Reactive, Distributed Layered Architecture for Resource-Bounded Multi-Robot Cooperation: Application to Mobile Sensor Network Coverage

Kian Hsiang Low, Wee Kheng Leow
Dept. of Computer Science
National University of Singapore
3 Science Drive 2, Singapore 117543, Singapore
Email: lowkh, leowwk@comp.nus.edu.sg

Marcelo H. Ang Jr.
Dept. of Mechanical Engineering
National University of Singapore
10 Kent Ridge Crescent, Singapore 119260, Singapore
Email: mpeangh@nus.edu.sg

***Abstract*— This paper describes a reactive, distributed layered architecture for cooperation of multiple resource-bounded robots, which is utilized in mobile sensor network coverage. In the upper layer, a dynamic task allocation scheme self-organizes the robot coalitions to track efficiently in separate regions. It uses the concepts of ant behavior to self-regulate the regional distributions of robots in proportion to that of the targets to be tracked in the changing environment. As a result, the adverse effects of task interference between robots are minimized and sensor network coverage is improved. In the lower layer, the robots use self-organizing neural networks to coordinate their target tracking within a region. Quantitative comparisons with other tracking strategies such as static sensor placements, potential fields, and auction-based negotiation show that our approach can provide better coverage and greater flexibility in responding to environmental changes.**

I. Introduction

Sensor network has recently received significant attention in networking, embedded systems, and multi-agent systems [1] due to its wide array of real-world applications (e.g., disaster relief, environment monitoring). In these applications, the distributed sensing task is achieved by the collaboration of a large number of static sensors, each of which has limited sensing, computational, and communication capabilities.

One of the fundamental issues that arises in a sensor network application is coverage. Traditionally, network coverage is maximized by determining the optimal placements of static sensors in a centralized manner, which can be related to the class of art gallery problems [2]. However, recent investigations in sensor network mobility reveal that mobile sensors can self-organize to provide better coverage than static placement [3]. Existing applications have only utilized uninformed mobility (i.e., random motion or patrol) [1]. In contrast, our work here focuses on informed, intelligent mobility to further improve coverage.

Our coverage problem is motivated by the following constraints that discourage static sensor placement or uninformed mobility: a) no prior information on the exact target locations, population densities or motion pattern, b) limited sensory range, and c) very large area to be observed. All these conditions may cause the sensors to be unable to cover the entire region of interest. Hence, fixed sensor locations or uninformed mobility will not be adequate in general. Rather, the sensors have to move dynamically in response to the motion and distributions of targets and other sensors to maximize coverage. We will now refer to mobile sensors as robots since they are the same in this paper's context.

This paper presents a reactive layered multi-robot architecture (Fig. 1a) for distributed sensor network coverage in complex, unpredictable environments. In the upper layer, the workspace is segmented into regions (Fig. 1b) in which the lower-layer method operates. In the lower layer, each robot uses a coordinated motion control strategy based on self-organizing neural networks to cooperatively track the moving targets within a region (Section II). This strategy is also responsible for obstacle avoidance, robot separation to minimize task interference, and navigation between regions via beacons or checkpoints plotted by a motion planner. In the upper layer, the robots employ a dynamic ant-based task allocation scheme to cooperatively distribute themselves in a decentralized manner according to the distributions of targets in the regions (Section III). This scheme enables the robots to track the changing environment on a regional scale and continually self-organizes the regional distributions of robots to the distributions of targets. This paper focuses on the upper-layer task allocation problem, and will describe the lower-layer motion control briefly.

Our framework addresses the following issues, which distinguish it from the other multi-robot architectures:

1) Resource-bounded multi-robot cooperation: Robots with limited communication and sensing capabilities can only extract local, uncertain information of the environment. As such, distributed methodologies are required to process and integrate the noisy, heterogeneous information to improve its quality so that it can be effectively utilized to boost the task performance. Furthermore, if the robots have limited computational power, their cooperative strategies cannot involve complex planning or negotiations. Existing multi-robot architectures have either assumed perfect communications, high computational power

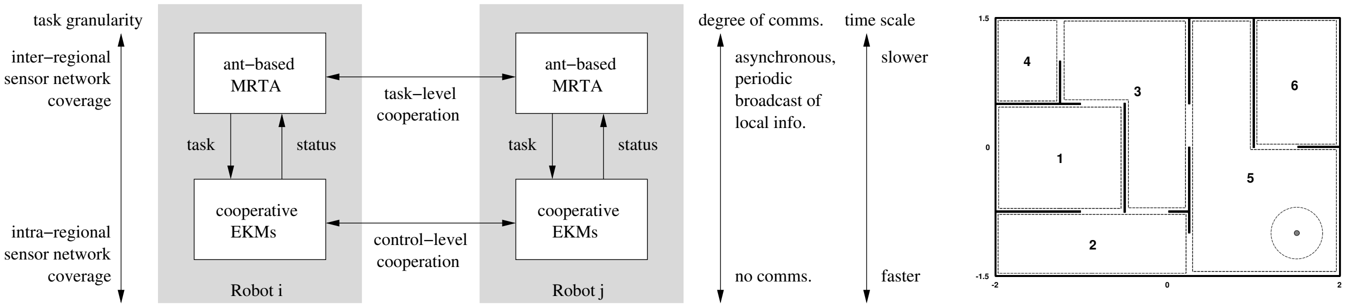

Fig. 1. (a) Reactive distributed layered architecture for multi-robot cooperation (MRTA = Multi-Robot Task Allocation, EKM = Extended Kohonen Map). (b) A 4 m $\times$ 3 m environment that is divided into 6 regions. The circle at the bottom right represents the robot's sensing radius of 0.3 m (drawn to scale). The environment is 42.44 times as large as the robot sensing area.

or global knowledge of the task and other robots ([4]–[6]). In contrast, our proposed architecture caters to resource-bounded robots.

2) Task allocation for multi-robot tasks: Existing Multi-Robot Task Allocation (MRTA) algorithms (i.e., auction- and utility-based) ([4], [6]) generally assume that a multi-robot task can be partitioned into several single-robot tasks. But this may not be always possible or the multi-robot task can be more efficiently performed by groups or coalitions of multiple robots. In our coverage application, each robot coalition is assigned to a region. Our proposed task allocation scheme self-organizes the robot coalitions to the distribution of tasks in these regions.

3) Coalition formation for reactive robots: Existing multi-agent coalition formation schemes ([7], [8]) require complex planning, explicit negotiations, and precise estimation of coalitional cost [9]. Hence, they may not be able to operate in real-time in a large-scale sensor network. Our task allocation method via self-organizing swarm coalitions is reactive, dynamic, and can operate with uncertain coalitional cost and resource-limited robots.

II. Reactive Coordinated Motion Control

In the lower layer (Fig. 1a), a reactive motion control strategy known as *Cooperative Extended Kohonen Maps* (EKMs) is responsible for cooperative target tracking within a region. EKMs have been used for goal-directed, collision-free robot motion in complex, unpredictable environments ([10], [11]). Our implementation extends our previous work [12] by connecting several EKMs to form cooperative EKMs. These self-organizing neural networks cooperate and compete to produce an appropriate motor action for the robot to approach targets, negotiate unforeseen, possibly concave, obstacles, and keep away from robot kins when it is tracking moving targets (Fig. 2). Since its implementation is not the main emphasis of the paper, it will only be described briefly here (see [13] for more details).

The motion control system consists of four types of EKMs: target localization, obstacle localization, robot kin localization, and motor control EKMs. In the presence of a target, the neurons in the target localization EKM, which encodes target location in the local sensory input space $\mathcal{U}'$, are activated (Fig. 2a). A *target field* with the shape of an elongated Gaussian is produced (Fig. 2b) such that the neurons at and near the target location have the strongest activities. The elongated target field plays an important role in allowing the robot to avoid small concave obstacles during obstacle avoidance.

Similarly, the presence of an obstacle activates neurons in the obstacle localization EKMs. The neurons in these EKMs at and near the obstacle locations will be activated to produce *obstacle fields* (Fig. 2c). These obstacle fields are stretched along the obstacle directions such that neurons beyond the obstacle locations are also inhibited to indicate inaccessibility. *Robot kin fields* are activated in a similar way in the robot EKMs in the presense of robot kins.

In activating the motor control EKM, the obstacle fields are subtracted from the target field (Fig. 2d). If the target lies within the obstacle fields, the activation of the motor control EKM neurons close to the target location will be suppressed. Consequently, another neuron at a location that is not inhibited by the obstacle fields becomes most highly activated (Fig. 2d). This neuron produces a control parameter that moves the robot away from the obstacle. While the robot moves around the obstacle, the target and obstacle localization EKMs are continuously updated with the current locations and directions of the target and obstacles. Their interactions with the motor control EKM produce fine and smooth motion control of the robot to negotiate the obstacle and reach the target. In the case of multi-robot tracking of multiple targets, multiple target fields and robot kins fields are activated. The robots act like highly repulsive obstacles to other robots, thus separating them from each other.

One noteworthy aspect of cooperative EKMs is that no communication is needed for the robots to cooperate in target tracking. They are only required to discriminate between targets, obstacles and robot kins. Cooperative EKMs can be used by the robots to perform tracking in a region with simple, unpredictable obstacles. However, when it is used in a more realistic and complex environment with several occluded regions (Fig. 1), it needs to be integrated with our task allocation scheme to perform the distributed sensor network coverage task well (Section IV).

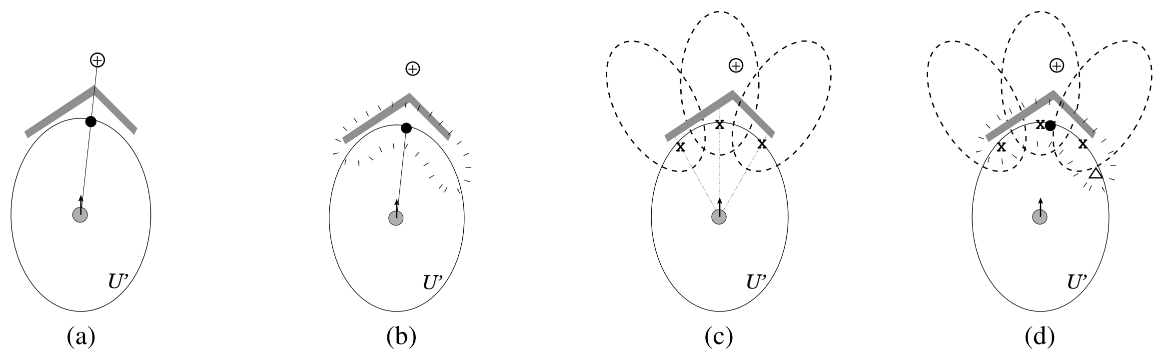

Fig. 2. Cooperative EKMs. (a) In response to the target $\oplus$, the nearest neuron (black dot) in the target localization EKM (ellipse) of the robot (gray circle) is activated. (b) The activated neuron produces a target field (dotted ellipse) in the motor control EKM. (c) Three of the robot's sensors detect obstacles and activate three neurons (crosses) in the obstacle localization EKMs, which produce the obstacle fields (dashed ellipses). (d) Subtraction of the obstacle fields from the target field results in the neuron at $\triangle$ to become the winner in the motor control EKM, which moves the robot away from the obstacle.

III. Ant-Based Multi-Robot Task Allocation

Many multi-robot tasks (e.g., foraging, transportation, manipulation, sensing, and exploration) have been inspired by social insects [14], in particular, ants. Our MRTA scheme encapsulates three concepts of ant behavior: (a) encounter pattern based on waiting time, (b) self-organization of social dominance, and (c) dynamic task allocation. These features help to self-organize the robot coalitions to the distributions of targets in different regions.

A. Encounter Pattern Based on Waiting Time

Encounter patterns provide a simple, local cue for ants with sensory and cognitive limitations to assess regional densities of ants and objects of interest, which are crucial to regulating the division of labor [15]. Instead of relying on global communication to relay target positions and density estimation [16], our scheme uses encounter patterns to predict target density via local sensing. Regional robot density is captured in a similar way using local communication.

An encounter pattern can be derived from a series of waiting time or interval between successive encounters. This simple form of information processing has accounted for the complex adaptive process of task allocation in ant colonies [17]. In our distributed sensor network coverage task, the waiting time for other robots and targets is defined in terms of encounters with the robots and targets. A robot encounter is defined as a reception of a message from another robot in the same region. A target encounter is defined as an increase in the number of targets tracked between previous and current time steps. For a robot i in region r, the waiting time for other robots $w_{ir}(k)$ and targets $w'_{ir}(k)$ is the time interval between the $(k-1)$th and kth encounters. Note that each waiting time is subject to stochastic variation. Hence, multiple samplings of waiting time have to be integrated to produce an accurate estimate of the regional density. The average waiting time $W_{ir}(k)$ between the $(k-1)$th and kth robot encounters for a robot i in region r is computed as:

$$\begin{aligned} W_{ir}(k) = &\ \frac{1}{n} w_{ir}(k) + \frac{n-1}{n} W_{ir}(k-1) \\ n = &\ \min(k, n_{max}) \ . \end{aligned} \tag{1}$$

$W'_{ir}(k)$ is updated in the same manner. Both waiting times are updated according to the changing environment, and are inversely proportional to the robot and target densities in region r. The target density directly reflects the task demand of the region. The robot density reflects the amount of physical interference in the region, which is inversely proportional to the task demand. Therefore, the task demand $S_{ir}(k)$ of a region r can be determined by robot i using the average robot waiting time $W_{ir}(k)$ and the average target waiting time $W'_{ir}(k)$:

$$S_{ir}(k) = \frac{W_{ir}(k)}{W'_{ir}(k)} \ . \tag{2}$$

The task demand $S_{ir}(k)$ will be used in the self-organization of social dominance (Section III-B) as well as in dynamic task allocation (Section III-C).

B. Self-Organization of Social Dominance

The division of labor in an ant colony is strongly influenced by its social dominance order [18], which self-organizes to match the task demands of the colony and the changing environment. Our scheme is inspired by this concept to move the robots out of a region that has a lower target-to-robot density ratio than the other regions. Rather than fixing the dominance order [19], the social dominance of the robots in each coalition is self-organized according to their individual task performance. To elaborate, robots in the same coalition engage in dominance contests at a regular interval τ if they are within communication range. The winner increases its tendency to stay in the current region while the loser increases its tendency to leave the current region and join another coalition in other regions. When robot i encounters robot j in region r, the probability of robot i winning a contest against robot j is defined as:

$$P(\text{robot } i \text{ winning}) = \frac{n_i^2 S_{ir}^2}{n_i^2 S_{ir}^2 + n_j^2 S_{jr}^2} \tag{3}$$

where S_{ir} and S_{jr} are respectively the task demand of region r determined by robot i and robot j, and n_i and n_j are the number of targets currently under observation by robot i and robot j respectively. Equation 3 implies that robot i would most likely win the contest if it observes more targets than robot j. However, if both are tracking the same number of targets, then their individual evaluation of the task demand can be used to differentiate them. This will distinguish a robot that has been observing the targets for a long time from another that just encounters the same number of targets.

To inject the influence of social dominance on the self-organization of robot coalitions, each time a robot i wins a contest (Eq. 3), it increases its tendency of staying in the current region, which is represented by the threshold $\theta_i(t)$ to be used for dynamic task allocation (Section III-C):

$$\theta_i(t) = \theta_i(t-1) + \delta \tag{4}$$

where δ is small constant. Conversely, each time the robot loses, it decreases its tendency of staying in the region.

$$\theta_i(t) = \theta_i(t-1) - \delta \ . \tag{5}$$

θ_i varies in the range [0,1] to prevent robots from being overly submissive or dominating.

C. Dynamic Task Allocation

The distributed task allocation algorithm in ants can efficiently arrange the ants in proportion to the amount of work in the changing environment [20]. In a similar spirit, our scheme aims to self-organize the robot coalitions to the distributions of targets in different regions. In a cooperative multi-robot task, the robots are bound to interfere with each other's ongoing activity either physically (e.g., space competition) or non-physically (e.g, shared radio bandwidth, conflicting goals). In the extreme case, when too many robots are involved, little or no work gets done as they totally restrict each other's movement. Hence, task interference has an adverse effect on the overall system performance [19]. Knowing that physical interference can be implied from robot density [19], our task allocation scheme has taken physical interference into consideration by estimating robot density. In contrast, existing MRTA methods ([4], [6]) generally assume that the multi-robot task can be partitioned into independent single-robot tasks. Thus, no interference would result. Bucket brigade algorithm [21] can eliminate interference by assigning the robots to separate regions. However, it cannot respond in real-time to changing regional distributions of targets due to target motion.

Our dynamic task allocation scheme is based on the notion of response thresholds [22]. In a threshold model, robots with low response thresholds respond more readily to lower levels of task demand than do robots with high response thresholds. Performing the task reduces the demand of the task. If robots with low thresholds perform the required tasks, the task demand will never reach the thresholds of the high-threshold robots. However, if the task demand increases, high-threshold robots will engage in performing the task.

MRTA strategies that utilize fixed response thresholds ([5], [16]) are incapable of responding effectively to dynamic environments [22]. In contrast, the thresholds in our scheme are continuously updated by the self-organizing process of social dominance (Section III-B).

To be effective in task allocation, a robot must at least have some knowledge of the task demands in its neighboring regions in order to make rational task decisions. To do so, robot i maintains a memory of the task demand S_{ir} of each region r and the amount of time T_{ir} that it previously spent in region r. T_{ir} can be used as a certainty measure of S_{ir}. In addition to computing S_{ir} using Equation 2, S_{ir} can also be updated when robot i receives a message from a neighboring robot j with S_{jr} less than S_{ir}. Then S_{ir} and T_{ir} are updated to take the values S_{jr} and T_{jr} respectively. In this manner, the task demands of the regions are kept in memory. Robot i can then predict the region with the greatest task demand and join that region. At every time interval of τ, if S_{ir} receives no update, the certainty value T_{ir} is decreased by τ while the task demand S_{ir} is increased by a small amount, such that its magnitude reflects the robot's motivation to explore. However, S_{ir} will not be increased beyond the maximum of the S_{ig} values over all regions g.

Our distributed MRTA scheme uses a stochastic problem solving methodology. It is performed at intervals of τ to allow for multiple samplings of waiting time during each interval (Section III-A). The probability of a robot i to stay in its current region c is defined as:

$$P(\text{stay}) = \frac{S_{ic}^2}{S_{ic}^2 + (1-\theta_i)^2 + T_{ic}^{-2}} \ . \tag{6}$$

On the other hand, the probability of a robot i to leave region c to go to region r is defined as:

$$P(\text{leave}) = \frac{S_{ir}^2}{S_{ir}^2 + \theta_i^2 + T_{ir}^{-2} + d_{cr}^2} \tag{7}$$

where d_{cr} is the pre-computed collision-free distance between region c and region r, which can be viewed as the cost of task switching. Note that a robot that loses in the dominance contest in a coalition does not always leave the region. If it experiences a higher task demand in its region than in other regions, it will have a high tendency of remaining in its coalition.

From Equations 6 and 7, if the robot does not respond to any regions, it will not switch task and will remain in the current coalition. The robot may also respond to more than one region. This conflict is resolved with a method that is similar to Equation 3. The probability of a robot i choosing a region r that it has responded to is:

$$P(\text{choose}) = \frac{(S_{ir} \ln T_{ir})^2}{\sum_r (S_{ir} \ln T_{ir})^2} \ . \tag{8}$$

If the robot i chooses region r that is not the current region c, then it will employ the reactive motion control strategy in Section II to move through the checkpoints plotted by the planner to region r. The generation of checkpoints is performed by the approximate cell decomposition method for motion planning [23].

IV. Experiments and Discussion

This section presents a quantitative evaluation of the reactive, layered multi-robot architecture for distributed mobile sensor coverage in a complex, unpredictable environment. The experiments were performed using Webots, an embodied simulator for Khepera mobile robots, which incorporated 10% white noise in its sensors and actuators. 12 directed distance sensors were also modelled around its body of radius 3 cm. Each sensor had a range of 17 cm, enabling the detection of obstacles at 20 cm or nearer from the robot's center, and a resolution of 0.5 cm to simulate noise. Each robot could also sense targets and kin robots at 0.3 m or nearer from its center. A 4 m $\times$ 3 m environment (Fig. 1) was used to house the Khepera robots and targets, which were randomly scattered initially. The number of robots varied from 5, 10 to 15, which corresponded to total robot sensing area of 11.8%, 23.6%, and 35.3% of the environment size. The mobile targets were forward-moving Braitenberg obstacle avoidance vehicles [24] that changed their direction and speed with 5% probability.

Two performance indices were used. The first performance index determines the overall sensor network coverage performance of the robots [25]:

$$\text{sensor network coverage} = \sum_{t=1}^{T} 100 \frac{n(t)}{NT} \tag{9}$$

where N is the total number of targets, n is the number of targets being tracked at time t, and the experiment lasts T amount of time. N and T are fixed respectively as 30 targets and 10000 time steps at intervals of 128 ms for all experiments.

Using this index, a quantitative test was conducted to compare the sensor network coverage of the robots adopting four *fully distributed* tracking strategies: (1) static, (2) potential fields, (3) cooperative EKMs only, (4) reactive, layered architecture (ant-based MRTA with cooperative EKMs). In the first method, static sensors are placed at least 0.6 m apart to ensure no overlap in coverage. The potential fields method is a well-known motion control technique utilized in [25] for cooperative multi-robot tracking of moving targets. While potential fields and cooperative EKMs allow the robots to cooperate in tracking at control level, ant-based MRTA enables them to cooperate at task level.

On the other hand, recent proposals of sensor network organization are hierarchically structured ([3], [26]). At the bottom of the hierarchy, the robots track with cooperative EKMs, which is the same as the third method. In each region, a coalition leader is elected. It regards the robots in its coalition as resources and negotiates with coalition leaders in the other regions to efficiently allocate them according to the regional distributions of targets. This negotiation is conducted iteratively at every interval of τ using an auction-based mechanism and attempts to balance the ratio of number of robots (resources) over all regions with the ratio of targets. To do so, we assume that each coalition leader is capable of obtaining more information, i.e., the exact number of robots and targets in its own region. Furthermore, it is able to communicate with all robots in its own coalition to obtain their task performance and command them to move to other regions if they are tracking minimal targets and observing many kin robots. Lastly, it has to synchronize its negotiation with coalition leaders in the other regions via long-range communication. In contrast, the robots endowed with our reactive, layered architecture only require local sensing information (i.e., 0.3 m range) and short-range communication (i.e., 1.0 m range). The robot coalitions can also be self-organized asynchronously without negotiation. We have implemented this hierarchical approach to compare with our distributed layered architecture.

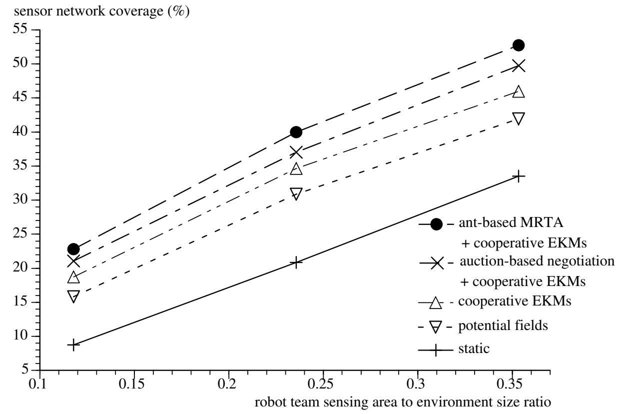

Fig. 3. Comparison of sensor network coverage using different tracking strategies.

Test results (Fig. 3) showed that our reactive, layered architecture provided better coverage than the other strategies. Notice that while the hierarchical approach (auction-based negotiation with cooperative EKMs) used more information, longer communication range and more complex negotiation, it did not perform better than our reactive, layered architecture. This will be explained later.

The second performance index determines the total coalitional cost [8] of the robots. Given a set of connected regions where coverage tasks are to be performed, and a set A of M robots, the task allocation algorithm assigns a robot coalition $C_r \subseteq A$ to the coverage task in region r such that (a) $\bigcup_r C_r = A$, (b) $\forall r \neq s, C_r \bigcap C_s = \emptyset$, and (c) each C_r has a positive cost $|n_r/N - m_r/M|$ where m_r and n_r are the number of robots and targets in region r respectively. The objective is to minimize the total coalitional cost [8]:

$$\text{total coalitional cost} = \sum_r \left| \frac{n_r}{N} - \frac{m_r}{M} \right| . \tag{10}$$

This index varies within the range [0,2]. A coalitional cost of 0 implies that the robot distribution over all regions is exactly proportional to the target distribution. In this manner, interference between robots is at its minimum, which will improve overall coverage. High costs imply the opposite.

Test results (Fig. 4) showed that if either auction-based negotiation or ant-based MRTA was integrated with cooperative EKMs, the total coalitional cost could be reduced further. Hence, we could conclude from Figures 3 and 4 that with a lower cost, a higher coverage can be achieved. Coalitional cost is higher with fewer robots because with less robots, it is

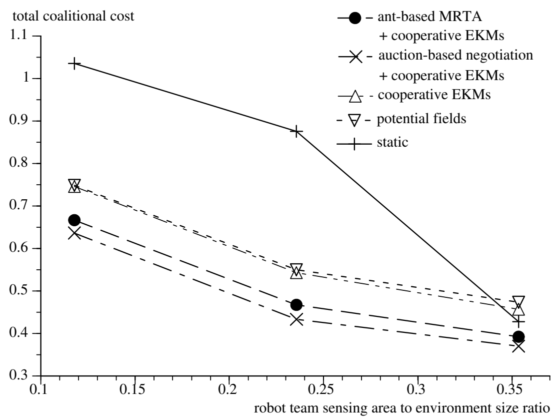

Fig. 4. Comparison of total coalitional cost using different tracking strategies.

more difficult to achieve the same proportion of robots to that of the targets over all regions. Furthermore, with ant-based MRTA, each robot received fewer messages from the other robots when there were less robots. As a result, the robots were less certain about the task demands in other regions. This caused the scheme to be less effective in the distribution of robots.

Interestingly, although auction-based negotiation achieved slightly lower coalitional cost than ant-based MRTA, its coverage performance was worse. The auction-based negotiation among the coalition leaders prioritized the task of balancing the ratio of number of robots over all regions with the ratio of targets. As a result, a robot that performed the worst in a region over-populated with robots would be commanded by its coalition leader to move to another region deprived of robots. This robot would be forced to drop the targets that it was currently tracking and renew its search for targets in the new region that it moved to. If this new region was large, it might take even longer to find the targets. In contrast, since our ant-based MRTA was stochastic, the robot had a tendency to remain in its current region and continue tracking the targets (Section III-C). This accounted for higher coverage of ant-based MRTA over auction-based negotiation.

V. Conclusion

This paper describes a reactive, distributed layered architecture for resource-bounded multi-robot cooperation, which is utilized in the application of mobile sensor network coverage. In particular, our proposed upper-layer method for task allocation via self-organizing swarm coalitions uses the concepts of ant behavior to self-regulate the division of labor in proportion to the task demands across regions in the changing environment. Hence, our multi-robot system functions like a complex adaptive system.

References

[1] V. Lesser, J. C. L. Ortiz, and M. Tambe, *Distributed Sensor Networks: A Multiagent Perspective*. Kluwer Academic Publishers, 2003.

[2] J. O'Rourke, *Art Gallery Theorems and Algorithms*. London, UK: Oxford University Press, 1987.

[3] O. Yadgar, S. Kraus, and C. L. Ortiz Jr., "Hierarchical information combination in large-scale multiagent resource management," in *Communications in Multiagent Systems: Agent Communication Languages and Conversation Policies*, ser. LNCS, M.-P. Huget, Ed. Springer-Verlag, 2003, vol. 2650, pp. 129–145.

[4] B. P. Gerkey and M. J. Matarić, "Sold!: Auction methods for multi-robot coordination," *IEEE Trans. Robot. Automat.*, vol. 18, no. 5, pp. 758–768, 2002.

[5] M. J. B. Krieger and J.-B. Billeter, "The call of duty: Self-organized task allocation in a population of up to twelve mobile robots," *Robotics and Autonomous Systems*, vol. 30, no. 1-2, pp. 65–84, 2000.

[6] L. E. Parker, "ALLIANCE: An architecture for fault tolerant multi-robot cooperation," *IEEE Trans. Robot. Automat.*, vol. 14, no. 2, pp. 220–240, 1998.

[7] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé, "Coalition structure generation with worst case guarantees," *Artificial Intelligence*, vol. 111, no. 1-2, pp. 209–238, 1999.

[8] O. Shehory and S. Kraus, "Methods for task allocation via agent coalition formation," *Artificial Intelligence*, vol. 101, no. 1-2, pp. 165–200, 1998.

[9] S. Kraus, O. Shehory, and G. Taase, "Coalition formation with uncertain heterogeneous information," in *Proc. AAMAS*, 2003, pp. 1–8.

[10] K. H. Low, W. K. Leow, and M. H. Ang Jr., "Enhancing the reactive capabilities of integrated planning and control with cooperative extended Kohonen maps," in *Proc. IEEE ICRA*, 2003, pp. 3428–3433.

[11] R. P. H. Rao and O. Fuentes, "Hierarchical learning of navigational behaviors in an autonomous robot using a predictive sparse distributed memory," *Autonomous Robots*, vol. 5, no. 3-4, pp. 297–316, 1998.

[12] K. H. Low, W. K. Leow, and M. H. Ang Jr., "A hybrid mobile robot architecture with integrated planning and control," in *Proc. AAMAS*, 2002, pp. 219–226.

[13] ——, "Action selection for single- and multi-robot tasks using cooperative extended Kohonen maps," in *Proc. IJCAI*, 2003, pp. 1505–1506.

[14] E. Bonabeau, M. Dorigo, and G. Théraulaz, "Inspiration for optimization from social insect behaviour," *Nature*, vol. 406, pp. 39–42, 2000.

[15] D. M. Gordon, "Interaction patterns and task allocation in ant colonies," in *Information Processing in Social Insects*, C. Detrain, J. L. Deneubourg, and J. M. Pasteels, Eds. Basel, Switzerland: Birkhäuser Verlag, 1999, pp. 51–67.

[16] B. Jung and G. S. Sukhatme, "Tracking targets using multiple robots: The effects of environment occlusion," *Autonomous Robots*, vol. 13, no. 3, pp. 191–205, 2002.

[17] A. E. Hirsh and D. M. Gordon, "Distributed problem solving in social insects," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1-4, pp. 199–221, 2001.

[18] S. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau, *Self-Organization in Biological Systems*. Princeton, NJ: Princeton University Press, 2001.

[19] D. Goldberg and M. J. Matarić, "Interference as a tool for designing and evaluating multi-robot controllers," in *Proc. AAAI*, 1997, pp. 637–642.

[20] C. Tofts, "Algorithms for task allocation in ants (a study of temporal polyethism: Theory)," *Bull. Math. Biol.*, vol. 55, no. 5, pp. 891–918, 1993.

[21] M. Schneider-Fontań and M. J. Matarić, "Territorial multi-robot task division," *IEEE Trans. Robot. Automat.*, vol. 14, no. 5, pp. 815–822, 1998.

[22] E. Bonabeau, M. Dorigo, and G. Théraulaz, Eds., *Swarm Intelligence: From Natural to Artificial Systems*. Oxford, UK: Oxford University Press, 1999.

[23] S. Quinlan and O. Khatib, "Towards real-time execution of motion tasks," in *Experimental Robotics II: Proc. 2nd International Symposium on Experimental Robotics*, R. Chatila and G. Hirzinger, Eds. Springer-Verlag, 1991.

[24] V. Braitenberg, *Vehicles: Experiments in Synthetic Psychology*. MIT Press, 1984.

[25] L. E. Parker, "Distributed algorithms for multi-robot observation of multiple moving targets," *Autonomous Robots*, vol. 12, no. 3, pp. 231–255, 2002.

[26] J. C. L. Ortiz, E. Hsu, M. desJardins, T. Rauenbusch, B. Grosz, O. Yadgar, and S. Kraus, "Incremental negotiation and coalition formation for resource bounded agents: Preliminary report," in *Negotiation Methods for Autonomous Cooperative Systems: Papers from the AAAI Fall Symposium*, C. Tsatsoulis, Ed. Technical Report FS-01-03, Menlo Park, CA: AAAI Press, 2001, pp. 53–62.