

Dynamic Re-Planning in Multi-Agent Networks

I. PROBLEM AND ITS IMPORTANCE

THE use of heterogeneous multi-agent robotic systems can be used to solve a number of problems in military and industrial applications. For example, unmanned underwater vehicles (UUV) can collaborate with unmanned aerial vehicles (UAV) to search for hostile ships. When the AUVs and UUVs detect a hostile ship, they can communicate the necessary information to an unmanned surface vehicle (USV) to intercept the ship. In another application, a heterogeneous network of robotic systems collaborate to assemble an automobile in a factory. However, the issue of delegating tasks and mission objectives to individual robotic systems becomes daunting and does not scale well as the number of agents in the network increases. The mission designer has to take into account agent capabilities, mission constraints, and mission objectives. Another major issue is fault control of the system. In order to design a robust multi-agent network, the mission designer would have to provide cases for every possible failure during the mission. This is not feasible. A possible solution to this problem is the use of the Contract Net Protocol (CNP) architecture, which allows for dynamic re-planning and allocation of resources through group auctions [1].

II. RELATED WORK

Starting with the seminal paper on CNP [1], a number of researchers implemented CNP for robotic systems. In [2], researchers implemented a collaborative network of robots with the use of CNP and a publish and subscribe network. Other researchers, such as Arkin [3], developed software applications that simplified the use of CNP for non-researchers and military personnel. Another group of researchers concerned with re-planning of missions did not use CNP, but instead attempted to solve the Multiple Traveling Salesman Problem (MTSP) [4]. However, their approach was costly in terms of computation compared to CNP. In [5], the researchers explored the use of both centralized and distributed methods for multi-agent network planning. Although, the distributed aspects of the design were merely the execution of the central planner results. Balch and Sariel evaluated the use of various heuristics within the auction-based planner paradigm to solve the MTSP [6]. A major issue with the previous research is that none of the auction-based mission planners described in the literature were truly decentralized or distributed. Each architecture relied upon a master robot calling for an auction from the other robots and communicating the results to the other agents.

III. APPROACH

The Contract Net Protocol (CNP) dynamic re-planning algorithm was designed within the architecture provided by the Mission Oriented Operating Suite (MOOS). MOOS is a publish and subscribe database architecture that is common

within the Autonomous Underwater Vehicle (AUV) community. The publish and subscribe communications allow for individual users to push information to the database that other applications can access. This architecture also reduces the probability of a single application hijacking the system due to poor programming failures. Researchers at the Massachusetts Institute of Technology (MIT) developed an extension to MOOS called MOOS-IvP, where the ‘‘IvP’’ stands for Interval-Programming. MOOS-IvP uses discretized objective functions to efficiently blend individual robotic behaviors for an optimal output behavior. This research made use of MOOS-IvP’s waypoint and loitering behaviors. The design of a typical single-vehicle MOOS system usually consists of several applications processing sensor inputs, controlling actuators, and controlling high-level planning. All of the applications share a common database called the MOOSDB. However, in this multi-vehicle MOOS system, each vehicle possessed its own MOOSDB with associated applications and the MOOSDBs on each of the vehicles communicated via the MOOSBridge application. While the applications within a single MOOSDB can access any of the data in the database, in a multi-vehicle system, the MOOSBridge has to be configured to only permit the passing of specific messages between the MOOSDBs on each of the vehicles. The multi-vehicle configuration with multiple MOOSDBs is depicted in Figure 1.

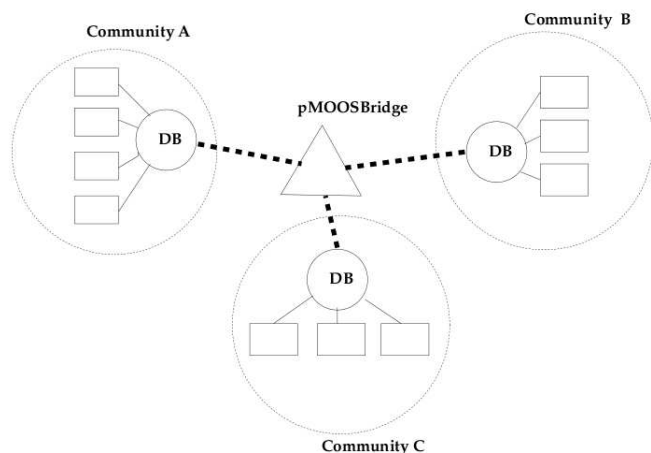


Figure 1. MOOSBridge connecting three MOOS communities [7]

In this implementation, the only messages that were passed between the MOOSDBs were specific command and response messages between the master auctioning vehicle and the vehicles that were bidding on tasks.

Auction-based planning was accomplished through an automated auctioneer-to-bidder process in which the auctioneer requested bids to complete a specific task from each of the bidding agents. The bidders generated bids based on the

expected cost that each agent calculated it would take to accomplish the task. After the auctioneer received all necessary bids, the auctioneer designated a winner of the task, at which point, the winning bidder was commanded to perform the task. The auction-based planning was accomplished through the development of the pAuctioneer and pBidder MOOS applications. Each of the vehicles in the system instantiated both the pAuctioneer and pBidder applications, but only the master vehicle enabled the pAuctioneer functionality via the MOOS mission file. However, if necessary, such as a master vehicle failure, the role of auctioneer could be shifted to another vehicle. The purpose of the pAuctioneer and pBidder applications was to efficiently distribute the resources of the vehicles amongst the available tasks. In order to accomplish this, the pAuctioneer and pBidder had a specific command and response protocol that began with the master auctioneer identifying a task and then requesting bids from each of the slave vehicles for that task. In this implementation, the bids were directly related to the vehicle's cost of completing the task. Since the tasks consisted of exploring a way-point, the cost to complete a task was the straight line distance between the vehicle's position at bid request time and the position of the way-point. After a vehicle calculated its costs for exploring a way-point, it responded to the bid with its calculated cost. The pAuctioneer gathered bids from each of the vehicles and sent an EXECUTE_BID command to the vehicle with the lowest bid. The end of the auction was defined by a predetermined time limit. This allowed for scalability in the number of vehicles in the AUV network. After the winning bidder received the EXECUTE_BID command, the vehicle modified the MOOS-IvP objective function to redirect itself to the location that needed to be explored. When the vehicle finally arrived at the location, it notified the master auctioneer that it had successfully completed the task. The complete auctioneer and bidder message protocol is shown in Figure 2.

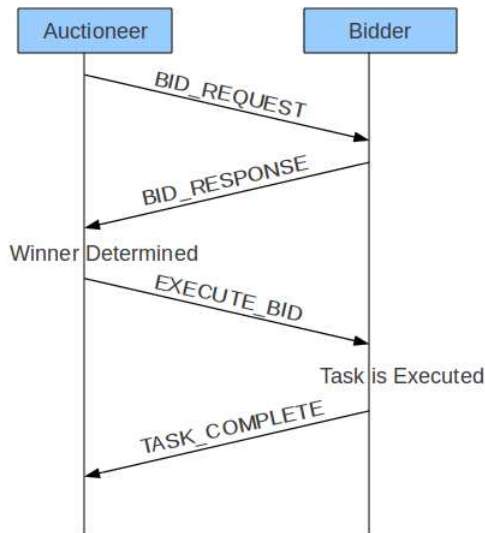


Figure 2. Auctioneer and bidder command and response

The auctioneer made use of a list of currently open auctions,

defined in the AreaOfInterest C++ class, so that it could process multiple auctions concurrently. The primary key for each auction was a label designator that was associated with the time at which the task was first identified. The bidders also maintained a list of AreaOfInterests, so that each bidder could add awarded tasks to a queue for future processing. After the auction-based planning was successfully implemented, a test structure was developed to run the auction-based planning through multiple iterations of simulation. The control for the experiment was a planning system that randomly assigned tasks to each of the available vehicles.

IV. EVALUATION OF RESULTS

In this experiment, the metric that was of paramount importance was the total distance traveled by the AUV network. The auction-based planning system was compared to the random assignment-based planning system in a series of simulations. The simulations were carried out using the MarineViewer that is integrated with MOOS and BASH scripts that controlled the multiple simulation runs. In each run of the simulation, three AUVs were instantiated and then a script randomly placed eight areas of interest throughout the map. The auction-based system attempted to visit each location using costs and bids, while the control algorithm used random assignments. Each simulation run completed when all eight areas of interest were explored. The distance traveled by each AUV was recorded to a log file. A screen capture of the auction-based simulation in progress is shown in Figure 3.

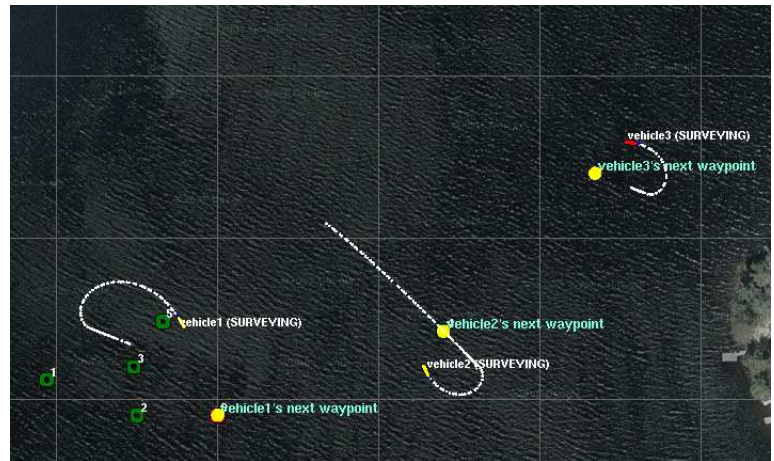


Figure 3. Three AUVs using auction-based planning

In the screen capture, red locations designate areas that have been assigned and green locations designate areas that have already been explored. A yellow dot designates the vehicle's next way-point.

The distances traveled, in meters, for the random assignment planner are shown in Table I and the results from the auction-based planning are shown in Table II.

Both the random assignment planner and the auction-based planner successfully visited each area of interest. However, the vehicles in the auction-based planner completed their tasks while traversing less distance than the random assignment planner. The AUVs in the auction-based planner traveled a

Table I
DISTANCES TRAVELED BY VEHICLES IN SIMULATION USING RANDOM
ASSIGNMENT PLANNING

Sim Run	Vehicle 1	Vehicle 2	Vehicle 3
1	94	510	413
2	237	499	224
3	67	405	220
4	144	454	384
5	174	529	355
6	208	401	525
7	56	612	342
8	99	422	542
9	146	445	514
10	182	378	461
Sub Totals	1,408	4,656	3,981
Total	10,045 meters		

Table II
DISTANCES TRAVELED BY VEHICLES IN SIMULATION USING
AUCTION-BASED PLANNING

Sim Run	Vehicle 1	Vehicle 2	Vehicle 3
1	871	228	52
2	655	251	28
3	492	142	208
4	128	357	163
5	337	179	109
6	384	178	93
7	201	708	298
8	300	99	164
9	207	365	164
10	493	72	240
Sub Totals	4,069	2,579	1,519
Total	8,167 meters		

total 8,167 meters during the ten simulation runs, while the AUVs in the random assignment planner traveled 10,045 meters. In the auction-based planner, vehicle number one consistently traveled farther than the other two vehicles.

V. DISCUSSION

The auction-based planner outperformed the random assignment planner in the total distance traveled metric. The main reason for this was due to the fact that the auction-based planner assigned vehicles to areas of interest based on proximity. However, the random assignment planner did not take proximity to areas of interest into account. Thus, the random assignment planner might command a vehicle to travel across the map to explore an area of interest while another vehicle could be right next to that same area of interest. Also, the auction-based planner visited groups of areas of interest that were close together. This significantly decreased the distance traveled by the AUV because it could explore several areas of interest in quick succession.

It is interesting to note that vehicle number one traveled consistently farther than the other two vehicles in the auction-based planner. This was mainly due to how the simulation was initialized. All of the vehicles were initialized to the same location at simulation start time and when a new area of interest was generated, if all costs were equal, vehicle number one was awarded the auction. Thus, vehicle number one consistently was awarded auctions at the first generation of tasks. This will be improved in further iterations of the

simulation by implementing an avoid vehicle behavior and initializing the vehicles to different locations.

One of the main drawbacks of this implementation of the auction-based planner was that each bidder added new areas of interest to the end of its list of tasks. Thus, with dynamic task generation, a new area of interest could spawn between two areas of interest already in the queue, but the AUV would service the middle area last. It would be more efficient for the AUV to service the newly generated area in between servicing the already defined areas that surround it. In order to accomplish this, the next iteration of the auction-based planner will have to be able to sort the task list, so that the AUV can service newly generated areas that are in the current planned path. The generation of new tasks while executing previous tasks suggests a dynamic traveling salesman problem.

The author learned a great deal about publish and subscribe database architectures while implementing this design. While the system is simple to use, it does suggest quite a bit of overhead with communications, especially when embedded systems are considered. The large overhead in communications will be extremely important when this is implemented on an actual AUV where acoustic communications are severely bandwidth limited.

The MOOS simulator can be improved with the addition of random vehicle and communication failures. The auctioneer and bidder communication protocol will have to be more robust to handle these types of failures. For example, the auctioneer will have to check on the progress of tasks after they have been awarded in case a vehicle has failed. Also, the calculation of costs could be improved by taking into account the task list size. The network might better distribute its resources if vehicles with a large task list add a higher cost to its future bids until the task list size decreases. Water currents could also be incorporated into cost calculation. If a vehicle has to travel directly against a strong current to reach a goal, the calculated cost should be high. There are a number of future improvements that will be implemented.

REFERENCES

- [1] R. Davis and R. Smith, "Negotiation as a metaphor for distributed problem solving," *Communication in Multiagent Systems. Agent Communication Languages and Conversation Policies*, pp. 51–97, 1980.
- [2] B. Gerkey and M. Mataric, "Sold!: Auction methods for multirobot coordination," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 5, pp. 758–768, 2002.
- [3] P. Ulam, Y. Endo, A. Wagner, and R. Arkin, "Integrated mission specification and task allocation for robot teams-design and implementation," *Robotics and Automation, 2007 IEEE International Conference on*, pp. 4428–4435, 2007.
- [4] B. Brumitt and A. Stentz, "Grammps: A generalized mission planner for multiple mobile robots in unstructured environments," *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 1, pp. 1564–1571, 1998.
- [5] C. Le Pape, "A combination of centralized and distributed methods for multi-agent planning and scheduling," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*. IEEE, 1990, pp. 488–493.
- [6] S. Sariel and T. Balch, "Real time auction based allocation of tasks for multi-robot exploration problem in dynamic environments," *Proc. AAAI Workshop: Integrating Planning Into Scheduling*, pp. 27–33, 2005.
- [7] P. Newman, "Bridging communities with pmoosbridge," 2009.