Market-Based Coordination of UAVs for Time-Constrained Remote Data Collection and Relay

Amir Ajorlou, Abdollah Homaifar, Albert Esterline, Jay G. Moore, and Robert J. Bamberger

Abstract**—** This paper studies the coordination problem for a team of unmanned aerial vehicles (UAVs) that frequently collects data from a set of unmanned ground sensors (UGSs) and delivers it to a ground station (GS). We formulate this problem as a continuous, two-boundary time-constrained version of the multi-traveller salesmen problem. We present a market-based coordination mechanism that uses the concepts of price, revenue, and cost and a sequence of first-price, one-round auctions conducted by the GS to efficiently distribute data collection tasks among team members. This approach promises robustness, adaptation, and graceful degradation in dynamic environments. We have two main contributions. First, we have used time-varying prices to model different levels of importance for the tasks. Secondly, we have modified the way revenue is anticipated so that UAVs may take into account dependencies among tasks by looking ahead. A behavioral parameter is used to adjust this anticipated revenue to problem requirements experimentally. We have shown how this parameter allows UAVs to learn appropriate bidding strategies. This coordination system has been validated in simulation.

Keywords**—** Coordination, market-based, auction, multi-UAV.

I. Introduction

USE of sensor technologies for remote battlefield applications has greatly increased over the last few years. Unattended ground sensors (UGS) can be used to perform various mission tasks even in locations that cannot be penetrated by other means. To exploit applications of remote UGS systems, timely data transmissions back to a control center should be provided. Considering the limited communication range of the UGSs and the typically large distances between them and the control center, a team of cooperating unmanned aerial vehicles (UAVs) can be employed to frequently collect data from the UGSs and to transfer it back to the control center.

In this paper, we focus on the coordination problem of a team of UAVs that is employed to frequently visit a set of remote UGSs, collect data from them, and return to the ground station to deliver the collected data. We consider a two-boundary time constraint on data delivery time. In particular, for any given UGS, the time between two successive data deliveries should stay between a lower and an upper bound. This constraint is imposed by the nature of UGS applications, in which data delivered early may not contain new information (as in reconnaissance and surveillance), and late-delivered data may not be useful anymore and may result in hazardous situations (as in target detection and situation awareness). The coordination system should also satisfy a constraint on energy consumption. This can be operationalized by requiring the coordination method to optimize the average distance travelled per data delivery. Under these conditions, this problem can be formulated as a continuous and time constrained version of the multi-traveller salesmen problem (MTSP).

Multi-agent systems operating in dynamic environments are highly prone to failures of many kinds, and it is crucial for the coordination method to be robust to these failures. This leads to the topic of distributed coordination mechanisms. One promising method of this type is market based coordination, in which frequent auctioning, time limited contracts, and time-dependent prices ensure robustness in the face of the loss of team members and the failures of individuals [1].

Inspired by the TraderBots market-based coordination approach of Dias *et al.* [2] and the original Contract Net Protocol of Smith [3], we propose a distributed negotiation protocol that uses the concepts of cost, revenue, and profit to efficiently distribute available tasks among team members through a sequence of simple first-price, one-round auctions. The desire of each agent to maximize its individual profit can lead to a globally near optimal plan for the entire team provided there are well-defined costs and price functions.

We present two main contributions. To begin with, this is the first market based approach that uses the concept of time-varying price, which can result in a desirable balance between importance and cost. Secondly, we introduce the concept of anticipated revenue functions that enable team members to take into account the dependencies among tasks by looking ahead to the future even when the auction structure allows trading only single tasks at a time. A behavioral parameter, called the aggression rate, is used to adjust the anticipated revenue functions in light of previous performance. We have shown how choosing different aggression rates can lead to different bidding strategies.

II. Background

Several market-based coordination mechanisms for multi-robot task allocation have been developed during the last decade. The M+ [4] architecture, based on a greedy algorithm, was the first market based approach to multi-robot task allocation. MURDOCH [5] is an online task assignment algorithm that assigns a newly created task to the fittest available robot. TraderBots [2] models a multi-robot team as an economy of self-interested agents that try to maximize their individual profits. Agents with enough resources (viz., leaders) can also

Amir Ajorlou, Abdollah Homaifar, and Albert Esterline are with the Autonomous Control and Information Technology Center, North Carolina A&T State University, 1601 Market Street, Greensboro, NC 27411, USA (e-mail: {aajorlou, homaifar, esterlin}@ncat.edu).

Jay G. Moore and Robert J. Bamberger are with the Johns Hopkins University, Applied Physics Laboratory, 11100 Johns Hopkins Rd. , Laurel, MD 20723, USA (e-mail: {Robert.Bamberger, Jay.Moore}@jhuapl.edu).

form sub-teams by making a plan and offering it to other team members to improve efficiency. Given appropriate cost and revenue functions, this method can lead to a near globally optimal allocation. Although TraderBots considers interrelated costs among the tasks, it does not deal with constrained tasks.

Hoplites [6] seems to be the first market based approach to constrained task execution. In Hoplites, agents frequently exchange information on their intended actions and locally select their actions; this is called passive coordination. In case of constraint violation, agents actively propose and bid on joint plans to resolve the constraint. The performance of Hoplites is validated in perimeter sweeping [6] and, more recently, in constrained exploration [7], where robots should maintain communication with the base station during exploration of a hazardous area. Lemaire *et al.* [8] puts soft time constraints on subtasks of a complex task to synchronize subtask execution. Then the cost of a plan is defined to be the sum of the distance cost of the plan and a cost term corresponding to the quality of the time-constraint satisfaction for the tasks in the plan. Agents, therefore, will try to reduce the deviation from the expected execution time while trading tasks.

In generic market based task allocation systems, the priorities of tasks are determined by the tasks' costs. The lower the cost, the more demand to perform it. It is clear, however, that in some situations we are willing to give higher priority to an important task even though it has a high execution cost compared to the other tasks. Parker's ALLIANCE [9], which is not market based, is able to model task importance levels by assigning different impatience increments to different (robot,task) pairs. Since a task will be undertaken by the first available robot whose impatience for it exceeds a prespecified threshold, tasks will be attempted in the order of their importance.

III. Problem Definition

Consider a set of UGSs deployed in a remote area. A team of UAVs is employed to frequently visit these sensors, collect their data, and deliver it to the ground station (GS). We are looking for a continuous task allocation mechanism that satisfies the constraint

$$_L \leq {}^j_{i+1} \quad {}^j_i \leq {}_H \quad \in \mathbb{N}, 1 \leq \quad \leq$$

on data delivery time while optimizing the team average for the distance travelled per data delivery. ${}^j_i$ denotes the th data delivery time for the th sensor, and $_L$ and $_H$ are, respectively, the lower and upper bounds on the time between two successive data deliveries for any sensor.

To express the form of a solution, we define *sensor-visit* tasks. Each sensor-visit task consists of visiting a sensor, collecting its data and returning to the GS to deliver the collected data. At the beginning of the mission, the GS creates sensor-visit tasks corresponding to the UGSs and assigns them to UAVs via negotiation. Upon receiving a sensor's data, the GS renews the corresponding task and reassigns it to some team member. We refer to the time when a task has been created or renewed as the task's *creation time*. Also, we define the *data delivery time* of a task to be the time passed from the task's creation time to the time when that task's data is received by the GS. The above problem statement can be reformulated using the terms just introduced as the problem of continuously allocating sensor-visit tasks to UAVs such that the data delivery times of all these tasks fall within the time interval $[\ _L, \ _H]$.

IV. Approaches and Implementation

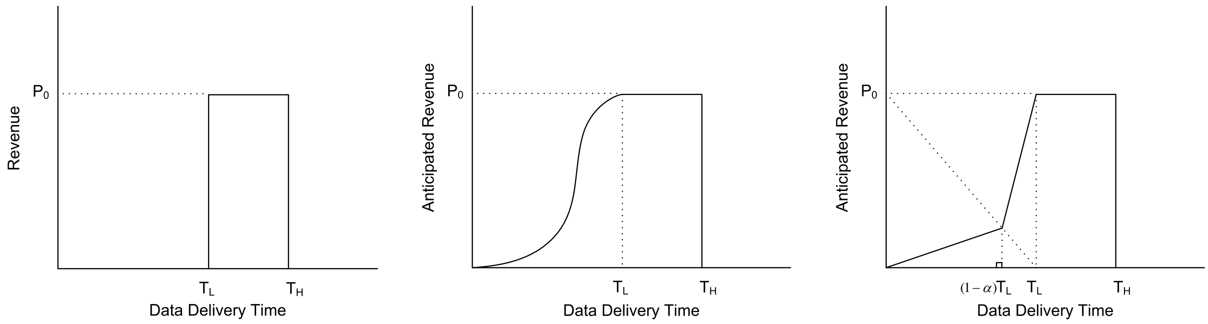
We approach this hard time-constrained task allocation problem using a market based coordination mechanism. The sensor-visit tasks created at the start of the mission are assigned by the GS to UAVs via a sequence of first-price single-task auctions. A UAV inserts a task awarded to it into its current plan, where the task remains until the UAV delivers the corresponding data to the GS. Upon receiving a task's data, the GS renews the corresponding task and reassigns it to some team member by a new auction. Below, we describe the process of auctioning, bidding and clearing in detail.

At each time step, the GS offers all unassigned tasks to the UAVs by means of a broadcast message. The sensor locations, the task creation times, and the prices of offered tasks are contained in this message. The price of a sensor-visit task determines the amount of virtual money that will be given to a UAV that completes the task within the time window. On receiving the message from the GS, each UAV computes the marginal cost of adding each offered task to its current plan (as will be explained immediately below). A UAV then submits a bid on the task most profitable for it. The profit earned by a UAV for a task is the difference between the UAV's cost for doing the task and the price (determined by the GS) of the task. The bid message returned by a UAV should indicate the task and the corresponding profit. The GS evaluates all bids received and finds the one showing the most profit. It then assigns the task in this bid to the UAV that submitted the bid. The GS repeats the same process for the remaining tasks at the next time step.

In our problem, two types of cost contribute to the marginal cost of adding a new task to a plan:

1) the cost due to the additional distance that the UAV should travel—the *distance cost*—and
2) the cost due to the latency that performing this task will cause in the data delivery time of the other tasks already in the plan—the *time cost*.

We consider the distance cost of adding a new task $_{new}$ to a plan $_{cur}$ as $_{dist}(\ _{cur}, \ _{new}) = \ _c$, where $_c$ is the unit distance cost and is the marginal distance that a UAV must travel when adding $_{new}$ to $_{cur}$. Incorporating this cost term in the bid evaluation results in a hill climbing approach to minimizing the sum of the distances travelled by the UAVs since it assigns an available task to a UAV with the lowest distance cost. Adding a task $_{new}$ to a Plan $_{cur}$ also increases the data delivery time of the tasks already in the plan and thereby may change the revenue associated with these tasks as the revenue depends on the data delivery time. A UAV can use the revenue function , shown in Fig. 1a, to calculate the revenue it will earn by completing the tasks already in its plan. In the same way, it can calculate the revenue from



(a) (b) (c)

Fig. 1. Revenue and Anticipated revenue functions. Shown are, (a) the original revenue function, (b) a typical anticipated revenue function, and (c) the α-aggressive anticipated revenue function. Note that anticipated revenue function may take non-zero values before T_L.

completing these same tasks after accepting $_{new}$. The time cost is the difference between these two revenue values.

Using this revenue function , however, may not result in a good bidding strategy. This is because the data delivery time of an offered task depends on future tasks that a UAV may win, but does not account for this dependency. To address this issue, we introduce an extension to the revenue function, which we call the *anticipated revenue function*, $_A$. The value $_A( )$, where is the data delivery time of an offered task based on the current plan, reflects the average revenue that a UAV expects to earn by adding that task to its plan and performing it. Using this modified revenue function, when a UAV evaluates its bids, it considers not only the tasks already in its plan but also the average length that its final plan may have. Therefore, $_A$ may take on a non-zero value for a task that would be delivered before $_L$ according to the UAV's current plan (see Fig. 1b). Instead of the revenue function , a UAV will use its anticipated revenue function $_A$ for bid evaluation. The time cost of adding $_{new}$ to $_{cur}$ can thus be formulated as

$$
\begin{aligned}
{}_{time}(\ {}_{cur}, \ {}_{new}) \ = \ & \sum_{T\in P_{cur}} {}_A(\ {}_{P_{cur}} \quad {}_T) \\
& \sum_{T\in P_{cur}} {}_A(\ {}_{P_{cur}} + \Delta \quad {}_T)
\end{aligned}
$$

where $_{P_{cur}}$ is the time at which the UAV expects to finish its current plan, $_T$ is the creation time of task , and Δ is the marginal time that accepting $_{new}$ will add to the completion time of the current plan.

By including the time cost into the marginal cost, a UAV takes into account the time constraints while evaluating its bids. So, to cover both minimization of the sum of the distances travelled by the UAVs and the time constraints on data deliveries, we take the marginal cost to be the sum of the time cost and the distance cost:

$$
\begin{aligned}
(\ {}_{cur}, \ {}_{new}) \ = \ & {}_{dist}(\ {}_{cur}, \ {}_{new}) + \\
& {}_{time}(\ {}_{cur}, \ {}_{new})
\end{aligned}
$$

Calculating the marginal cost itself is NP hard since it requires replanning for the new set of tasks. For simplicity, we use a heuristic in which we insert the new task in all possible positions in the current plan and choose the one that minimizes the distance cost of the new plan.

Turning to the communication complexity of our coordination approach, since only the most profitable task is sold in each auction, each UAV needs to submit only its most profitable bid. This results in a light communication load, with ( ) messages per auction (where is the number of UAVs).

A. Cautious Bidding

The most straightforward method for bidding is simply to use the original revenue function as the anticipated revenue function in bid evaluation:

$$ {}_A(\ ) = \begin{cases} \ {}_0 & {}_L \qquad {}_H \\ 0 & \text{otherwise} \end{cases} $$

where is the time passed from the task creation time.

Following the cautious bidding strategy, a UAV just considers its current plan in evaluating its bid on an offered task . If the delivery time of based on the current plan is before $_L$, it will not bid on it and will not take the risk of accepting in the hope that tasks it may win in the future will push the delivery time of into the desired window.

This cautious behavior may result in frequent task timeouts. To illustrate, consider the situation where one or more UAVs is loitering above the GS and a set of tasks that have just been created or renewed is offered to them. If $_{max}$ is the time required to visit the farthest sensor and come back to the GS and if $_{max}$ $_L$, then nobody will bid on any of these tasks until $_L$ $_{max}$ time has elapsed since their creation. After that, the task associated with the farthest sensor will be assigned to one of the UAVs, and this UAV will try to add more tasks that it can perform in the marginal time of $_H$ $_L$ to its plan. Note that other UAVs may still need to wait until the task associated with the next farthest sensor becomes profitable. It is thus very probable that they will not be able to perform all the tasks without a timeout.

B. Aggressive Bidding

At the opposite extreme from cautious bidding, there is aggressive bidding. The anticipated revenue function for this bidding strategy is

$$ {}_A(\ ) = \begin{cases} \ {}_0 & \qquad {}_H \\ 0 & \text{otherwise} \end{cases} $$

where is the time passed from the task creation time. Following this bidding strategy, each UAV assumes that its final plan will be long enough to push its delivery times beyond $_L$. As a result, a UAV considers just the upper bound in its revenue function for bid evaluation. Clearly, this strategy may result in premature data delivery (i.e., delivery before $_L$ time has elapsed since task creation time) because the lower constraint on delivery time is ignored.

To illustrate, consider how this strategy affects the distribution of tasks among team members. Consider the case where one or more UAVs is loitering above the GS and a set of tasks just created or renewed is offered to them. It can be shown that the UAV that wins the first auction will greedily fill its plan until either no more tasks are available or the upper time bound would be exceeded if an additional task were accepted. After that, the other UAVs will repeat the same procedure for the remaining tasks, if any. Since this bidding strategy dictates an order on the UAVs for task allocation, the chance of making a premature delivery is less for the UAVs earlier in this order because of the availability of more tasks for filling up their plans, pushing their tasks' data delivery times beyond their time lower bound. Also, some UAVs toward the end of the order may remain unutilized.

Thus, the order in which UAVs fill up their plans is the main reason here for premature data delivery, particularly by UAVs later in the order. It can be shown that, however, even the earlier UAVs in this order can deliver data prematurely.

C. Time Varying Price

We suggest varying a task's price to model the different levels of importance it may have. Specifically, instead of using a fixed price $_0$, the GS uses a time-increasing function $()$ to motivate UAVs to bid on older tasks. The price of a task offered by the GS at time is $(\quad _T)$, where $_T$ is the creation time of task . Since the price is time varying and the profit is the price minus the cost, the most profitable task clearly is not necessarily the task with the lowest cost. In other words, the presence of time varying prices can result in a balance between task cost and task importance in the process of task allocation. must be great enough so as to overcome the extra cost an expensive important task may have compared to the other offered tasks. Time varying price can improve the performance of the allocation by decreasing the probability of successive timeouts for a given task.

For simplicity, we choose the linear price function

$$( ) = \ _0 \quad (1 + \ _p \quad \frac{}{_H}),$$

where $_0$ is the initial price of the tasks at creation time, $_H$ is the upper bound for data delivery time, and $_p$ determines the increment in the price. Note that the price function is a function known by the auctioneer (GS) used to calculate the price that it announces to the UAVs when offering a task , and the UAVs will use that price as $_0$ in their anticipated revenue functions for that task. Therefore, the revenue that a UAV will receive upon performing a task depends both on the time when it accepted the task and the time when it delivers the data.

D. Anticipated Revenue Learning

Instead of the extreme anticipated revenue functions of the cautious and aggressive bidding strategies, we suggest using a set of anticipated revenue functions that exhibit reasonable levels of risk. We propose a family of functions $\{ \ ^{\alpha}_{A} \ ^{1}_{0}$, as shown in Fig. 1c, that represent different levels of risk acceptability, from the absolutely cautious level of the cautious bidding strategy ($\alpha = 0$) to the absolutely aggressive level of the aggressive bidding strategy ($\alpha = 1$). This family of functions can be formulated as

$$^{\alpha}_{A}( ) = \begin{cases} \frac{\alpha t}{(1-\alpha)T_L} \quad _0 & 0 \quad \leq (1 \quad \alpha) \quad _L \\ (1 \quad \frac{(1-\alpha)(T_L-t)}{\alpha T_L}) \quad _0 & (1 \quad \alpha) \quad _L \quad \leq \quad _L \\ _0 & _L \quad \leq \quad _H \\ 0 & \text{otherwise} \end{cases}$$

where is the time passed from the task creation time. We call the parameter α the *aggression rate*, a UAV that uses $^{\alpha}_{A}$ as its anticipated revenue function is said to be an α*-aggressive UAV*, and the induced bidding strategy is called α*-aggressive bidding*. Clearly, $^{0}_{A}$ and $^{1}_{A}$ are the anticipated revenue functions of the cautious and aggressive bidding strategies.

Considering that our problem is a long-term application, we employ a learning algorithm to adjust the aggression rates of UAVs by experiment. The algorithm for adjusting α values is as follows: we initially set all the α values to $1 \ 2$ since no a priori knowledge of the environment is assumed. The GS uses two messages, which we call and , to inform the UAVs about their performance. *ComeOn* is a broadcast message that informs the UAVs that the team's overall performance is so cautious that some tasks are timed out. On receiving a message, each UAV increases its aggression rate by a certain amount in order to change team behavior to be more aggressive in an attempt to prevent later timeouts. is a unicast message sent to a UAV that has delivered data prematurely. A premature delivery implies that the UAV that delivered the data should decrease its aggression rate to prevent future premature deliveries. Thus, upon receiving a message, the UAV decreases its aggression rate by a certain amount.

A UAV uses a time-varying step size of $\frac{\alpha}{}$, where is the number of the control messages it has received, and $_\alpha$ (which is less than 1) is chosen to be rather large, giving large steps at the beginning to allow the UAV to settle rapidly around an appropriate aggression rate. A decreasing step size avoids overreaction to temporarily bad performance so that a UAV changes its aggression rate significantly only if its local performance is consistently bad. It is clear that, using this learning algorithm, UAVs may end up with different aggression rates, which induce different bidding strategies for different team members.

E. ARL+TVP

We can use time-varying price together with anticipated reward learning. We expect that time-varying price will result in a more reasonable performance by preventing a large number of successive timeouts for a given sensor while UAVs

TABLE I

PERCENTAGE OF TIMEOUTS, PREMATURE DELIVERIES, AND ON-TIME DELIVERIES

Method	% timed out	% premature	% on time
Cautious	41	0	59
Aggressive	0	28	72
Cautious+TVP	48	0	52
ARL	3	9	88
ARL+TVP	5	12	83

TABLE II

DATA DELIVERY TIME STATISTICS

Method	Longest	Shortest	Mean	STD
Cautious	7145	181	425	682
Aggressive	210	41	175	37
Cautious+TVP	1030	181	352	219
ARL	600	60	195	37
ARL+TVP	450	50	196	37

adapt their aggression rates to the problem. Indeed, this is borne out in simulation.

V. SIMULATION AND ANALYSIS

We discussed five market based coordination methods in the last section. In this section, we study their performance in simulation. The problem under study consists of a team of three UAVs and twelve UGSs randomly selected at different angles and different distances around the GS. The speed of the UAVs is 25 m/s. The closest UGS is 625 m from the GS, which requires 50 s for a round trip. The farthest UGS is 1250 m from the GS, requiring 100 s for a round trip. We choose $_L = 180$ s and $_H = 210$ s. We have set the unit distance cost $_c$ to be \$0.1/m and the initial price $_0$ to be \$300. This price is chosen to be higher than the distance cost of travelling to any sensor and back to the GS, which is $2500 \quad \$0\ 1 \quad = \$250$ since the round trip to the farthest sensor is 2500 m. We set $_\alpha$ to 0.1. Also, $_p$ is 1, implying that, in the TVP method, the price of a task will double after $_H$ time has passed from its creation time if it is still not sold.

Under this setting, we conducted five experiments, each with a different method. Each experiment was long enough to exhibit long-run system performance. Table I shows the percentage of timeouts, premature deliveries, and on-time deliveries in the five experiments. Table II shows data delivery statistics for the five experiments.

From these tables, it can be seen that, with the cautious bidding strategy, we have a high percentage of timeouts since a UAV has only $_H \quad _L = 30$ s marginal time to add other tasks to its plan after accepting its first task.

Another disadvantage of the cautious bidding strategy is the high number of successive timeouts there may be for a sensor. Table II shows that the longest data delivery for the cautious method is 7145 s, indicating 33 successive timeouts for at least one of the sensors. As the results show, adding TVP to the cautious method significantly decreased the longest data delivery time hence the number of successive timeouts, as claimed in Section IV-C. Adding TVP also greatly reduced the mean and standard deviation of data deliveries by shortening long deliveries. Adding TVP to the cautious method, however, has the drawback of slightly increasing the number of timeouts.

The aggressive bidding strategy avoids timeouts but allows a moderate percentage of premature deliveries. It, however, has fewer premature deliveries than the cautious method has timeouts. This is because, with the cautious method, all UAVs contribute to timeouts while, with the aggressive method, the main contribution to premature deliveries is made by the UAVs at lower orders as described in Section IV-B.

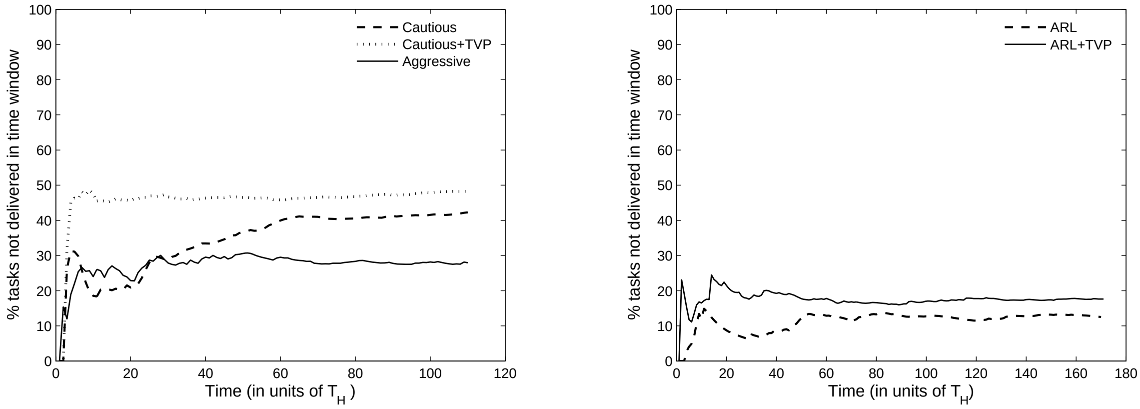
As can be seen from Fig. 2 and table I, both methods using anticipated revenue learning have significantly decreased the number of deliveries not within the time window. These methods achieved this decrease by adapting the team members' behaviors to the problem through learning appropriate aggression rates. Interestingly, these two methods have pushed the mean of the delivery times right into the middle of the $[\ _L,\ _H]$ window. Compared with ARL, ARL+TVP improved the longest delivery time by 25 percent, as we expected, at the cost of a slight increment in the number of premature deliveries and timeouts. They both have the same standard deviation, however, because, although TVP has pushed deliveries long after $_H$ toward the mean, it has more deliveries after $_H$.

All five approaches stabilized their performance quite rapidly, as can be seen in Fig. 2. The only exception may seem to be the cautious method. In fact, even this method has stabilized its local performance by $25\ _H$. But, since its stabilized performance is much worse than its overall performance before stabilizing, the transition time to move the overall performance to the stabilized local performance is quite long. It can also be seen from Fig. 2 that the performances of ARL and ARL+TVP during learning are very good. This is important since online learning should have good performance even before convergence.

In both ARL and ARL+TVP, the UAVs have found their roles quite rapidly. ARL ended up with a team of 0.3-aggressive, 0.65-aggressive, and 0.7-aggressive UAVs, and ARL+TVP ended up with a team of 0.28-aggressive, 0.52-aggressive, and 0.6-aggressive UAVs. It is notable that the aggression rates with ARL are less than those with ARL+TVP as the latter provides additional motivation to take tasks by increasing their prices.

VI. CONCLUSION

In this paper, we studied the problem of collecting data from a set of unmanned ground sensors (UGSs) using a team of cooperating unmanned aerial vehicles (UAVs) and delivering the data to a ground station (GS). This problem was formalized as a continuous, two-boundary time-constrained version of the multi-traveller salesmen problem. We implemented a market-based coordination approach that uses the concepts of price, revenue, and cost to assign sensor-visit tasks to team members through a sequence of first-price, one-round auctions conducted by the GS. The data delivery time of an offered task is determined by the time-length of the UAV's final plan, which depends on the tasks it may win in the future. To handle these



(a) (b)

Fig. 2. Percentage of the tasks not delivered within the time window $[T_L, T_H]$ in (a) Cautious, Aggressive, and Cautious+TVP, and (b) ARL, and ARL+TVP. The graphs show the cumulative performance from the beginning of each experiment.

dependencies, we introduced anticipated revenue functions as a variation of our original revenue functions to enable a UAV to estimate the expected time-length of its final plan based on its current plan length. We approximate this function by a family of functions represented by a single parameter, the *aggression rate*, and learn an appropriate member of this class by adjusting this parameter experimentally; we refer to this as *anticipated reward learning* (ARL). Learning allows the UAVs to adapt to problem requirements and to other team members' activities so that they may easily recover from other members' failures, accommodate team modifications, and gracefully degrade while some members malfunction.

Simulations have shown good performance for this method. The mean data delivery time after learning was pushed exactly to the middle of the desired time window with an acceptable standard deviation. Adding time-varying prices (TVP) significantly shortened the longest timeout by providing more revenue for tasks that are more liable to timeout. The initial values of aggression rates influence the performance of the system during learning. We initialized all aggression rates to $1\ 2$, assuming no a priori knowledge about the environment. Where some a priori information is available, however, one can try to find better initial values. If, in addition, the problem layout is known in advance, we can run the program offline and start the real application with the learned values.

Future work will include extending anticipated revenue functions to other domains with dependencies among tasks. Also, one could try approximating the anticipated revenue function with other families of functions. Furthermore, we could use a learning method for the creation-time price, which, with TVP, can affect the rate of change in importance, and, with ARL, can affect the team members' aggression rates.

REFERENCES

[1] M. Dias, R. Zlot, M. Zinck, and A. Stentz, "Robust multirobot coordination in dynamic environments," in *Proceedings of the International Conference on Robotics and Automation*, 2004.

[2] M. B. Dias and A. T. Stentz, "Traderbots: A market-based approach for resource, role, and task allocation in multirobot coordination," Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI -TR-03-19, August 2003.

[3] R. G. Smith, "The contract net protocol: High-level communication and control in a distributed problem solver," *IEEE Transactions on Computers*, vol. 29, no. 12, 1980.

[4] S. S. da Costa Botelho and R. Alami, "M+: A scheme for multi-robot cooperation through negotiated task allocation and achievement." in *Proceedings of the International Conference on Robotics and Automation*, 1999.

[5] B. P. Gerkey and M. J. Matarić, "Sold!: Auction methods for multi-robot coordination," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 758–768, Oct. 2002.

[6] N. Kalra, D. Ferguson, and A. T. Stentz, "Hoplites: A market-based framework for planned tight coordination in multirobot teams," in *Proceedings of the International Conference on Robotics and Automation*, 2005.

[7] ——, "Constrained exploration for studies in multirobot coordination," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006.

[8] T. Lemaire, R. Alami, and S. Lacroix, "A distributed tasks allocation scheme in multi-uav context," in *Proceedings of the International Conference on Robotics and Automation*, 2004.

[9] L. Parker, "Alliance: An architecture for fault-tolerant multi-robot cooperation," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 220–240, 1998.