Emergent Task Allocation for Mobile Robots

Nuzhet Atay
Department of Computer Science and Engineering
Washington University in St. Louis
Email: atay@cse.wustl.edu

Burchan Bayazit
Department of Computer Science and Engineering
Washington University in St. Louis
Email: bayazit@cse.wustl.edu

***Abstract*— Multi-robot systems require efficient and accurate planning in order to perform mission-critical tasks. However, algorithms that find the optimal solution are usually computationally expensive and may require a large number of messages between the robots as the robots need to be aware of the global spatiotemporal information. In this paper, we introduce an emergent task allocation approach for mobile robots. Each robot uses only the information obtained from its immediate neighbors in its decision. Our technique is general enough to be applicable to any task allocation scheme as long as a utilization criteria is given. We demonstrate that our approach performs similar to the integer linear programming technique which finds the global optimal solution at the fraction of its cost. The tasks we are interested in are detecting and controlling multiple regions of interest in an unknown environment in the presence of obstacles and intrinsic constraints. The objective function contains four basic requirements of a multi-robot system serving this purpose:** *control regions of interest*, *provide communication between robots*, *control maximum area* **and** *detect regions of interest*. **Our solution determines optimal locations of the robots to maximize the objective function for small problem instances while efficiently satisfying some constraints such as avoiding obstacles and staying within the speed capabilities of the robots, and finds an approximation to global optimal solution by correlating solutions of small problems.**

I. Introduction

Several real life scenarios, such as fire fighting, search and rescue, surveillance, etc., need multiple mobile robot coordination and task allocation. Such scenarios generally include distinct regions of interest that require the attention of some robots. If the locations of these regions are not known, the mobile robots need to explore the environment to find them. In this paper, we propose a solution to the problem of detecting and controlling multiple regions of interest in an unknown environment using multiple mobile robots. In our system, we assume a bounded environment that is to be controlled by a group of heterogeneous robots. In this environment, there are regions of interest which need to be tracked. These regions are dynamic, i.e. they can appear at any point, anytime and can move, spread or disappear. Each region may require more than one robot to track and control. Robots do not have initial information about the environment, and the environment is only partially-observable by the robots. Each robot has wireless communication capability, but its range is not uniform. Two robots can communicate between each other only if both of them are in the communication range of each other. They can have different speed limits and are equipped with the sensors to identify the obstacles and the regions of interest if they are within robots' sensing range. Sensor ranges on these robots are not necessarily uniform. The environment can have static or dynamic obstacles, and the robots need to avoid them in order to perform their tasks.

We propose an emergent solution to the task allocation problem for heterogeneous robots. The tasks we are interested in are: (i) *covering all regions of interest*, (ii) *providing communication between as many robots as possible*, (iii) *controlling maximum total surface by all the robots*, (iv) *exploring new regions*. Our objective is to maximize these items while satisfying the constraints such as avoiding the obstacles or moving within the speed capabilities of individual robots. Additional constraints we are considering are the communication between two robots (which exists only if either two robots are in the communication range of each other or there is a route between them through other robots satisfying the communication constraints), and, the sensing of the obstacles and regions of interest when they are within the robots' sensor range. Our approach is general enough to be easily adapted to additional constraints and objectives, making it customizable for various mobile robot problems.

Several linear programming based solutions have been proposed for mobile robot task allocation problem. Although these proposals are generally successful in finding the optimal solution, they usually require collecting information about all robots and regions of interest, and processing this information at a central location. As a result, these approaches can be infeasible in terms of the computation time and communication cost for large groups. In order to provide scalability and efficiency, we are proposing an emergent approach. In this approach, each robot solves a partial problem based on its observations, then exchanges information (such as intentions and directives) with the robots in the communication range to maintain coordination. The system is fully distributed which allows this technique to be applied to any number of robots with computation and communication cost limited by constant parameters which can be defined according to the application requirements. We experimentally show that this approach gives results comparable to global optimal solution, and performs hundreds of times faster with little communication cost.

Since we use mixed integer linear programming for the solution of the partial problems, our contributions also include a customizable multi-robot task allocation solver which can be used to find global optimal solution under the given constraints. In contrast to other linear programming solutions, we also present an efficient way to check obstacle collisions.

While we are concentrated on the mobile robots, our solution is applicable to other distributed task allocation problem as long as a function to evaluate the goodness of the solution is defined. The technical report version of this paper that includes the details of the mixed integer linear programming solution with the description of constraints and variables, as well as some proofs including the convergence of our approach to the global solution, extensions that show the flexibility of the approach, and a larger set of experiments on different environments can be found at [1].

The rest of the paper is organized as follows. The next section gives a summary of the related research and brief comparison to our approach when it is applicable. Section III gives the problem definition. Section IV describes our mixed integer linear programming solution, and Section V explains the emergent behavior task allocation approach. We present simulation results in Section VI and Section VII concludes our paper.

II. Related Work

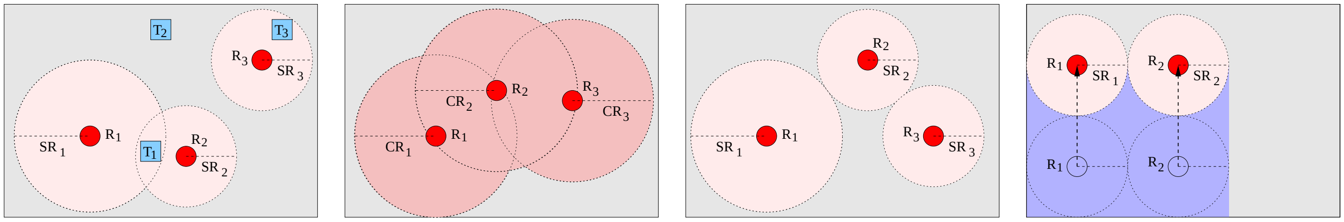
Multi-robot task allocation has been studied extensively because of the importance of application areas. One quite popular approach to this problem is utilizing negotiation or auction based mechanisms. In this approach, each distributed agent computes a cost for completing a task, and broadcasts the bid for that task. Auctioneer agent decides the best available bid, and winning bidder attempts to perform this task. Following the contract-net protocol [2], several variations of this method has been proposed [3]–[7]. Another important approach is using behavior based architecture. ALLIANCE [8] is a behavior-based architecture where robots use motivational behaviors such as robot impatience and robot acquiescence. These behaviors motivate robots to perform tasks that cannot be done by other robots, and give up the tasks they cannot perform efficiently. BLE [9] is another behavior-based architecture which uses continuous monitoring of tasks among robots and best fit robot is assigned to each task. A detailed analysis and comparison of these methods can be found at [10], [11]. These methods propose distributed algorithms where resource allocation is an approximation to the global optimum. The main difference between these methods and our approach is that we are using a formulation that can provide global optimum solution when information propagation is not limited. However, instead of finding the global optimal solution using all the information which has high computation and communication cost, we distribute computation and information processing among robots and reach an approximation to the global optimal solution through iteration.

Task allocation problem is also studied in the context of cooperation of Unmanned Aerial Vehicles (UAVs). Several methods are proposed for search and attack missions of UAVs [12]–[20]. Our method is similar to the methods proposed in [13], [14], [17], [20], since these methods are also using mixed-integer linear programming task allocation. However, in these papers, the problem is defined as minimizing mission completion time while UAVs visiting predetermined waypoints and avoiding no-fly zones. The solution to this problem is formulated as finding all possible combinations of task allocations, and choosing the best combination. This definition of task allocation is actually quite different than our problem definition. Our aim is to explore environment, find regions of interest, and assign tasks optimally obeying the constraints imposed at that moment. In other words, we are finding a solution in real-time, instead of finding an initial plan and executing it.

III. Problem Definition

In our problem definition, there are regions of interest we want robots to explore and cover. In the rest of the paper, we will call these regions “targets”. Since larger areas can be represented with multiple points, without loss of generality, we assume targets are represented as points in planar space. A target is assumed to be covered if there are enough robots that have the target in their sensing range. The number of robots required to cover a target varies for each target. We assume the future locations of known targets after a time period can be predicted. Our primary purpose is to find locations of robots in order to cover as many targets as possible using the estimated locations of targets. While covering all the targets, it is also desirable to provide communication between as many robots as possible because this will allow robots to exchange the information about the environment and the targets. In a centralized approach, this also leads to a better solution since the solver will be aware of more information. It is also preferable that robots need to cover as much area as possible in addition to covering targets to increase the chances of detecting other undiscovered targets. Similarly, in order to discover new targets and avoid waiting at the same location when no targets are being tracked, the robots are expected to explore new regions.

We define the state of the system as current locations of targets, number of robots needed to cover a target, current positions of the robots, positions of the obstacles, previously explored regions, and each robot’s speed, communication range and sensing range. The output of our algorithm is the optimal locations of the robots for the next state of the system after a brief period of time. Please note that, we assume we can predict the location of the targets at the next step. There are approaches for motion prediction that can be used for this purpose [21]. We also assume that there are no sensor or odometry errors, however, implementation of our method on real robots can introduce these errors. The method we are planning to utilize for handling noisy measurements, sensor errors and mechanical errors like slippage or odometry errors takes advantage of communication among nearby robots. We believe our approach promotes robots to stay in the contact as much as possible and make it possible to share as much sensor information as possible.



(a) (b) (c) (d)

Fig. 1. SR stands for sensing range, and CR stands for communication range (a) A target is covered when it is in sensing range of some robots, where number of robots is determined according to the requirements of the target. Robots R_1 and R_2 cover T_1, while R_3 covers T_3. T_2 is not covered. (b) Two robots can communicate if both robots are in communication range of each other. R_2 can communicate with R_1 and R_3, and works as a hub between R_1 and R_3 which cannot communicate directly. (c) Maximum area coverage is obtained if sensing range of robots do not overlap. In the figure, sensing regions of robots barely touch each other (d) Robots mark regions they explored before, and move towards unexplored regions. R_1 and R_2 move upward toward unexplored region after marking dark (blue) region as explored

IV. Mixed Integer Linear Programming for Task Allocation

Although our main contribution is the emergent task allocation, we first would like to show how a centralized approach can be utilized to find the optimal placement of robots after a defined time period. In the next section, we will show how individual robots can use the same approach to solve their partial problems to achieve emergent task allocation.

Our centralized approach utilizes a mixed integer linear program. Either a designated robot runs the solver or each robot in a group executes the same solver with the same data to find its placement. A group consists of the robots that are in the communication range of each other, hence states and observations of all the robots are known to the solver(s). If there are multiple groups of robots that cannot communicate with each other, each group will have its own task allocation based on its world view. If two groups merge, they can share their knowledge. The program runs periodically to find the best placements for each robot. It also runs if a new event happens, such as the discovery of an obstacle or a target. The linear program should satisfy some constraints: (i) an evaluated location is not acceptable if the robot cannot reach there either because of its speed limits or because of an obstacle, (ii) two robots cannot communicate if one of them is outside the communication range of the other, (iii) an obstacle or target is detectable only if it is within the sensing range of the robot. Our goal is then to maximize the number of targets tracked, the number of robots that can communicate with each other, the area of the environment covered by the robot sensors, and the area of the environment that was explored. In the next subsections, we will first discuss different objective functions and constraints, then we will show our overall optimization criterion and we will discuss the complexity. We give only the overview of the linear program because of space limitations, but detailed formulations and explanations can be found in the technical report version [1].

A. Obstacle Avoidance

In our system, we assume there are only rectangular shaped obstacles for the sake of simplicity of defining linear equations. However, more general shaped obstacles can be represented as rectangular meshes. When considering obstacles, we are not finding a path to avoid them, but we are finding whether or not it is possible to avoid them with the robot speed and timestep as the constraints. As it is mentioned before, output of the linear program is the final positions of the robots. When computing these positions, we utilize Manhattan paths to identify if there is a way for a robot to avoid an obstacle. As long as there is a Manhattan path that bypasses the obstacle and has a length that is possible for the robot to traverse under the given speed constraints, we consider the final position of the robot as a feasible configuration. Otherwise, that configuration is eliminated. Once a position is selected, more advanced navigation algorithms can be utilized to find more efficient paths. The alternative approach, i.e., finding exact path, requires finding intermediate states of the system at a fine resolution which increases complexity drastically. Please note that we are not aware of any other linear programming approach that addresses navigation problem.

B. Target Coverage

A target can be considered covered only if the number of robots following it is greater than or equal to its coverage requirement.[1] A robot can sense and control a target only if its sensing range is greater than or equal to the distance between itself and the target. A sample organization of the robots and targets is shown in Fig. 1(a). R_1 and R_2 are covering target T_1 and R_3 is covering T_3 while T_2 is not covered by any of the robots.

C. Communication

Each robot has a communication range. A robot can have a duplex communication link to another robot only if each robot is in the communication range of the other one. However, robots can communicate between each other with the help of other robots. So, if two robots cannot directly communicate with each other, but they share a common robot both of which can communicate, we assume that they can communicate. In other words, transitive links are allowed in the system. It

[1] Please see the technical report [1] for the proof that our optimization criterion results in continuous target coverage of all targets, if this optimization has highest priority.

should be noted that this condition implies communication between robots with the help of multiple intermediate robots, i.e. one or more robots can participate in a transitive link between two robots. A communication pattern of the robots is shown in Fig. 1(b). R_2 can communicate with both R_1 and R_3. R_1 and R_3 do not have a direct communication link, but they can communicate with the help of R_2.

D. Area Coverage

Robots have limited and constant sensing range, so the only way to maximize area coverage is by preventing the overlap of sensing ranges of robots. An ideal area coverage for the robots is represented in Fig. 1(c), where robots have no overlapping sensing range.

E. Exploration

In order to explore the environment, robots need to know places they have visited recently. We store this information as rectangular regions defining explored areas. Then the linear program tries to move robots into unexplored regions by checking the final position of the robots. So, the program gives a final position not located in an explored region. [2] A sample exploration scenario is shown in Fig. 1(d). Dark (blue) region is explored in the first step, so robots try to locate themselves outside of the explored area.

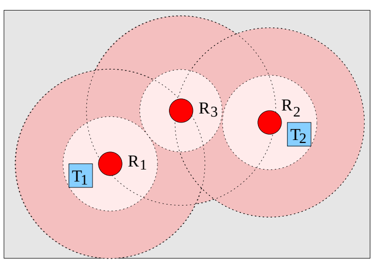


Fig. 2. An example distribution of robots providing optimum target coverage, communication and area coverage. Robot R_1 covers target T_1 and R_2 covers target T_2. R_3 is located to provide communication between them, and its sensing range does not overlap with others. Dark colored circles represent communication range, light colored circles represent sensing range.

F. Optimization Criterion

Optimization criterion consists of four components, *target coverage, communication between robots, area covered by the robots* and *the number of robots located in unexplored regions.*

Target Coverage: We utilize the number of targets that are covered, i.e.,

$$T = \sum_{j=1}^{n} coverage_j \tag{1}$$

where n=number of targets, $coverage_j$ is 1 when the number of robots that are covering $target_j$ is greater than or equal to the minimum requirement for that target, 0 otherwise.

[2]Please see the technical report [1] for the proof that given sufficient number of robots for communication and target tracking, our algorithm will result in the exploration of the all environment.

Communication: We utilize the number of pairs of robots that can communicate with each other, i.e.,

$$C = \sum_{i=1}^{n}\sum_{j=1}^{n} communication_{ij} \tag{2}$$

where n=number of robots, $communication_{ij}$ is 1 when robots i and j are within their communication range or they can communicate with the help of other robots, 0 otherwise.

Area Coverage: We utilize the number of pairs of robots whose sensor ranges do not intersect, i.e.,

$$A = \sum_{i=1}^{n}\sum_{j=1}^{n} area_{ij} \tag{3}$$

where n=number of robots, $area_{ij}$ is 1 when robots i and j cover non-overlapping regions, 0 otherwise.

Exploration: We utilize the number of robots in unexplored regions, i.e.,

$$E = \sum_{i=1}^{n}\sum_{j=1}^{m} exploration_{ij} \tag{4}$$

where n=number of robots, m=number of explored regions, $exploration_{ij}$ is 1 if the robot i is not in the explored region j, 0 otherwise.

Optimization Criterion: Our objective function is weighted sum of the above components.

$$maximize\ \alpha T + \beta C + \gamma A + \delta E \tag{5}$$

where α, β, γ, and δ are constants defining priorities.

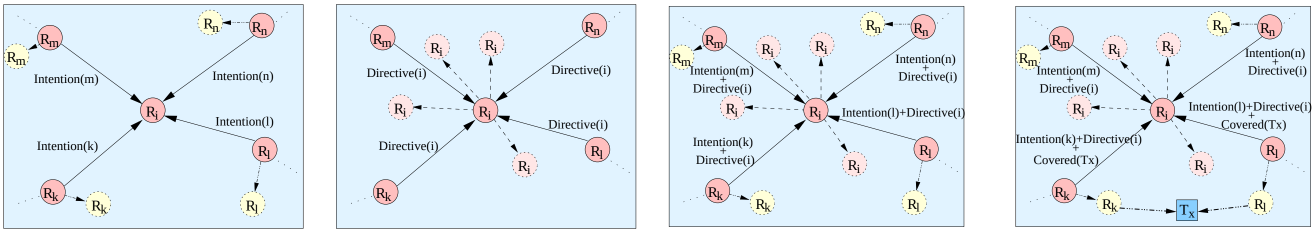
Figure 2 represents an optimal distribution of robots according to this optimization criterion. Robots arrange themselves so that they cover all targets, provide communication between each other, and cover as much area as possible.

G. Complexity

Our formulation results in a mixed-integer linear program, which is NP-Hard in the number of binary variables, so complexity of our program is dominated by the number of binary variables. Definitions and properties of binary variables can be found at the technical report [1]. For a problem with n targets, m robots, p obstacles and q explored regions, there are $n + nm + 2nn + 5mq + 4mp$ binary variables. So, the complexity can be stated as $O(n + nm + n^2 + mq + mp)$.

V. Emergent Task Allocation

As we have mentioned in the previous section, finding the optimal solution is an NP-Hard problem. While it may be possible to solve simple problems with on-board processors, finding solution for larger networks is very expensive even for a more powerful central server (because of both the cost of computation and the number of messages). In order to overcome this problem, we propose a distributed approach where each robot in the network finds a local solution based on the information from the vicinity of the robot. This approach utilizes the mixed integer linear program we described in Section IV. The local vicinity of the robot contains the region covered by



(a) (b) (c) (d)

Fig. 3. Information exchange for emergent task allocation: (a) Intentions, (b) Directives, (c) Intentions and Directives, (d) Intentions, Directives and Target Assignment. The dashed-circles connected to the neighboring robots $R_{k,l,m,n}$ represent their intentions, the dashed-circles connected to the R_i represent the directives to that robot by its neighbors.

the robot and its first-degree neighbors (1-hop away). Each robot uses the information about targets and obstacles that can be sensed by the robot itself and 1-hop neighbor robots in its computation. In order to increase efficiency, we further restrict the vicinity to k-closest robots if the number of 1-hop neighbors is large. While this segmentation of the problem makes the individual problems solvable by the mobile robots, each robot is concentrated on its own problem which is usually different than those of neighboring robots. As a result, its solution may be different from another robot's solution. In order to provide coordination between the neighboring robots, the robots exchange information among the neighbors (mainly contains intentions and/or directives) and update their local solutions based on this information. This exchange makes the solution of emergent task allocation comparable to that of centralized approach. Algorithm 1 summarizes this approach.

Algorithm 1 Coordination (robot i)

1: Find a solution with local information
2: **for all** k-closest 1-hop neighbor j **do**
3:     Send solution information to j
4:     Receive solution information from j
5: **end for**
6: Update solution according to new information
7: return $position(i)$

Although it is possible to iterate through lines $2-6$ several times, i.e., continuously updating the solution until it converges, we are interested in only a single exchange for efficiency purposes. In the technical report [1], we show that as the number of iterations increases, the solution converges to the global optimum. Similarly, if there is sufficient computational power on individual robots, the size of neighborhood can be increased to include the robots that are more hops away for obtaining better solution.

The information exchange between the robots could range from single position information which may require a single message between the robots to all the state information which may require multiple messages. We have selected the following methods for the information exchange:

A. Intentions

In the most simple approach, after finding a position that maximizes its utility (based on the current sensor information and neighbor information), each robot sends this location to its neighbors as its intended location. When a robot gets intentions from all neighbors, it assumes that these locations are final, and computes its own location that would maximize the utility. Note that, we still use the algorithm of Section IV, however, other robots' positions now become constraints of the system. Figure 3(a) represents this approach for robot i.

B. Directives

In the second approach, each robot computes a location for its neighbor, and sends this location to the neighbor as a directive. When a robot gets location information from all neighbors, it uses the list of locations as the potential locations, and finds the one that gives the highest value of the objective function using the linear program. The information transferred for robot i is shown in Figure 3(b).

C. Intentions and Directives

In the third approach, each robot computes optimal locations of itself and its neighbors, and sends these locations to the neighbors. When a robot gets these locations, for each potential location given by the neighbors, it evaluates the utility of that directive based on the intended locations of all neighbors. The directive that gives the highest value of the objective function is selected as the next location for that robot. This is represented in Figure 3(c) for robot i.

D. Intentions, Directives and Target Assignment Information

The last approach is similar to the third approach, but in addition to the information about locations, target assignment information is also sent to the neighbors. Target assignment states whether or not a robot is assigned to cover a target. This information can be used in different ways, but we use this so that no two robots try to cover the same target, unless that target needs to be covered by more than one robot. This approach provides better exploration and better area coverage, as robots can ignore a target and spread out when the target is covered by another robot. Figure 3(d) represents this approach for robot i.

E. Comparison to Centralized Global Optimization

Global optimization through centralized computation requires all information about the environment to be collected at one location. Assuming the central server is physically located in the center of the network and average hop count from other robots to the central server is p, average message count in the system for one planning phase is $O(p * n)$, where n is the number of robots. On the other hand, number of messages at the emergent approach is k for each robot, where k is the maximum number of neighbors that a robot can have. Total number of messages in the system is $O(k * n)$ at emergent approach. It should be noted that p is dependent on the network size, whereas k is a constant and for practical applications $p >> k$. Average delay for transmitting messages at the global approach is $O(p)$, whereas average delay is constant and 1 at emergent approach when each robot communicates to only 1-hop neighbors.

Once all the information is collected at a central location, the linear program can find the global optimal solution if the problem instance is not too big for the processing capability and the memory available. On the other hand, the solution with emergent approach is found using limited information, so the solution may not be optimal. However, as the information is shared among neighbors, the quality of the solution improves and optimal solution can be obtained if information sharing is continued until the system reaches a stable state, which is when all robots find the same solution. The proof showing that these iterations finally converge can be found at [1].

VI. Simulation Results

In our simulations, we want to evaluate how well emergent task allocation (ETA) behaves with respect to centralized global optimization approach (CGO) using mixed integer linear programming. For this purpose we have designed an experimental scenario and run ETA with different information exchange methods and CGO. Next, we will discuss the environment, present the behaviors of individual techniques and compare them. Since our main application is mobile sensors, we are interested in finding how well either technique can cover targets. For this purpose we compared the number of targets covered by each technique as well as the solution times. We also experimented with larger networks of robots and targets on bigger environments to show the scalability of ETA. Simulation results with 20 robots - 10 targets and 30 robots - 15 targets can be found at [1].

A. Environment

The environment is bounded and has size 12×12. There are three rectangular obstacles, which are located at $\{(0,4),(5,6)\}$, $\{(4,8),(8,10)\}$ and $\{(8,2),(10,6)\}$ (darkest (dark blue) regions in Figs. 4 and 5). In the environment there are 8 robots which are located at point $(0,0)$, and 6 targets whose locations are unknown initially. The targets follow predefined paths and we assume we can predict their locations for the next timestep, if their locations are known at the current step. Robots are heterogeneous with sensing range and speed differing between 1-2, and communication range 4. Detailed parameters can be found at [1]. All targets except the target t_3 require a single robot for coverage, whereas t_3 requires two robots. Timestep is selected to be 4, so robots arrange themselves according to the environment which they estimate to be in 4 steps. In the experiments, we chose constants at the optimization criterion as $\alpha > \beta > \gamma > \delta$. In other words, the linear program optimizes (1) *target coverage*, (2) *communication between robots*, (3) *area coverage* and (4) *exploration* from highest to lowest priority, respectively.

B. Centralized Global Optimization (CGO)

We show a sample execution of our program to highlight the properties of the solution. Robots start exploring the environment by moving out of the region they explored when they were all at $(0,0)$. The initial explored region is the rectangle $\{(0,0),(1,1)\}$ because the robot with highest sensing range can sense a region of radius 2.

Since there are no targets detected yet, and the communication constraints are satisfied, the robots try to cover as much area as possible while obeying the movement constraints. The new environment is shown in Fig. 4(a) where blue (darker) areas indicate explored regions. Exploration reveals targets t_1 and t_2, and predicts their positions to be $(0,4)$ and $(2,2)$, respectively. Optimal allocation is shown in Fig. 4(b). Robots r_6 and r_8 cover targets, and other robots continue exploration while staying within the communication range. Next, target t_3 is found, which requires two robots to be covered. Robots r_2, r_3 and r_7 continue exploration and r_6 works as the communication bridge while remaining robots are assigned to the targets. Distribution of robots is shown in Fig. 4(c). Two other targets, t_4 and t_5 are discovered at the next step. Moreover, targets t_1 and t_2 move faster than their controller robots, r_1 and r_4, which cannot catch them. However, global optimization finds a solution to this problem by assigning the covering task to other robots that can reach the targets (Fig. 4(d)). Target t_6 is discovered at the next step. At this time, it is not possible to cover all the targets while keeping the communication between all robots. Since target coverage is given more importance, robots are distributed into two independent groups. Robots r_3 and r_5 form one team, while others form the other team. Each team has communication in itself, but cannot reach to the other team. An optimal solution is found and applied for each team. Fig. 4(e) represents result of two optimal solutions. Targets t_1 and t_5 leave the environment at the next step. Team of robots r_3 and r_5 has one target to follow, so while one robot follows target, the other robot, in this case r_3, which is the faster robot, continues exploration. The other team covers all targets, and provides communication in itself. Fig. 4(f) shows the final state of the environment which is totally explored.

Our experiment shows that we can successfully assign tasks to the robots. We can successfully cover individual targets, keep communication distance as long as possible, provide maximum area coverage and explore the environment.

C. Emergent Task Allocation

In this section, we present the performance of the distributed emergent approach under the same scenario. We have run emergent approach for each information exchange method described in Section V with k-closest neighbors where $k = 4$.

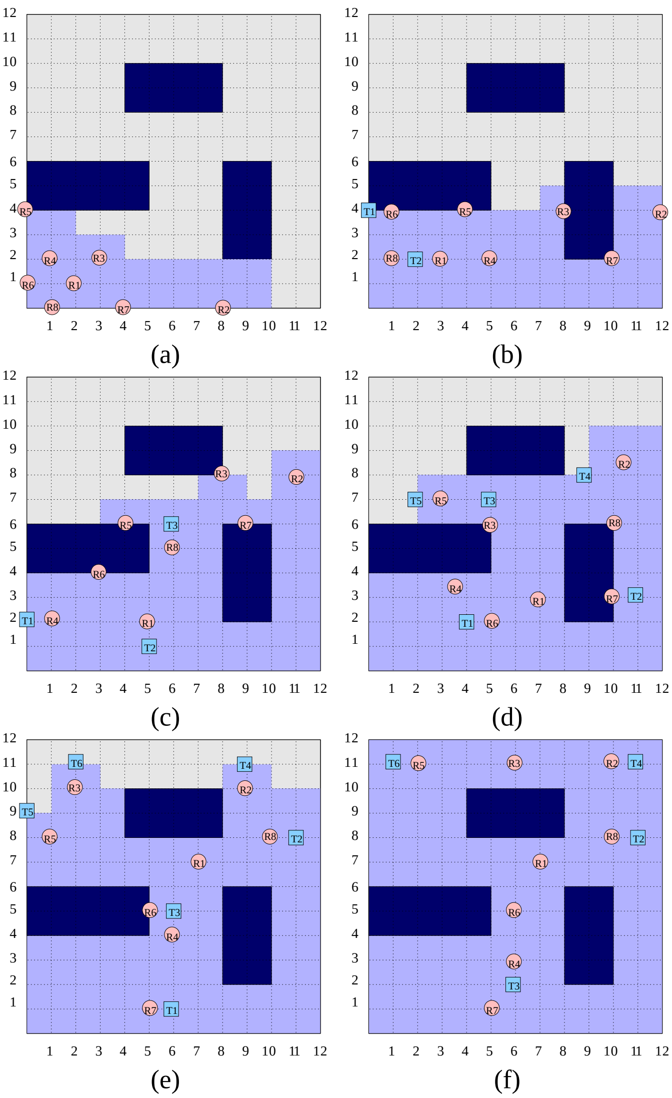


Fig. 4. Sample execution of the Centralized Global Optimization. Robots are represented as circles, and targets are represented as squares. Dark blue (darkest) regions are obstacles, blue (darker) regions are explored regions, and gray (light gray) regions are unexplored regions.

Table I presents running times for each method. It can be seen that there is no significant difference in computation times among ETA methods. On the other hand, as the amount of shared information increase, the performance of ETA increases (see Table II which shows the number of targets covered at each time step). We obtain the worst performance if we just utilize "Intentions", i.e., the least number of targets is covered. The performance of the "Directives" and "Intentions and Directives" are similar and both are better than "Intentions" which suggests that "Directives" are more important. However, both fail to capture all targets. This is because no target information is shared among neighbors, so multiple robots can assign themselves to the same target independently. Finally when the target information is distributed, we obtain the best performance with "Intentions, Directives and Target Assignment" where ETA can cover all the targets. Figures 5 (a) to (f) shows the behavior of ETA in this case. We also run ETA on larger environments and networks to measure the scalability of this approach [1]. These experiments show that the quality of the solution is satisfactory also in large networks, and execution time per robot stays constant irrespective of the network size.

TABLE I
AVERAGE, MAXIMUM AND MINIMUM EXECUTION TIMES PER ROBOT FOR EACH METHOD

	avg. time	*max. time*	*min. time*
ETA w/ Int.	4 s	11 s	<1 s
ETA w/ Dir.	7 s	15 s	<1 s
ETA w/ Int.Dir.	7 s	16 s	<1 s
ETA w/ Int.Dir.Tgt	5 s	16 s	<1 s
CGO	36 min	120 min	9 min

TABLE II
RATIO OF TARGETS COVERED BY ROBOTS FOR EACH METHOD

steps	1	2	3	4	5
ETA w/Int.	2/2	2/3	2/5	2/6	2/4
ETA w/Dir.	2/2	3/3	3/5	4/6	2/4
ETA w/Int.Dir.	2/2	3/3	3/5	4/6	2/4
ETA w/Int.Dir.Tgt	2/2	3/3	5/5	6/6	4/4
CGO	2/2	3/3	5/5	6/6	4/4

Please remember that we chose to exchange information among neighbors only once for each planning phase because of the time limitations of real world applications. However, each update increases the performance and if updates are continued until the system reaches a stable state, the final state will be closer to the global optimal solution.

D. Comparison of CGO and ETA

As it is seen at Table II, the performance of ETA with "Intentions, Directives and Target Assignment" is similar to CGO. On the other hand, ETA is 400 times faster than CGO (Table I). This shows the main drawback of CGO which is the infeasible computation time as the number of robots and targets increase (e.g., when the number of robots is 8 and number of targets is 6, the execution time can reach 2 hours).

VII. CONCLUSIONS

We have presented an emergent task allocation method to solve the task allocation problem of multiple heterogeneous robots for detecting and controlling multiple regions of interest in an unknown environment under defined constraints. We compared our results to a mixed integer linear programming approach which finds the global optimal solution for the given state of the robots, targets and environment. Emergent approach guarantees that each robot in the system computes a limited sized problem, no matter what the number of robots

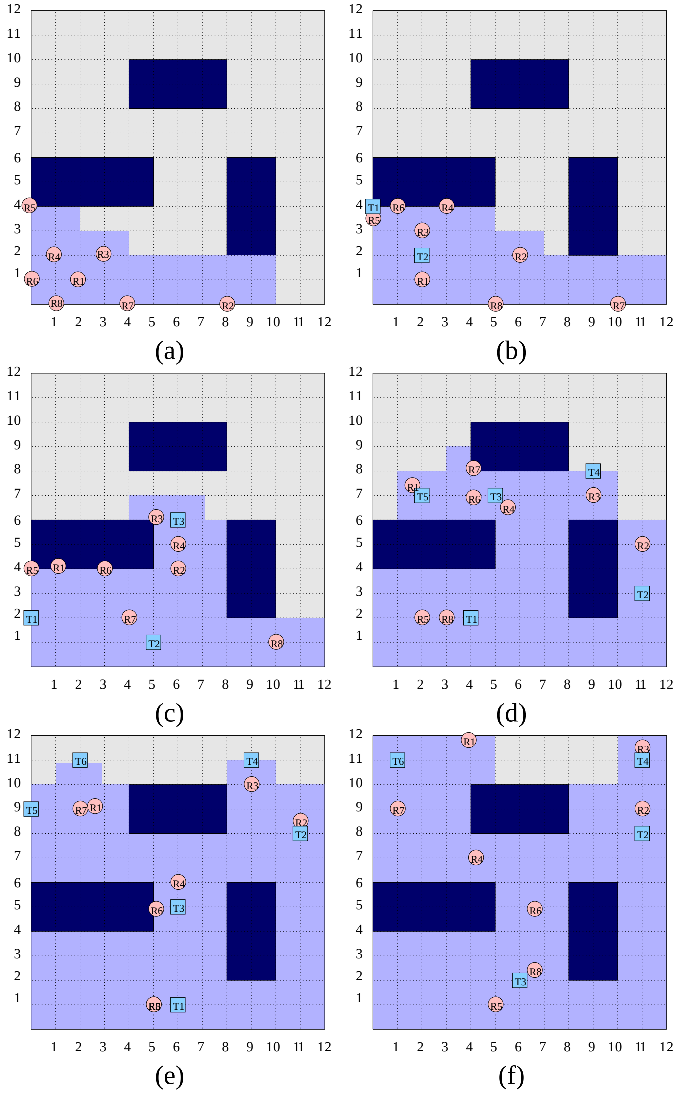


Fig. 5. Sample execution of the Emergent Task Allocation. Robots are represented as circles, and targets are represented as squares. Dark blue (darkest) regions are obstacles, blue (darker) regions are explored regions, and gray (light gray) regions are unexplored regions.

or targets in the environment is. Our simulation results and analysis show that our approach performs similar to global optimal solution at the fraction of its cost (hundreds of times faster). We are planning to implement this approach to in-network task allocation for sensor networks.

References

[1] N. Atay and B. Bayazit, "Emergent task allocation for mobile robots through intentions and directives," Dept. of Computer Science and Engineering, Washington University in St. Louis, Tech. Rep. WUCSE-2007-2, Jan 2007.

[2] R. Davis and R. G. Smith, "Negotiation as a metaphor for distributed problem solving," *Artificial Intelligence*, vol. 20, pp. 63–109, 1983.

[3] B. P. Gerkey and M. J. Matarić, "Sold!: Auction methods for multirobot coordination," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 758–786, October 2002.

[4] S. Botelho and R. Alami, "M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Detroit, Michigan, May 1999, pp. 1234–1239.

[5] R. Zlot and A. Stentz, "Complex task allocation for multiple robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Barcelona, Spain, April 2005, pp. 1515–1522.

[6] G. Thomas, A. M. Howard, A. B. Williams, and A. Moore-Alston, "Multi-robot task allocation in lunar mission construction scenarios," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, Hawaii, October 2005, pp. 518–523.

[7] T. Lemaire, R. Alami, and S. Lacroix, "A distributed tasks allocation scheme in multi-uav context," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, New Orleans, LA, April 2004, pp. 3822–3827.

[8] L. E. Parker, "Alliance: An architecture for fault tolerant multirobot cooperation," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 220–240, April 1998.

[9] B. B. Werger and M. J. Matarić, "Broadcast of local eligibility for multi-target observation," in *5th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, Knoxville, TN, October 4-6 2000, pp. 347–356.

[10] B. P. Gerkey and M. J. Matarić, "Multi-robot task allocation: Analyzing the complexity and optimality of key architectures," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Taipei, Taiwan, September 17-22 2003, pp. 3862–3867.

[11] ——, "A formal analysis and taxonomy of task allocation in multi-robot systems," *Intl. Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, September 2004.

[12] K. Nygard, P. Chandler, and M. Pachter, "Dynamic network flow optimization models for air vehicle resource allocation," in *The American Control Conference*, Arlington, Texas, June 25-27 2001, pp. 1853–1858.

[13] J. Bellingham, M. Tillerson, A. Richards, and J. How, "Multi-task allocation and path planning for cooperating uavs," in *Conference on Coordination, Control and Optimization*, November 2001, pp. 1–19.

[14] C. Schumacher, P. Chandler, M. Pachter, and L. Pachter, "Uav task assignment with timing constraints," in *AIAA Guidance, Navigation, and Conference and Exhibit*, Arlington, Texas, 2003.

[15] Y. Jin, A. Minai, and M. Polycarpou, "Cooperative real-time search and task allocation in uav teams," in *42nd IEEE Conference on Decision and Control*, Maui, Hawaii USA, December 2003, pp. 7–12.

[16] C. Schumacher, P. Chandler, S. Rasmussen, and D. Walker, "Task allocation for wide area search munitions with variable path length," in *The American Control Conference*, Denver, Colorado, June 2003, pp. 3472–3477.

[17] M. Alighanbari, Y. Kuwata, and J. How, "Coordination and control of multiple uavs with timing constraints and loitering," in *The American Control Conference*, vol. 6, Denver, Colorado, June 4-6 2003, pp. 5311–5316.

[18] D. Turra, L. Pollini, and M. Innocenti, "Fast unmanned vehicles task allocation with moving targets," in *43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, December 14-17 2004, pp. 4280–4285.

[19] P. B. Sujit, A. Sinha, and D. Ghose, "Multi-uav task allocation using team theory," in *44th IEEE International Conference on Decision and Control, and the European Control Conference*, Seville, Spain, December 12-15 2005, pp. 1497–1502.

[20] M. A. Darrah, W. Niland, and B.M.Stolarik, "Multiple uav dynamic task allocation using mixed integer linear programming in a sead mission," in *Infotech@Aerospace*, Arlington, Virginia, September 26-29 2005.

[21] A. Elganar and K. Gupta, "Motion prediction of moving objects based on autoregressive model," *IEEE Transactions on Systems, Man and Cybernetics-Part A:Systems and Humans*, vol. 28, no. 6, pp. 803–810, November 1998.