Distributed Multi-AUV Cooperation Methods for Underwater Archaeology

Nikolaos Tsiogkas*, Zeyn Saigol*, David Lane*
*Ocean Systems Laboratory
Heriot Watt University, EH14 4AS Edinburgh, Scotland, UK
Email: nt95;d.m.lane@hw.ac.uk

***Abstract*—Autonomous Underwater Vehicles (AUVs) are a useful tool for science and industry. They significantly reduce the risk to humans in operations in hazardous and high cost situations. The use of multiple AUVs can enhance the operational capabilities by introducing specialisation of AUV capabilities and parallelising task execution. The coordination of the multi-AUV team requires communication among its members. Underwater communications are low bandwidth, high latency and error prone. This paper studies different task allocation strategies for an underwater archaeological inspection scenario under communication constraints. Three different distributed methods are implemented and compared in simulation. The first is a greedy allocation method used as a baseline for comparison. The second is a k-Means based formulation aiming to balance the load among the robots. The third is the linear programming formulation of the multiple travelling salesmen problem. Results are analysed in the scope of mission completion time and the distance travelled by the robots. Results indicate that the k-Means method performs better when communication error rates are lower, while the mTSP method performs better when communication error rates are higher.**

I. Introduction

Scientific advances in the field of robotics have allowed the use of robotic teams to solve complex real-life problems. The use of autonomous robots facilitates the execution of tasks that were previously dangerous, expensive or time consuming when performed by humans. It has also enabled tasks that cannot be performed by humans. One environment that combines all the previous characteristics is the underwater environment, as humans are not naturally fit for that environment.

In particular, this paper focuses in the field of underwater archaeology. As described in [1], an archaeological exploration mission requires the successful cooperation of a heterogeneous fleet of autonomous underwater vehicles (AUVs). Specifically, the fleet is composed by two vehicle types. The first type is a fast vehicle capable of mapping an area and detecting potential archaeological artefacts called the search AUV (SAUV). The second type is a hover capable, but potentially slower vehicle, which is able to inspect and classify targets of interest. This vehicle is called the inspection AUV (IAUV). In the underwater archaeology scenario, the vehicles should map a given area and then search and inspect objects of archaeological interest. The coordination of the vehicles is enabled by using acoustic communications which are known to be lossy, low bandwidth and high latency [2], [3], [4].

This paper studies efficient ways to complete the aforementioned mission. Efficiency is studied from two different perspectives, namely, time and energy efficiency. Time efficiency studies the total time required to perform a mission. It is important as it allows more missions to be executed in a single day. The energy efficiency is achieved by trying to minimise the distance the vehicles have to travel in order to explore and inspect archaeological artefacts. Energy efficiency enables the execution of longer missions. To achieve the required efficiency two different methods are implemented and compared in a simulated environment. The first method is a k-Means based method described in [5]. The second method utilises the linear programming formulation of the multiple travelling salesmen problem (mTSP). Both methods are compared against a greedy method. Regarding the distance efficiency, the k-Means method performs better on low communications error rates. In higher error rates the mTSP approach produces better results. The baseline greedy method shows a consistent performance close to the other two methods. From the scope of time efficiency, the k-Means consistently produces the best results maximising the robot utilisation. The mTSP method follows closely, while the greedy method requires much more time for mission execution.

The rest of the paper is organised as follows. Section II describes the related work with the current research. In section III the methods implemented and tested are presented. Section IV analyses the simulation setup used to evaluate the different methods. In section V the results of the simulation evaluation are presented and discussed. Finally, in section VI the paper concludes, proposing possible future research directions.

II. Related work

The multi-robot task allocation (MRTA) problem has been actively researched in the past years. The first formal taxonomy is presented in [6]. In that study the problem is categorised using three different metrics. The first metric considers the robot type and whether it can complete multiple tasks in parallel or not. The second metric is about the task type. It refers to tasks that can be completed by one robot and tasks that require more. Finally, in the third metric tasks are classified based on their allocation type. It can be either instantaneous assignment, where future allocations are not considered, or time extended assignment, where future tasks have to be considered in a schedule.

Building on [6], the work presented in [7] further refines the MRTA problem by presenting a two level taxonomy. The first level is one dimensional and categorises the task allocation based on the interdependence of the agent-task utilities. In this way the task allocation problem is categorised based on how coupled it is, and thus how difficult is to solve. For this first level four different categories are defined, namely, tasks with no dependencies, with in-schedule dependencies, with cross-schedule dependencies and with complex dependencies. The second level is optional and it uses the taxonomy proposed in [6] to provide more information regarding the problem.

In [8] MRTA is considered as a sub-problem of the general distributed intelligence problem. In the same work a categorisation based on the interactions among the team members is attempted.

Several examples of applications of multi-robot systems can be found in the literature. In [9] multi-robot exploration and mapping is performed. The system presented there is a centralised system. Exploration is performed by frontier search and the maximisation of a utility function. In [10] the extension of the aforementioned paper is presented. In that case the information gain of the robot traversing a path is used to calculate the utility for the allocation. Behaviour based methods have also been used in solving the MRTA problem. For example in [11] an architecture that matches limited capability robots to tasks based on their fitness to perform each task. In [5] a balanced MRTA (BMRTA) is presented. This approach is using the k-Means machine learning algorithm to cluster tasks and then allocates them to robots. The work presented in the current paper builds upon [12] where a centralised experimental evaluation of the k-Means allocation was presented.

III. Methods

As the robotic team is composed of several heterogeneous members, an efficient method to exchange information is required. This is achieved by using a distributed world model architecture which is described in [13]. This architecture allows efficient communication by transmitting data based on the information needs of each peer. It also provides mechanisms that make it robust to communication errors.

For the efficient cooperation, two task assignment strategies are examined in this paper. The first task assignment strategy is based on the linear programming formulation of the multiple travelling salesmen problem (mTSP). The solution of the mTSP provides optimised solutions to the problem where many robots have to visit multiple locations. The second is based on the k-means clustering algorithm. This approach is used as a faster approximation to the mTSP. Both strategies are compared against a greedy strategy.

To create a more realistic setting where the methods are evaluated, it was decided that the targets would be incrementally discovered as the mission progresses. At a user defined interval a new target is discovered and transmitted to the inspection vehicles. This is different from what is found in the classic mTSP literature, where all the targets to be visited are known a priori. To overcome that, a replanning is triggered whenever new target information is available.

The task allocation strategies are implemented in a distributed manner. Each robot will calculate a task allocation solution based on its local knowledge about the team status and execute the part that is allocated to it. The fact that the communications are erroneous can lead to inconsistencies of the knowledge between the agents. This can lead to different allocations and thus different performance.

A. *Greedy task allocation*

In the greedy task allocation strategy the targets are assigned taking into consideration only the distance of that target from each candidate AUV. Whenever a new target is discovered the distance from each robot is calculated. Then the target is assigned to the robot currently closest to the target.

B. *k-Means task allocation*

The k-means based task allocation strategy tries to take advantage of the spatial proximity of the targets to be inspected. The k-means algorithm, as presented in [14], provides a method of clustering a set of observations based on a distance metric. Given a set of n observations $X \in \mathbb{R}^d$, the algorithm tries to partition these observations into k sets with $k \leq n$. It is achieved by choosing k mid-points M that minimize the potential function,

$$\phi = \sum_{x \in X} \min_{m \in M} \|x - m\|^2$$

The algorithm is simple and can be seen in the following listing:

1) Randomly choose k centres $M = (m_1, ..., m_k)$.
2) For each $i \in (1, ..., k)$, calculate the points of X that are closer to m_i than m_j for all $i \neq j$, and assign them to cluster M_i.
3) For each $i \in (1, ..., k)$, set m_i as the center of mass of points in M_i.
4) Repeat steps 2 and 3 until M not changes.

In the case of task allocation studied in this paper k is set to be equal to the number of IAUVs, thus creating one cluster per IAUV. Whenever a new target is detected the clustering algorithm clusters all the unclassified targets. Each cluster is then assigned to the IAUV that is closest to the mid-point of the cluster. Finally each IAUV solves a simple and fast TSP instance to obtain the optimal way to visit the targets within a cluster. In this paper the k-means++ algorithm is used, as presented in [15]. An example of the k-means algorithm can be seen in figure 1.

C. *mTSP task allocation*

The multiple travelling salesmen problem, described in [16], is an extension to the classic travelling salesman problem, where the aim is to find the shortest route to traverse a set of waypoints. In the mTSP there are many salesmen, and thus a set of routes that minimise the total cost of travelling are generated. The aim of the mTSP is to minimise the objective function:

$$\sum_{(i,j) \in A} c_{ij} x_{ij}$$

The sum describes the total cost of traversing all the waypoints, where c_{ij} is the cost of travelling from waypoint i to waypoint j and x_{ij} is a binary variable denoting the existence of a path between the two waypoints in a proposed solution. A is a set of arcs that connect the waypoints.

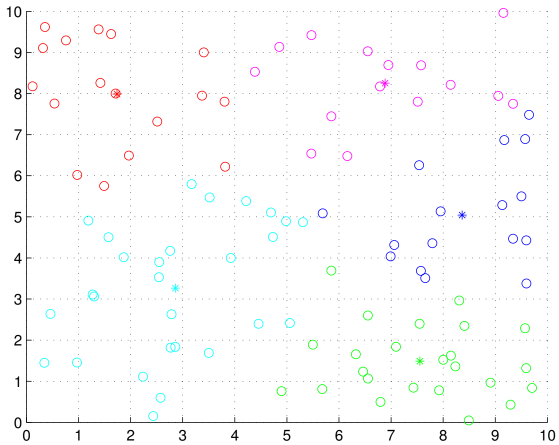

Fig. 1. Clustering of 100 2-d observations in five clusters using the k-means algorithm. The stars represent the k centres and the data points are coloured the same as their respective centre of mass.

The rest of the linear programming formulation is the following:

$$
\begin{aligned}
s.t. \sum_{j=2}^{n} x_{1j} &= m, && && (1)\\
\sum_{j=2}^{n} x_{j1} &= m, && && (2)\\
\sum_{i=1}^{n} x_{ij} &= 1, && j = 2, ..., n && (3)\\
\sum_{j=1}^{n} x_{ij} &= 1, && i = 2, ..., n && (4)\\
u_i + (L-2)x_{1i} - x_{i1} &\leq L-1, && i = 2, ..., n && (5)\\
u_i + x_{1i} + (2+K)x_{i1} &\geq 2, && i = 2, ..., n && (6)\\
x_{1i} + x_{i1} &\leq 1, && i = 2, ..., n && (7)\\
u_i - u_j + Lx_{ij} + (L-2)x_{ji} &\leq L-1, && 2 \leq i \neq j \leq n && (8)\\
x_{ij} &\in 0, 1, \forall (i, j) \in A. &&&&
\end{aligned}
$$

Constraints 1 and 2 ensure that the number of routes created are equal to the number of salesmen. Constraints 3 and 4 allow each target to be visited by only one salesman. Constraints 5 and 6 are used to enforce the minimum and the maximum number of targets that each salesman can visit. The minimum and maximum number of targets is user tunable and is given in L and K. Constraint 7 prevents the salesmen to travel to only one target. Finally, constraint 8 is a subtour elimination constraint preventing the creation of disconnected subtours. A solution of a simple mTSP problem can be seen in figure 2. The linear programming problem is solved by the GUROBI software [17].

Whenever new target information is received an allocation procedure over the remaining unclassified targets is triggered. The mTSP requires the robots to start from the origin. While the mission is performed the robots will be in different

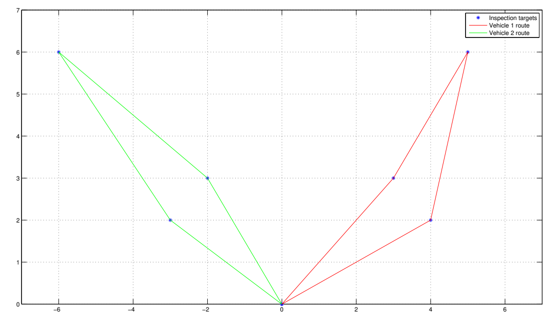

Fig. 2. mTSP solution of six cities and two salesmen. The two salesmen must start and return to the same place. Each city is visited by only one salesman.

positions, but for allocation purposes it is assumed that they are at the origin. The mTSP produces a number of paths that equals the number of robots. To assign a path to a robot the midpoint of the targets belonging to each path is calculated. Then each robot is assigned its closest midpoint. The targets are then ordered using a standard TSP algorithm based on the robot's current position. In cases where the targets to be assigned are less than the robots a greedy allocation is performed.

IV. Experimental setup

Experiments were conducted using the communications and navigation simulators developed by the Ocean Systems Laboratory. The communications simulator is an application level simulator that allows user-defined error, bandwidth and latency level in communications. The navigation simulator is a dynamics simulator using the hydrodynamic model of Nessie VII AUV [18].

The robots had to perform a simulated archaeological inspection mission. In that type of mission SAUVs map a predefined area of interest and search for potential archaeological artefacts. Any discovered artefacts are transmitted to the IAUVs, which in turn perform a reacquisition and a classification. In the current paper the robotic team was composed by an SAUV, that was detecting targets, and two IAUVs, which performed the inspection and classification.

During the mission execution, the phases of detection, inspection and classification were simulated. The SAUV was detecting targets that were randomly generated in an area of 100 by 100 meters around it. To ensure the correct evaluation of the allocation methods ten files with randomly generated targets were used. Each method is evaluated using the same target detection pattern. This will make sure that no method is favoured due to a specific target generation pattern. The target detection period was one minute. There were 10 different set of targets aiming to provide better insight to the allocation schemes and to remove any advantage due to the targets positions. To simulate the inspection and classification phases for the IAUVs an inactivity period of 1.5 minute was imposed whenever they reached a target. An instance of execution can be seen in figure 3.

Regarding the communications simulation, different packet error rates were used. Namely, 0%, 20%, 40% and 60% of packet error rate was selected. The packet size and flight time,

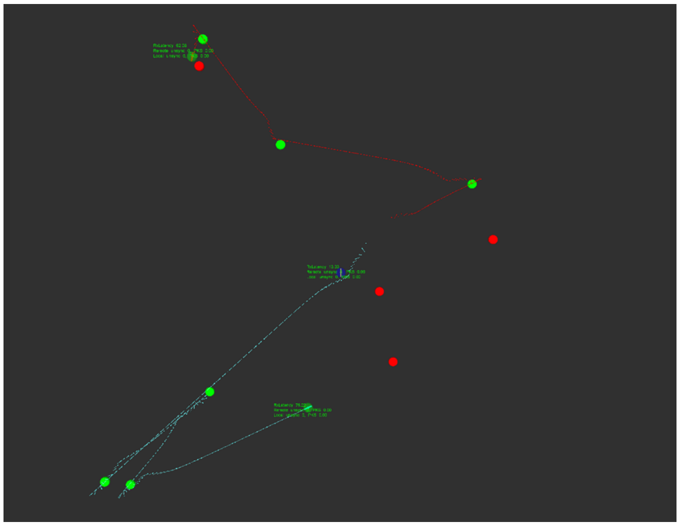

Fig. 3. Mission execution instance. The transparent spheres represent the vehicles. The green ones are the IAUVs and the blue the SAUV. The solid spheres are inspection targets. The red are unclassified and the green are classified. The red and cyan arrows represent the paths followed by the inspection vehicles.

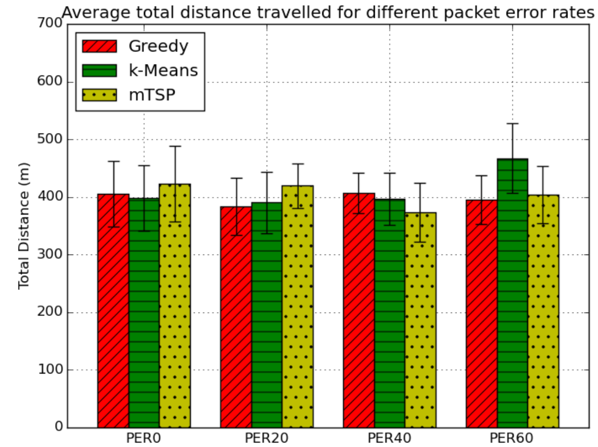

Fig. 4. Average total distance travelled for different values of packet error rate. Error bars represent one standard deviation. The greedy method performs consistently good. The k-Means method is better in lower packet error rates, while the mTSP performs in higher packet error rates.

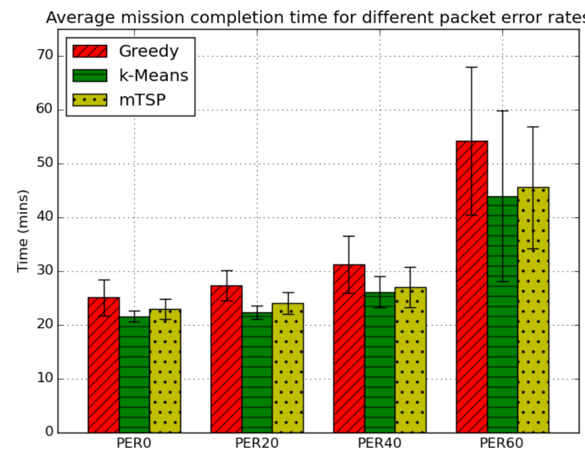

Fig. 5. Average total mission time for different values of packet error rate. Error bars represent one standard deviation. The k-Means method is constantly giving the best results achieving higher robot utilisation. The mTSP method is very close to that. The greedy method underperforms as it allows robots to remain idle.

thus the channel bandwidth, were selected to be 512 bytes and 2 seconds respectively. Finally, the time slot period, used to check for packet collisions, was set to be 10 seconds.

V. Results

In this section the simulation results are presented. The results are analysed under the scope of distance and time efficiency of each method. Distance was chosen as a metric of energy efficiency, as the less distance the robots have to travel, the more energy they have to spend on other tasks. Time efficiency is important regarding the mission execution, as users of robotic systems expect to have results in a reasonable amount of time. Finally the effect of the erroneous communications on each task allocation method is studied, as in real world conditions communications is not perfect and it is important that the methods are evaluated in such a scenario.

In figure 4, the results of the distance efficiency of each method are presented. For lower packet error rates the greedy and the k-Means based methods perform equally well, while the mTSP one performs on average slightly worse. On the other hand, when the error rates grow larger the mTSP method performs better, while the k-Means method underperforms. It is noteworthy that the greedy method gives good results distance-wise.

Figure 5 presents the results of the time efficiency of each method. The k-Means method constantly performs better than the other methods. This shows that it utilises the robots better. On the other hand, the greedy method is the one that performs the worst. The mTSP method performs closely to the k-Means method.

The utilisation of the robots, as indicated by the time it takes to complete a mission, explains the differences in the distance efficiency metric. The greedy method is assigning the targets to the closest robot and then the targets are visited in an optimised way provided by the internal TSP solver. In this way, the number of the targets assigned to each robot can be unbalanced and one robot may remain idle. Another issue that the greedy algorithm was facing emerged from the inconsistencies of the world model of each robot caused by latent and erroneous communications. There were cases where the robots had old information regarding the position of each other and wrongly assigned the target. As the greedy scheme is not allowing for a reallocation this caused problems with the mission execution specially in high packet error rates. The solution to that was to set a timer of the expected end of the mission with the current information. If this timer expired and the mission was not finished then a total reallocation was taking place. The k-Means and mTSP have an increased utilisation of the robotic team and achieve much better mission execution times.

VI. Conclusion

In this work the problem of multi-robot task allocation is studied under high latency and unreliable communications. Two task allocation methods are implemented and experimentally tested. Comparison between the methods can show the benefits of each and the effects of the communications limitations to task performance. The methods were tested in a simulated underwater archaeology mission scenario where a heterogeneous fleet of AUVs located, inspected and classified potential archaeological artefacts. Coordination was achieved using a simulated communications channel that allowed a user defined latency and error rate. Results indicate that the k-Means method is more suitable for lower error rates. It is the most time efficient and the robots have to travel less or equal to the other methods. In higher error rates the mTSP method is preferable as it requires almost the same time as the k-Means but produces better results distance-wise. The greedy method underutilises the team with robots remaining idle. This allows it to produce optimised results regarding the distance the robots have to travel but with a cost in mission execution time.

An extension to this work would be to use a multi-depot multiple travelling salesmen formulation. This would capture the dynamics of the mission better as it considers the current position of the vehicles and could produce more optimised results. Another interesting aspect would be the application of methods used in heterogeneous vehicle routing [19]. This would allow a heterogeneous team of robots to perform various tasks in an optimised way, as it would capture the different capabilities of each robot in the cost function.

Acknowledgment

The research leading to these results has received funding by the European Union Seventh Framework Programme FP7 ARROWS - under grant agreement No. 308724.

The authors would like to thank all the members of the Ocean Systems Laboratory at Heriot-Watt University.

References

[1] ARROWS. Online - http://www.arrowsproject.eu/, 2012. ARchaeological RObot systems for the Worlds Seas.

[2] M. Roth, D. Vail, and M. Veloso, "A real-time world model for multi-robot teams with high-latency communication," in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 3, Oct 2003.

[3] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke, "Data collection, storage, and retrieval with an underwater sensor network," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, SenSys '05, (New York, NY, USA), ACM, 2005.

[4] Z. Saigol, G. Frost, N. Tsiogkas, F. Maurelli, D. Lane, A. Bourque, and B. Nguyen, "Facilitating cooperative AUV missions: experimental results with an acoustic knowledge-sharing framework," in *Proceedings of IEEE-MTS Oceans'13, San Diego, USA*, 2013.

[5] M. Elango, S. Nachiappan, and M. K. Tiwari, "Balancing task allocation in multi-robot systems using K-means clustering and auction based mechanisms," *Expert Systems With App.s*, vol. 38, no. 6, 2011.

[6] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *IJRR*, vol. 23, no. 9, 2004.

[7] G. A. Korsah, A. Stentz, and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *IJRR*, vol. 32, no. 12, 2013.

[8] L. E. Parker, "Distributed intelligence: Overview of the field and its application in multi-robot systems," *Journal of Physical Agents*, vol. 2, no. 1, 2008.

[9] W. Burgard, M. Moors, and C. Stachniss, "Coordinated multi-robot exploration," *IEEE Transactions on Robotics*, vol. 21, no. 3, 2005.

[10] J. Butzke and M. Likhachev, "Planning for multi-robot exploration with multiple objective utility functions," in *IROS*, 2011.

[11] F. Tang and L. E. Parker, "Asymtre: Automated synthesis of multi-robot task solutions through software reconfiguration," in *ICRA*, 2005.

[12] N. Tsiogkas, G. Papadimitriou, Z. Saigol, and D. Lane, "Efficient multi-AUV cooperation using semantic knowledge representation for underwater archaeology missions," in *Oceans '14 St. John's*, 2014.

[13] F. Maurelli, Z. Saigol, J. Cartwright, D. Lane, A. Bourque, and B. Nguyen, "TDMA-based exchange policies for multi-robot communication of world information," in *IFAC MCMC*, 2012.

[14] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *The 5th Berkeley symposium on mathematical statistics and probability*, vol. 1, California, USA, 1967.

[15] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *18th ACM-SIAM symposium on Discretealgorithms*, 2007.

[16] I. Kara and T. Bektas, "Integer linear programming formulations of multiple salesman problems and its variations," *European Journal of Operational Research*, vol. 174, no. 3, 2006.

[17] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2014.

[18] N. Valeyrie, F. Maurelli, P Patron, J. Cartwright, B. Davis, Y. Petillot, "Nessie V Turbo: a new hover and power slide capable torpedo shaped AUV for survey, inspection and intervention," in *AUVSI North America 2010 Conference*, 2010.

[19] M. Gendreau, G. Laporte, C. Musaraganyi, and É. D. Taillard, "A tabu search heuristic for the heterogeneous fleet vehicle routing problem," *Computers & Operations Research*, vol. 26, no. 12, pp. 1153–1173, 1999.