of cost among the players should be chosen.

For a multi-agent system, the overall outcome depends critically on the choices made by each self-interested agent. Each agent simply computes the best utility outcome and knows that the other agents they are working with will do the same. Since agents can use game theory to predict what others will do, which obviates the need for explicit communication – coordination arises because of the assumption of mutual rationality.

V. Simulation Results

To evaluate the performance of the proposed game strategic searching approaches, the simulations with two robots using MATLAB have been conducted. The searching area is sketched as a square of 100 x 100 cells with multiple regions distributed. It is assumed that each robot takes 1 time unit to traverse one cell in the simulation. The searching time is calculated by the time units from the starting time till the target is found.

A. Two Other Heuristic Searching Strategies

In order to evaluate the searching performance of the proposed two *utility greedy (UG)* and *game theory (GT)* approaches, comparison with other heuristic searching strategies is necessary. Two such strategies are proposed. First one is called *randomly selection (RS)* approach, where each robot randomly selects the next region to search from the list of unsearched regions without taking into account of prior probability of the target distribution. Second strategy is called *probability-based (PB)* approach, where each robot only picks the region with the highest probability at any discrete time as its next objective region. If more than one region has the same highest probability, the robot randomly picks one from them.

B. Simulation Results

Due to the asynchronous actions of two robots during the searching, it is necessary to predict when the next event may occur based on the time for a robot moving to the destination region from its current position if the robot is free or the time to search the destination region if the robot is busy.

Four approaches, random selection, probability-based, utility greedy, and game theory approaches are conducted in the searching area. To evaluate the searching performance in a scalable environment, different configurations are applied, where each configuration has different number of regions. 1000 runs for each approach are conducted and the target is distributed according to the prior probabilities of different regions.

The simulation results of mean value and the corresponding variance of searching times are shown in Fig. 2. It is obviously that the searching time of the utility greedy and game theory approaches are much less than that of the probability-based approach since the travel and searching time was ignored in the latter case. The random selection has the worst performance since neither the prior target distribution information nor the travel/searching time is considered.

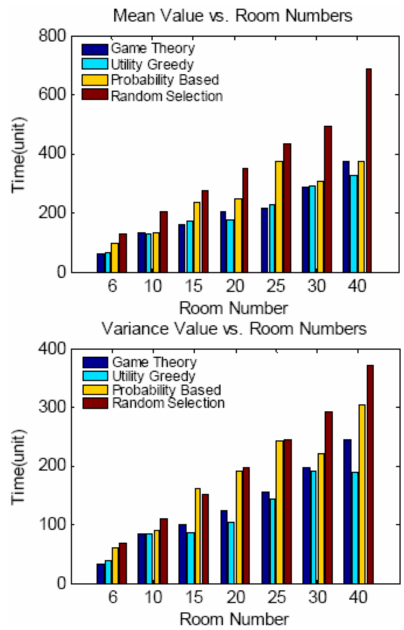

Fig. 2: Searching time (mean and variance) with different configurations

It can be seen that the performance of the game theory approach is competitive with that using the utility greedy approach. This scenario mainly depends on the procedure design of the game theory in the simulation. For this two-robot simulation, there are three scenarios of robot states exist at any discrete time: (1) both are free; (2) one is free and one is busy; and (3) both are busy. Among these three cases, the robots only need to make decisions on the first two cases.

When both robots are free, the utility matrix is calculated and their searching decisions are computed based on the game strategy. However, if one robot is busy and the other one is free, the free robot will make its decision only based on the utility value instead of starting a new game. The motivation for this simplified procedure is to reduce the computational time. If a new game is restarted whenever one robot is free, which may happen very frequently, especially for a large scale of robots, the computational time would be increased significantly since the game strategy takes much longer time to calculate than the utility greedy approach. In addition, since the other robot is busy in searching a region, it would make more sense to let it finish its current searching instead of reselecting the searching region again due to the new status. If the second case happens very often, then the overall performance of game strategy tends to close to that of the utility greedy. Sometimes the game strategy gets even worse performance than utility greedy especially in the case with more regions.

In the real world, it is difficult to obtain the accurate prior probability of the target distribution especially in USAR situations where the variance may be very large. To explore the system robustness with respect to the prior probability variations, another set of simulations are implemented. 1000 runs are conducted for each searching strategy, where the target is distributed in the searching regions with the variations of the priori probability from 10% to 50%. The simulation results in the configuration of 10-room are shown in Fig. 3. As can be seen, the probability-based approach is very sensitive to the probability variation since the probability is the only criteria for the robot to make searching decisions. The probability variation has no effect on the random selection approach at all, which is easily to be understood. The game theory and utility greedy approach are much more robust than the probability-based approach, where the game theory beats utility greedy approach in both mean searching time and variance. This indicates that the game theory is more robust to handle environmental uncertainty compared to utility greedy approach, although we have to pay the penalty of more computational time for game strategy.

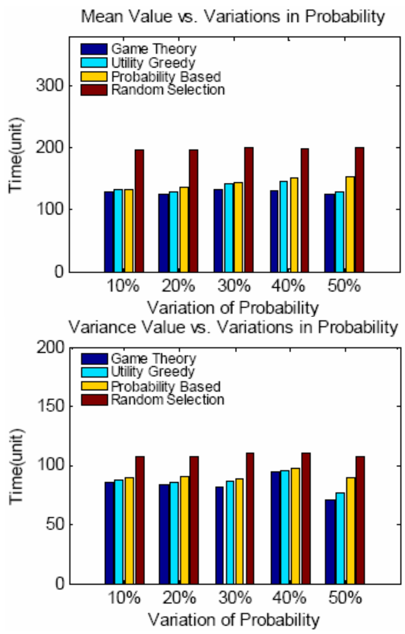

Fig. 3: Searching time (mean and variance) with different variations in probability of target distribution with a 10-room configuration

VI. Conclusion and Discussion

Utility greedy and game-theory based strategic searching approaches are proposed in this paper for a cooperative multi-robot searching task. Comparing to other heuristic searching strategies, such as random selection and probability-based approaches, the simulation results demonstrated that the proposed two approaches have much better performance. Comparing to the competitive utility greedy algorithm, the game theory has guaranteed better worst-case performance and be more robust to handle the environmental uncertainty. Another advantage of using game theory based approach is that the explicit communication between the robots can be obviated significantly due to their mutual rationality. Therefore, it can be applied to some emergency scenarios where the RF communication tends to attenuate or even broken.

Our preliminary simulation only contains two homogeneous robots and one target in the searching task. The proposed algorithm can easily be extended to the heterogeneous robots with different moving speeds and local sensing capabilities by setting up different travel and covering time for each robot. For the case of multiple targets, if the priori probability for each target is provided, the proposed algorithms can be applied by adding one more level of cooperative decision of which target to pick, which is our next research project.

As is known, the computation time would become intractable with large scale robot team using the game-theory based approach. Therefore our future research is to extend this game-theory based approach to the more realistic USAR regions with multi targets by large scale heterogeneous robots by simplifying the system model to cut down the computation time.

.

References

[1] R. R. Murphy, "Biomimetic search for urbane search and rescue", *IEEE/RSJ International Conference on Intelligent Robots and Systems,* vol. 3, pp. 2073-2078, 2000.

[2] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte, "Coordinated decentralized search for a lost target in a bayesian world", *IEEE/RSJ International Conference on Intelligent Robots and Systems,* 2003.

[3] H. Lau, S. Huang and G. Dissanayake, "Optimal search for multiple targets in a built environment", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05)*, Edmonton, Alberta, Canada, August 2005.

[4] A. L′opez-Ortiz1 and S. Schuierer2, "Online parallel heuristics and robot searching under the competitive framework", *8th Scandinavian Workshop on Algorithm Theory,* Turku, Finland, July 3-5, 2002.

[5] M.-Y. Kao, J. H. Reif, and S. R. Tate, "Searching in an unknown environment: An optimal randomized algorithm for the cow-path problem", In *Proc. 4th ACM-SIAM Sympos. Discrete Algorithms*, pages 441–447, 1993.

[6] R. Baeza-Yates, J. Culberson, and G. Rawlins. "Searching in the plane", *Informationand Computation*, 106:234–252, 1993.

[7] J. N. Eagle, "Optimal search for a moving target when the search path is constrained", *Operations Research*, 32(5): 1107-1115, 1984.

[8] J. N. Eagle and J. R. Yee, "An optimal branch and bound procedure dor the constrained path, moving target search problem". *Operations Research*, 38(1):110-114, 1990.

[9] T. J. Stewart, "Searching for a moving target when searcher motion is restricted". *Computers and Operations Research*, 6:129-140, 1979.

[10] J. R. Kok, M. T. J. Spaan, and Nikos Vlassis, "Multi-robot decision making using coordination graphs", In *Proceedings of the International Conference on Advanced Robotics (ICAR)*, pp. 1124–1129, Coimbra, Portugal, June 2003.

[11] K. Skrzypczyk, "Game theory based target following by a team of robots", *Fourth International Workshop on Robot Motion and Control,* June 17-20, 2004.

[12] S. LaValle, S. Hutchinson, "Path selection and coordination for multiple robots via Nash equilibria", Proc. *IEEE Int. Conf. Robot and Automation*, 1994, pp. 1847-1852.

[13] J. P. Hespanha, M. Prandini, and S. Sastry, "Probabilistic pursuit-evasion games: a one-step Nash approach. *In proc. of the 39th conf. on decision and control*, vol. 3, Dec. 2000.

[14] J. C. Harsanyi and R. Selten, "A general theory of equilibrium selection in games", the MIT Press, Cambridge, Massachusetts, 1998.