Greedy Extension of Localized Auction Based Protocols for Wireless Robot-Robot Coordination

I. Mezei*, V. Malbasa* and I. Stojmenovic**,*

* FTN, University of Novi Sad, Serbia

** SITE, University of Ottawa, Canada

imezei@uns.ac.rs, malbasa@uns.ac.rs, stojmenovic@gmail.com

***Abstract*— We assume that an event was reported to one of robots, and a response by one robot is required. The goal of robot-robot coordination for task assignment is to select the best robot for responding to a reported event so that communication cost for selecting, and response time for performing the task are minimized. Existing solutions, except those recently proposed in [3], are either centralized, neglecting communication cost, assuming complete graph, or based on flooding with individual responses to robot decision maker (*simple auction* protocol – *SAP*), ignoring communication cost and response time bound. This article proposes *greedy* improvement to previously proposed (in [3]) *k*-hop simple auction protocol (*k-SAP*) and *k*-hop *simple auction aggregation* protocol (*k-SAAP*) for task assignment in multi-hop wireless robot networks. After decision about the best robot is made by *k-SAP or k-SAAP*, new *1-SAP* greedy auction is initiated by that robot in order to search for possibly better robot in 1-hop neighborhood. Such greedy approach proceeds until no better robot is found. Improvement of new *k-SAPG* and *k-SAAPG* over *k-SAP* and *k-SAAP* by applying greedy approach is shown by simulation results.**

I. Introduction

According to recent review paper on status of robotics [1], communication between entities is fundamental to both cooperation and coordination and hence the central role of robot networks. Another major topics of robot-robot coordination is the task assignment. In this article we only consider a simple task assignment formulation. We assume that an event was reported to one of robots, and a response by one robot is required. Thus the task assignment problem is to find the best robot to respond to an event and assign task to it. The input of the problem is information about event (location, response time etc.), and the output is assigned task to the best robot.

Robots are organized into a network that is modeled as the unit disk graph (UDG). All robots have the same transmission radius R, and message sent by one robot is received by all neighboring robots located at distance up to R. Robot receiving the event information needs to decide which robot is the best to respond. To make decision, it initiates message exchange (sending and receiving messages) among robots, following UDG. Normally, messages originate from and converge toward robot decision maker.

Solutions described in this article do not depend on the particular environment served by networked robots. One such environment of interest to us is wireless sensor and robot networks (WSRN), as an extension of multi robot systems. WSRN consist of sensors and robots linked by wireless medium to perform distributed sensing of physical world, processing of sensed data, making decisions and acting upon sensed events. We will illustrate our problem statement using this scenario. Upon event occurrence (for example, a fire, or the failure of a sensor), sensors detect event and route information to one of robots in the vicinity, which may not be the closest one. The robot that receives report may itself be the best candidate for responding. However, a remote, busy or energy limited robot could receive report.

Most existing solutions referring to multi-robot coordination for single or multiple events, single or multiple robots, single or multiple tasks to each robot etc. are centralized. One of robots, or a central entity, gathers all the information from other robots and makes a decision. Communication cost for gathering information in case of multi-hop robot networks is rarely considered. Indirectly (since no details of communication protocols used are given), a complete graph (where each robot is within communication distance to any other robot) is assumed. Centralized solutions usually define coordination problem as an integer linear programming problem. The main advantage of a centralized solution is that, theoretically, optimal solution can be found. However, centralized solution features high computation and communication overhead, lack of scalability and slow responsiveness. Moreover, the actual cost for communicating is ignored, especially for large robot networks. It is further not clear how robots communicate if the graph is not complete one. Centralized solutions also have low fault tolerance if leader is malfunctioning.

Localized and distributed solutions utilize spreading all decision making and planning responsibility among robots. We consider here the multi-hop UDG scenarios, where the communication graph is not complete. Robots use only locally available information to make their decision. Good scalability and fault tolerance are the main advantages. Proposed solutions are normally close to optimal one. However, decisions made based on the local information can be sometimes highly suboptimal. Except article [3], only one distributed solution is designed for multi-hop scenario, simple auction protocol [2], which considers multi-hop UDG model of robot communication. It is flooding based; each robot retransmits received search request exactly once, and responds to auctioneer by separate routing task. For large robot networks, it incurs unacceptable delay in selecting the best robot, although the best responding robot is expected to be near the event.

We proposed a few localized solutions, based on market paradigm, called *auction aggregation protocols*

978-1-4244-5349-8/09/$26.00 ©2009 IEEE

Authorized licensed use limited to: King Fahd University of Petroleum and Minerals. Downloaded on February 26,2010 at 01:20:20 EST from IEEE Xplore. Restrictions apply.

and improved simple auction protocol in [3]. The bidding process is spreading to neighboring robots until no improvement can be envisioned within *k*-hop neighborhood of a robot that analyzes if any more remote robot could provide better service than the best service it is aware of. If not, it stops search process and responds back to its 'parent' robot with best possible recommendation it has. During the bid gathering process, best bids are forwarded back to the auctioneer robot by intermediate robots. The main advantage is that search is limited to some neighborhood and flooding potentially huge robot network is avoided.

To improve suboptimality of localized, especially for large networks and low multi-hop values, 1-hop greedy search for better robot could be added after original assignment is proposed. That is, the selected robot, after getting the task, will check if any of its neighbors has lower cost. If so, it can reassign the task, and such search can be repeated by the selected neighbor, until no improvement is possible. In this article we propose two greedy auction based protocols (namely *k-SAPG* and *k-SAAPG*) and compare it with our previous work and with simple auction protocol. Simulation data confirm findings and show the performance of our protocols in some scenarios.

II. Literature Review

A. Market-based task assignment and auctions

For robot-robot coordination, a market-based approach [4] is considered. It is based on auctions organized by robot or sink (central unit) collecting the task, the cost of performing tasks by each robot and potential benefit to the team. Robots positioned in local neighborhood participate, but the locality is not pre-determined; it is rather task-dependent. Robots participating in the auction decide on whether or not to 'invite' more robots to the auction, as the invitation themselves cause communication overhead. Auctions as a coordination tool have been used since Contract Net Protocol was published [5] and several similar protocols are used in robotics. One of the well-known auction protocols is MURDOCH [6]. It uses anonymous broadcasting as a means to communicate and has the following five distinct steps: task announcement, metric evaluation, bid submission, auction closing and progress monitoring/contract renewal. However, MURDOCH assumes complete graph among robots, while we use UDG. Similarly, in [7] and [8], local auctions are used as a distributed solution to dynamic multi robot task assignment (MRTA). However, all robots participating in auction can communicate directly to the auctioneer. There are some articles that report improvements of auction algorithms used as solution to MRTA in terms of computational complexity [9], [10], [11].

Survey article [12] summarizes research work done in the field of robot coordination using market-based approach. Auction can be either centralized (for all robots) or localized, where only nearby robots will respond. Market-based approaches have yet to be implemented on teams of more than a few robots [12]. There is no discussion of communication cost for large robot teams, except a simple statement (Table 3 in [12]) that communication cost is proportional to the number of robots.

To the best of our knowledge, bid aggregation in task assignment problem in robot-robot coordination was not considered in literature. We refer here to the aggregation of responses of several robots by an intermediate robot, which then selects the best of them and forwards only that bid to the auctioneer. For example, in [13], robots aggregate information gathered from bids of other robots to improve the decision making process for its own bid, for the map exploration task. Communication issues are not discussed in [13] (only two robots are used in experiments).

B. Simple auction protocol

We identified only one protocol [2] for single task single robot (called 'actor' in [2]) assignment, which explicitly considers multi-hop scenarios (UDG). It is a localized solution for actor-actor coordination based on 'auction protocol' [2], and is called here a simple auction protocol (*SAP*). The request for service is flooded from actor node that collected the report, and each actor responds back (the offer to provide service and the cost of doing it) to it by separate routing task. If blind flooding is used for actor search, each robot retransmits the request upon receiving it for the first time, and ignores it afterwards. This protocol can always find the closest robot to the event, since all robots are consulted. However, the response time can be large if the best robot is near the event, but robot network is large and the response from all robots is gathered before a decision is made.

C. Improved SAP and auction aggregation protocols

For simplicity, we assume that the robot network is connected and the event location is known to the robot collector which initiates auction. Note that in WSRN sensor nodes may be used to connect some robots; however this scenario is not considered here.

Five new protocols are proposed in [3]. In this paper we will propose greedy improvements for the following two: *k-SAP* and *k-SAAP*. The first one is an improvement of simple auction protocol (*SAP*). Instead of flooding the whole network, one can search for bids only among robots located up to *k* hops away from bidding robot. This protocol applies limited flooding (only up to *k*-hop neighbors), and it is designated as *k-SAP*. In example in Fig. 1, *1-SAP* gather bids from *R2* and *R3* only, and thus

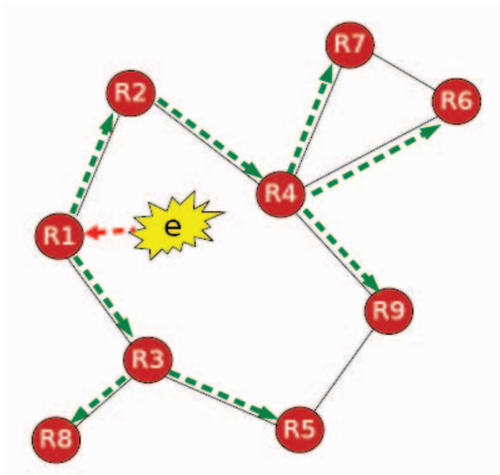

Figure 1. Auction aggregation protocols

Authorized licensed use limited to: King Fahd University of Petroleum and Minerals. Downloaded on February 26,2010 at 01:20:20 EST from IEEE Xplore. Restrictions apply.

the best bid from *R4* (which is the closest) is not considered. *2-SAP* floods from *R1* robots *R2, R4, R3, R8* and *R5*. *3-SAP* consults all robots in network.

Another improvement of *SAP* in terms of communication overhead reduction is the use of *auction aggregation* protocol. Instead of using separate routing tasks, the constructed tree can be used for reporting back. The protocol has tree 'expansion' and tree 'contraction' phases. Tree expansion starts from collecting robot *R1* by creating a tree rooted at *R1* (see Fig. 1). Retransmissions create a response tree. Each node, with retransmission, includes ID of its parent robot in the message, so that robots can locally decide whether or not they are leaves in the created tree. Note that each node selects only one parent, in case of multiple received bids (e.g. *R9* is joined only to *R4*). They become leaves if they do not retransmit the bid or do not hear any other robot listing them as their parent.

Leaf nodes start responding back to parent robots, with the best cost they are aware of. This is in fact auction aggregation and thus reduces number of messages in bidding phase. Each intermediate node waits to hear from all neighbors, which declared it as parent thus becoming a local collector. After hearing, they select the best cost and report further towards the collector. Collector at the end decides which robot is the best to perform the required action, and routes the decision to that robot. In the example in Fig. 1, robots *R5, R6, R7, R8* and *R9* are leaves in created tree, and return their bids to their parent nodes. *R4* returns to its parent *R2* its own bid as the best it is aware of; similarly *R3* also returns its own bid. *R2* returns *R4* as the best bidder. The root node (*R1*) then selects the best bid (in this case from *R4*) from two received offers, and delivers the task to *R4* along created path *R1-R2-R4.* This version of auction aggregation protocol is designated as *SAAP* (simple auction aggregation protocol). In case of limited flooding (only up to *k*-hop neighbors from the collector), it is called *k-SAAP*. The difference between *k-SAAP* and *k-SAP* (and similarly between *SAP* and *SAAP*) is that individual bids are aggregated at intermediate nodes, instead of routing all of them back to the collector robot.

III. Greedy Improvements

Only *SAP* and *SAAP* protocols are such that optimal robot is always found, since all robots are included in auctions. However, *SAAP* and especially *SAP* both feature high communication cost. First improvement of *SAP*, designated as *k-SAP*, is shown to provide much lower communication overhead in cases where it is not required to find the best robot in *all* cases.

To improve suboptimality of localized decisions of *k-SAP* and *k-SAAP*, especially for large networks and low values of *k*, 1-hop greedy search for better robot could be added after original assignment is proposed. That is, the selected robot, after getting the task, will check if any of its neighbors has lower cost. If so, it can reassign the task, and such search can be repeated by the selected neighbor, until no improvement is possible.

To enhance *k-SAP*, we propose the following greedy improvement. After *k-SAP* protocol is finished and (possibly) the best robot is decided, *1*-hop *SAP* can be initiated by the previously decided robot in order to search for better robot in *1*-hop neighborhood. If there is no such robot, no reassignment is done. If better robot exists, it becomes new collecting robot and initiates another *1*-hop *SAP*. The algorithm repeats until no better robot is found in *1*-hop neighborhood.

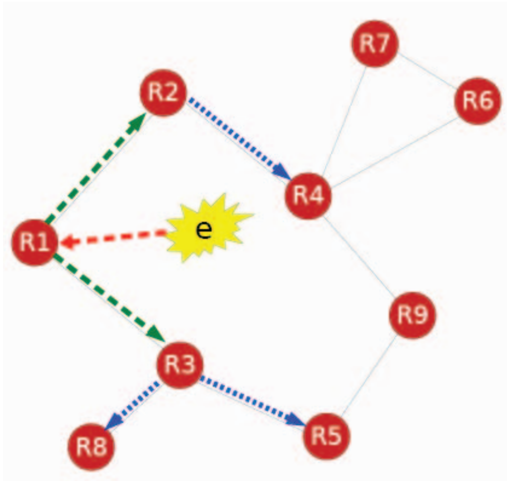

Figure 2. Greedy improvement of auction based protocols

To illustrate *k-SAPG* let's assume that *1-SAP* protocol is used to decide about (possibly) the best robot in scenario in Fig. 2. According to that protocol, collecting robot *R1* would decide that winner is robot *R2* (or *R3* but not both) since it is the closest of the three robots *R1*, *R2* and *R3*. However, there is closer robot (*R4*) that can be found by applying greedy phase. Localized additional *1-SAP* auctions initiated by *R2* or *R3* are illustrated by blue dotted lines. Note that there will be two greedy rounds of auctions. First will be initiated by *R2* and second by *R4*, which will not find any better robot.

Similarly, *k-SAAP* protocol can be enhanced by applying previously explained greedy strategy. It will be designated as *k-SAAPG* protocol.

Both protocols, *k-SAPG* and *k-SAAPG*, can be formalized as:

---

1. Find (likely) the best robot R_{PB} using *k-SAP* or *k-SAAP*
2. Initiate new *1-SAP* by R_{PB}
3. Find new R_{PB}
4. Repeat steps 2 – 4 until no better robot is found
5. The winner is the last R_{PB}

Note that greedy strategy cannot improve over *SAP* and *SAAP* since both protocols include all robots and the best robot is always found.

IV. Simulation Results

We used MIN-DPA algorithm [14] to generate connected pseudo random unit graphs that represent robot networks. This algorithm aims to distribute node degrees (number of neighbors in UDG) more uniformly while maintaining connectivity, and is very fast (especially for 'sparse' networks) compared to typical algorithm used in literature.

There is always one event at a time that needs response from robots. 2D space being monitored is a square 100m x 100m, and there are *n*=10, 20, 50 or 100 robots. Average

Authorized licensed use limited to: King Fahd University of Petroleum and Minerals. Downloaded on February 26,2010 at 01:20:20 EST from IEEE Xplore. Restrictions apply.

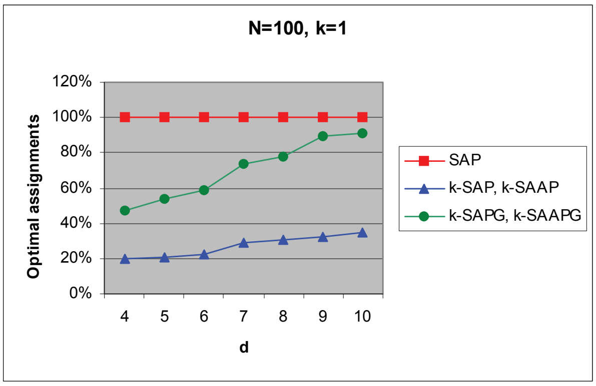

Figure 3. Percentage of optimal assignments as function of *d*

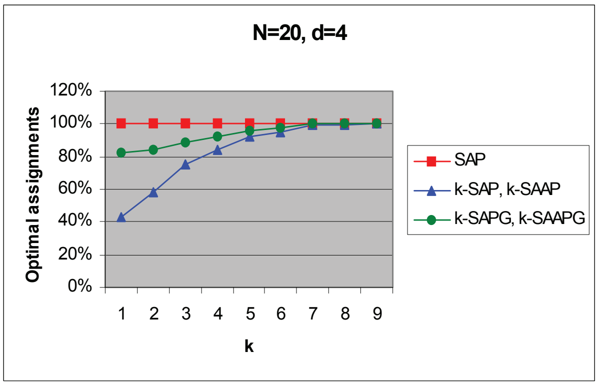

Figure 4. Percentage of optimal assignments as function of *k*

number of robot neighbors is *d*=4,5,6,7,8,9 or 10. Robots bid only according to their distance from event. We measured the number of messages per robot, the percentage of optimal (closest robot) assignments, and the average ratio of distance of selected robot to distance of closest robot.

Robot collector can be chosen in several ways. Obviously the comparison and measured metrics greatly depend on the relative distance of selected collector with respect to the closest collector, as provided by particular event reporting mechanism. To avoid dependence on particular anycasting or other algorithm, we used the following randomized selection for our experiments. Collector is chosen probabilistically, according to their distances from the event. Let d_i be the distance of robot i from the event, and let $D = 1/d_1 + 1/d_2 + \ldots + 1/d_n$. The probability of selecting robot i as the collector is then $1/(D^*d_i)$.

Ten graphs were generated, and 100 random events were selected for each. We compared algorithms described in section III: *k-SAPG* and *k-SAAPG* with *k-SAP* and *k-SAAP* for different values of *k* and *d*. In all protocols, the shortest path algorithm was used to report from each robot to collector (that is, hop distance is counted in simulations). For reference, values for *SAP* protocol are also given in all figures.

SAP protocol always finds optimal robot (Fig. 3). However, to make the assignment, it takes more messages per robot than for any other protocol. The denser the network, the better SAP performs in terms of communication overhead (Fig. 5). Simulations showed that for *k*=1, *d*=4-10, *N*=20,50, 100, for *k-SAP* and *k-SAAP* protocols, percentage of optimal assignments is in the range between 20% and 70%. Only for small and dense networks it can achieve 100% or for *k*>8 (Fig. 4). Percentage of optimal assignments for *k-SAPG* and *k-SAAPG* is more than twice better for the same values of independent variables (Fig. 3 and Fig. 4). *k-SAPG* and *k-SAAPG* feature best performance for *k*=1 and for large and dense networks (e.g. for *N=100, d=9* these protocols are 2.8 times better).

For small networks (*N*≤10), all protocols show similar behavior in terms of average message per robot needed to make the assignment (around 2 messages per robot for *k-SAP* and *k-SAAP* or 3 messages per robot for *k-SAPG* and *k-SAAPG* is needed). For larger networks (*N*>10), *k-SAP* and especially *k-SAAP* features communication overhead reduction compared to *SAP* (Fig. 5) for up to 200 times. However, having at the same time lower optimal assignment percentage. It could be improved with larger *k*-hop values, but with more messages per robot needed. In such cases *k-SAAP* is up to twice better than *k-SAP* [3].

Applying greedy strategy increase communication overhead but improves optimal assignment percentage. Additional communication overhead is 0.4 message per robot on average (actual values varies from 0.07 up to 1.01). The larger the network (but not denser) for low *k*, the additional communication overhead is lower (Fig. 5 and Fig. 6). *k-SAAPG* performs up to 3 times better than *k-SAPG* in terms of additional communication overhead.

Because of localized decisions, *k-SAP* and *k-SAAP* are suboptimal and have higher selected to closest distance (SCD) ratio compared to *SAP* which always selects optimal robot. This is also due to the collector

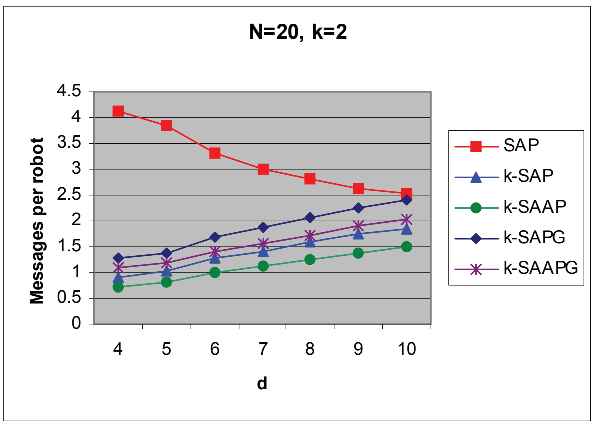

Figure 5. Average number of messages per robot as function of *d*

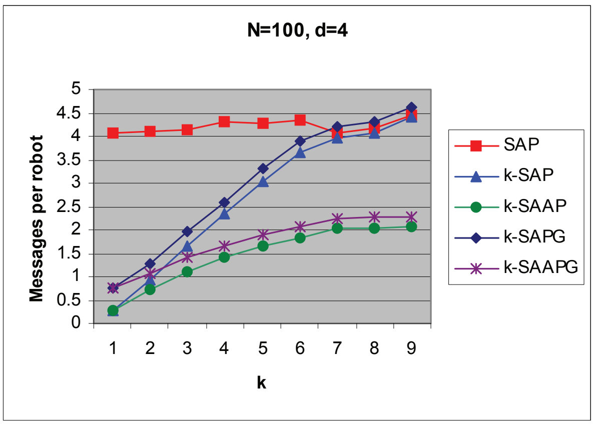

Figure 6. Average number of messages per robot as function of *k*

Authorized licensed use limited to: King Fahd University of Petroleum and Minerals. Downloaded on February 26,2010 at 01:20:20 EST from IEEE Xplore. Restrictions apply.

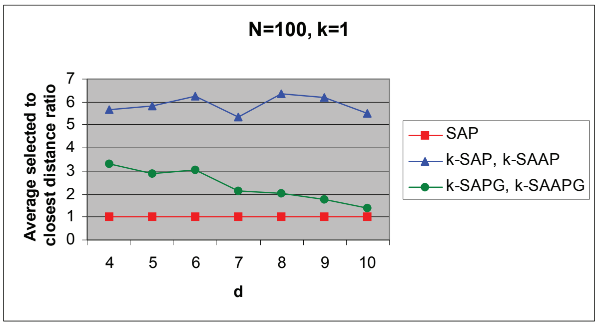

Figure 7. Average selected to closest distance ratio as function of *d*

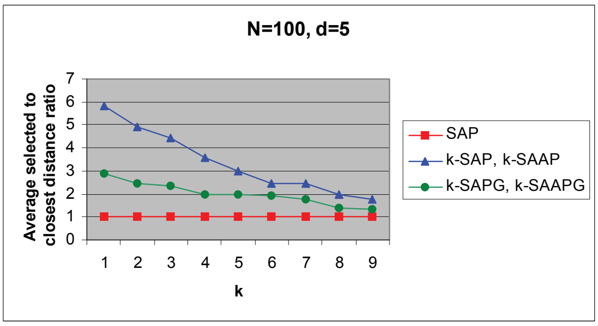

Figure 8. Average selected to closest distance ratio as function of *k*

selection function. For larger *N*, there is higher probability of selecting a distant collector, while the restricted locality (low *k*) prevents from finding a nearby robot to act.

Applying greedy strategy is justified especially for large and dense networks. *k-SAPG* and *k-SAAPG* features up to 4 times better SCD ratio than *k-SAP* and *k-SAAP* (Fig. 7). Fig. 8 shows SCD ratios as function of *k,* for *N=100* and *d=5*.

V. Conclusions

In this paper we presented improvements of previously proposed auction aggregation protocols as a mean to improve suboptimality of localized decisions in robot-robot communication for the task assignment. After initial task assignment is done, assigned robot initiates *1*-hop auction to check if any of its neighbors has lower cost. If so, it can reassign the task, and such greedy search can be repeated by the selected neighbor, until no improvement is possible.

It is shown that *k-SAPG* and *k-SAAPG* features better percentage of optimal assignment and better closest to selected ratio at the cost of reasonable additional communication overhead, especially in large networks and low values of *k* hop.

This paper is the work in progress and there are several possibilities for further improvements. We used distance as cost metrics in bidding phase. Alternatively, residual energy or energy balancing could be used. As communication cost, average number of messages needed to make assignment is used. Alternatively, total power needed for transmission or load balancing could be applied. Another improvement of aggregation protocols is development of versions that look for the best response within time limits.

Acknowledgment

This work is partially financed by the project “Development of systems and instruments for water, oil and gas exploration”, pr. num. TR11006, Serbian Federal Ministry of Science and Environmental protection. The work is partially supported by NSERC Strategic Grant STPGP 336406-07 on sensor and actuator networks.

References

[1] G. Bekey, J. Yuh, “The Status of Robotics,” *IEEE Robotics & Automation Magazine*, vol. 15, issue 1, pp. 80–86, 2008.

[2] T. Melodia, D. Pompili, V.C. Gungor, I.F. Akyildiz, “Communication and Coordination in Wireless Sensor and Actor Networks,” *IEEE Transactions on Mobile Computing*, vol. 6, issue 10, pp. 1116–1129, 2007.

[3] I. Mezei, V. Malbasa and I. Stojmenovic, “Auction Aggregation Protocols for Wireless Robot-Robot Coordination,” 8^{th} International conference on Ad Hoc Networks and Wireless, Murcia, Spain, 22-25, Sep. 2009, accepted for publication.

[4] C. Tovey, M.G. Lagoudakis, S. Jain, S. Koenig, “The Generation of Bidding Rules for Auction-based Robot Coordination,” in *Multi-Robot Systems: From Swarms to Intelligent Automata*, Vol. III, Springer, 2005, pp. 3–14

[5] R.G. Smith, “The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver,” *IEEE Transactions on Computers*, Vol. C-29, N.12, pp. 1104–1113, 1981.

[6] B.P. Gerkey, M.J. Mataric, “Sold!: Auction methods for multi-robot coordination,” *IEEE Trans. on Robotics and Automation*, Vol. 18, No. 5, pp. 758–768, 2002.

[7] N. Michael, M. Zavlanos, V. Kumar, G.J. Pappas, “Distributed multi-robot task assignment and formation control,” Proc. IEEE International Conference on Robotics and Automation (ICRA 2008), pp. 128–133, 2008.

[8] M. Nanjanath, M. Gini, “Auctions for task allocation to robots,” Proceedings of the 9th International Conference on Intelligent Autonomous Systems, IOS Press, pp. 550–557, 2006.

[9] S. Wei, D. Lihua, F. Hao, Z. Haiqiang, “Task Allocation for Multi-Robot Cooperative Hunting Behavior Based on Improved Auction Algorithm,” Proceedings of the 27th Chinese Control Conference, pp. 435–440, 2008.

[10] A. Kishimoto, N. Sturtevant, “Optimized Algorithms for Multi-Agent Routing,” Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008), pp. 1585–1588, 2008.

[11] X. Zheng, S. Koenig, C. Tovey, “Improving Sequential Single-Item Auctions,” Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2238–2244, 2006.

[12] M.B. Dias, R. Zlot, N. Kalra, A. Stentz, “Market-Based Multirobot Coordination: A Survey and Analysis,” *Proceedings of the IEEE*, Vol. 94, No. 7, pp. 1257–1270, 2006.

[13] F. Zhang, W. Chen, Y. Xi, “Improving Collaboration through Fusion of Bid Information for Market-based Multi-robot Exploration,” Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005), pp. 1157–1162, 2005.

[14] F. Atay Onat, I. Stojmenovic, H. Yanikomeroglu, “Generating Random Graphs for the Simulation of Wireless Ad Hoc, Actuator, Sensor, and Internet Networks,” *Pervasive and Mobile Computing*, Elsevier, Volume 4, Issue 5, pp. 597–615, 2008.

Authorized licensed use limited to: King Fahd University of Petroleum and Minerals. Downloaded on February 26,2010 at 01:20:20 EST from IEEE Xplore. Restrictions apply.