Robust Heterogeneous Multi-Robot Routing for Low-Intelligence Agents

Nick Sullivan[1*], **Steven Grainger**[1], **Ben Cazzolato**[1]
[1]The University of Adelaide, Australia
*nicholas.sullivan@adelaide.edu.au

Abstract

A solution to a case of heterogeneous multi-robot routing is presented using *opportunistic collaboration*, which draws on the concept that robots are held responsible for completing tasks no other robots can. These robots choose to complete tasks they are not responsible for (collaborate) when low-cost opportunities arise (opportunistically). The algorithm is low-cost in both implementation and processing, it is able to operate with unreliable or failed communications, and can be utilised dynamically with simple optimisation techniques. The algorithm is designed for use in agents with very limited processing capabilities. Its performance benchmarked against a sequential single-item auction, an efficient heuristic for agents of higher intelligence. We also compare performance to other greedy solutions, illustrating that our algorithm can be used to reduce task completion times.

1 Introduction

Multi-robot systems (MRSs) have the potential to outperform single robot systems at completing tasks. This is due to their spatial distribution, robustness through redundancy, improved versatility and scalability [Khamis et al., 2015]. Many multi-robot systems use homogenous robots for ease of design. However, heterogeneous robots can increase functional capabilities and improve performance. In addition, robots that are initially homogeneous may become heterogeneous through sensor failure or equipping tools. This paper deals with multi-robot systems using heterogeneous robots that occupy specialised roles within the team.

A fundamental challenge for multi-robot systems is that of *routing*. Given a set of tasks with locations and a set of robots, it is desired that the robots complete the tasks according to the system objectives. These objectives may relate to completion in the shortest time, the shortest distance travelled, the lowest energy usage, the highest likelihood of success, or many other options. In heterogeneous multi-robot systems, this problem is further complicated in that not all robots are capable of completing all tasks. In fact, it is common for certain robots to be specifically designed to accomplish certain tasks. Some robots fly, some drive, and others have expensive sensors, actuators, or processors. There is no robot that performs best in every scenario. Heterogeneous capabilities may also occur due to ownership of contested resources. For example, a robot with a drill must either perform the drilling or relinquish the drill.

We present a solution using the concept of opportunistic collaboration; a term used in work practices, where employees or workplace divisions encounter opportunities to help one another. Research in workplace management has shown how workplace collaboration enhances innovation and performance [Capdevila, 2015], and has explored techniques to promote collaboration [James et al., 2016]. If the collaboration is one-sided (e.g. Party A requires knowledge that Party B already has, so Party B helps Party A) parties must strike a balance between completing their own work and helping others. Clearly, optimal behaviour lies somewhere in between being purely selfish (refusal to help others) and being purely selfless (delaying their own work).

In the multi-robot system investigated here, there are tasks that each robot is *responsible* for completing. Few other robots can complete these tasks. There are also tasks that are *common* and can be completed by the majority of robots. Robots aim to complete the jobs they are responsible for, but in doing so may encounter opportunities to complete common tasks at lower than usual cost. The robots will complete these common tasks provided they are within a *selfless* threshold, which dictates how far out of its way a robot is willing to go. By adjusting this threshold, it is possible to complete all tasks efficiently. This paper will illustrate that this methodology is simple to implement, extremely quick to execute, can outperform other algorithms of similar complexity, and can operate in environments with no available communication during operation. We will also show that this method can adapt to task requirements that change over time through the use of simple optimisation functions, exemplified in this paper by the use of stochastic gradient descent [Saad, 1998], which converges towards inputs producing minima/maxima, albeit local. While it guarantees convergence for convex functions, it can also be used for many non-convex functions [Bianchi and Jakubowicz, 2013].

In this paper, we explore the use of opportunistic collaboration for 2D non-holonomic land vehicles. We assume that tasks are simple, in that they can be completed by a single robot. The tasks are given 2D locations that a robot must move to in order to be completed.

Section 1 introduces the paper, Section 2 reviews heterogeneous routing algorithms. Section 3 introduces the algorithms executed by the mobile robots, acting as agents. Section 4 describes the optimisation algorithm for handling dynamic scenarios. Section 5 describes the simulation platform, as implemented in MATLAB. Section 6 discusses the results obtained, and Section 7 concludes the paper.

2 Related Works

Multi-robot routing is a fundamental problem in multi-robot systems and is prevalent in robotics, computer science and mathematics. It has been modelled in a number of different ways, the most common being a multiple Travelling Salesman Problem (mTSP), and a Multi-Depot Vehicle Routing Problem (MDVRP). This is a non-deterministic polynomial-time (NP) hard problem [Montoya-Torres et al., 2015], and can require a large amount of time to solve optimally. These solutions can be segmented into two categories: centralised, and decentralised.

Centralised solutions deal with minimising or maximising the objective functions of a system, and are often capable of finding optimal solutions in worst-case exponential time. Examples of centralised solutions include branch and bound techniques, genetic algorithms and simulated annealing [Khamis et al., 2015]. Research in this area aims to improve performance of these algorithms [Kartal et al., 2016], or to apply them to realistic scenarios, such as a genetic algorithm to optimise time and fuel consumption for robots inspecting an industrial plant [Jose and Pratihar, 2016], or how best to send data mules to collect information in wireless sensor networks [Lukic et al., 2015]. Typically, these solutions require a large amount of communication in order to operate. They often have a single point of failure and do not scale well [Khamis et al., 2015].

Decentralised solutions are more common for multi-robot systems. A highly efficient decentralised solution uses auction-based task allocation [Khamis et al., 2015]. Tasks are locally sold or traded to nearby robots, and the more capable robots are able to bid higher than the less capable ones. Recent work in this area includes altering the price of tasks in order to distribute tasks amongst available robots evenly [Liu et al., 2014], using path-finding costs to ensure tasks can be completed quickly [Öztürk and Kuzucuoğlu, 2015], and improving solutions over time, which was experimentally validated using the Robot Operating System (ROS) [Koubâa et al., 2016]. Some algorithms include additional constraints such as time windows for task completion [Nunes and Gini, 2015]. Applications include using heterogeneous robots in assisted living environments, where some tasks require immediate prioritisation [Das et al., 2015]. These solutions can be complex to implement and require reliable communications but are far more robust than their centralised counterparts.

Solutions for low-intelligence agents have been explored for some systems. A dynamic task allocation for multi-robot foraging has been designed with robots that do not communicate and have limited planning [Lerman et al., 2006]. There has also been a characterisation of how computation power affects execution without global coordination [DEmidio et al., 2016].

Recently, heterogeneous robots have also been considered, producing the problem known as the Heterogeneous Multiple Travelling Salesman Problem (H-mTSP) and the Multi-Depot Heterogeneous Vehicle Routing Problem (MDHVRP). In some cases, the heterogeneous robots work in conjunction to access otherwise inaccessible areas, such as the firefighting robots and debris clearing robots to reach as many fires as possible [Jones et al., 2011]. A few use drones located on trucks to leverage the long-distance capabilities of trucks with the high-reaching capabilities of drones [Klaučo et al., 2016; Mathew et al., 2015]. Others look at foraging, where some robots gather information to find resources and others perform the foraging to collect those resources [Liemhetcharat et al., 2015].

There has also been work where heterogeneous robots do not intrinsically require one another to operate but still share the same environment. One solution is to segment the environment to reduce redundancy [Pereira et al., 2015], but is only applicable to situations where different tasks appear in different segments. Other solutions deal with task allocation where certain tasks could only be completed by certain robots, while other tasks can be completed by any robot [Sundar and Rathinam, 2015]. An application for this includes dial-a-ride services, where some passengers require specialised cars [Braekers et al., 2014]. An extension of this is where tasks can be completed by a subset of robots, and these robots can achieve more than one task [Chopra and

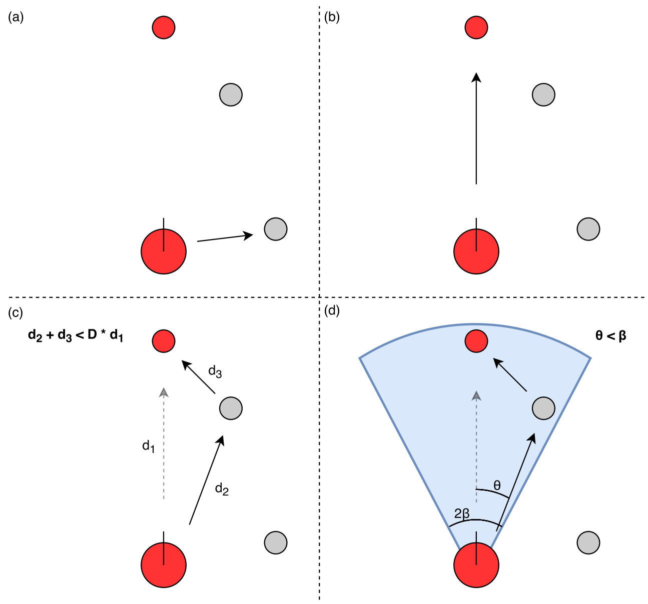

Figure 1: Examples of the compared agent algorithms. A robot (large disk) must complete common tasks (small disk, grey) as well as private tasks (small disk, red). (a) The Selfless algorithm goes to the closest task. (b) The Selfish algorithm goes to closest private task. (c) The Distance Limiter aims for the closest private task, but may detour if the extra distance is small enough. (d) The Bearing Limiter aims for the closest private task, but may detour if the bearing change is small enough.

Egerstedt, 2014]. This property is known as *degeneracy* and has been shown to improve the robustness and evolvability of a system [Whitacre and Bender, 2010]. Other papers have explored the idea of degeneracy without explicitly mentioning it, such as a paper that uses heterogeneous robots to search and identify humans [Nagy and Anderson, 2016]. A concept that builds on this is that of *preparedness*, where robots that have rare skills are given fewer tasks in order to ensure they are available for future tasks [Kim et al., 2012]. This has been implemented using an economy-based Gini-coefficient strategy [Wu et al., 2017]. This has also been explored in warehouse distribution, where all robots are able to distribute objects but have different levels of suitability for different objects [Luo et al., 2015].

Current solutions for heterogeneous task allocation are designed for intelligent agents with communication availability, and are unable to be applied to low-intelligence agents that may not have communication. This paper introduces the opportunistic collaboration algorithm as a means to fill this gap.

3 Agent Algorithms

This paper aims to use low-intelligence agents that are simple to implement and able to decide quickly which task to complete next. This leaves processing available for completing the task at hand. We use communicative and non-communicative model-based agents. Model-based agents are able to store a representation of their environment but do not perform long-term plans in order to achieve their goal. In this paper, the communicative model-based agents will perform task auctioning when more than one robot is interested in a particular task. This allows robots to purchase common tasks so that robots do not attempt to complete the same task. The purchasing process is done through auction. When a robot wishes to complete a task, it prompts nearby robots to see if they are also interested in that task. Each interested robot bids an amount based on ease of completion. The bids value is the distance to the task, i.e, a robot will always give away a task to a robot that is closer than they are, provided they want to buy it. In short, a single round auction is performed whenever a robot wishes to complete a task, and other robots only bid if they also wish to complete the task. No future planning occurs, so this process is very fast.

If communication is not available, the robot will assume no other robot wishes to claim that task. This means it is possible for multiple agents to attempt to complete the same task. In this case, the first agent to get to the location will complete the task. When the task is completed, all other agents will observe that their task is completed, and move on. This guarantees that a solution will be found for both the communicative and non-communicative case, robots will move directly to a task until that task becomes completed.

The following robot algorithms will be compared:

- Selfless
  - Goes towards the closest task it can complete
- Selfish
  - Goes towards the closest private task it can complete
  - If there are none, it will go towards the closest common task
- Distance Limiter
  - Selects the closest private task it can complete
  - If there is a common task that is closer and the detour multiplies the distance by less than D, goes to that task instead
- Bearing Limiter
  - Selects the closest private task it can complete

– If there is a common task that is closer and changes the robots bearing by less than $\beta°$, goes to that task instead

A graphical example of these algorithms are presented in Figure 1. These algorithms are *greedy* in that they only consider the immediate state when performing task allocation. Greedy solutions are often sub-optimal, but are much faster to solve than optimal solutions [Neapolitan and Naimipour, 2010]. As mentioned in Section 1, optimal solutions for this problem can take an enormous amount of processing time. Optimal solutions also need information about all robot and task locations, whereas greedy solutions only need information about nearby tasks. This allows them to operate in environments with limited communication and limited knowledge. Greedy solutions can therefore handle dynamic changes quickly.

The D and β variables specified for the opportunistic collaboration algorithms are selected at the beginning of each run. D and β both have minimum and maximum values which result in purely selfish or selfless behaviour. When $D = 1$, or $\beta = 0°$, robots will not go out of their way to complete a common task. When $D = 3$, or $\beta = 180°$, robots will complete the closest task they can regardless of ownership. These values are standardised using a selfless ratio w that ranges from 0 (selfish) to 1 (selfless),

$$D = 2w + 1 \tag{1}$$
$$\beta = 180° w. \tag{2}$$

4 Learning Algorithms

The robots are provided with task details at the beginning of each run. When using an algorithm involving opportunistic collaboration, the selfless ratio for the robots is also provided at the beginning of each run. The selfless ratio is used to decide if a common task is 'on the way', as shown in Section 3. The ratio is optimised using stochastic gradient descent for a single variable. Gradient descent has guaranteed convergence for convex data and semi-guaranteed for quasi-convex data [Saad, 1998], where there is a single local minima or maxima. Each run uses randomly generated task locations, therefore the time of each run is stochastic. It has been found previously that stochastic gradient descent will not converge to a point, rather it will move around within bounds that represent the stochastic uncertainty [Hardy et al., 2016].

A short description of the stochastic gradient descent algorithm is presented here. We wish to minimise the time it takes to complete all tasks, represented by function Q. The input to the function is the selfless ratio, w. The task locations are randomly generated each attempt, meaning that we wish to minimise the following

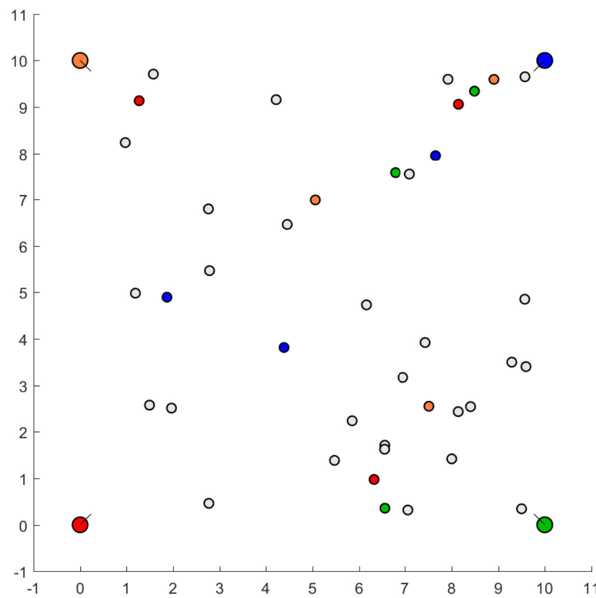

Figure 2: The simulation in MATLAB. The robots (large disks) complete tasks (small disks) by moving to the task location. Grey tasks can be completed by any robot, while other tasks can only be completed by the robot with matching colour.

function over n attempts,

$$Q(w) = \frac{1}{n}\sum_{i=0}^{n} Q_i(w). \tag{3}$$

The learning procedure is as follows. The input value w is updated by adding the step size η (also known as the learning rate) in the direction that decreases function output.

$$w := w - \eta \nabla Q(w) \tag{4}$$

Taking batches of n function outputs before updating the input value can improve convergence, as averaging multiple outputs will increase confidence at that point, represented as

$$w := w - \frac{1}{n}\sum_{i=0}^{n} \eta \nabla Q_i(w). \tag{5}$$

The value n can be adjusted, high values will make convergence smoother but it will take longer to converge.

5 Testing Platform

The testing platform is a 2D simulation for non-holonomic ground robots, implemented in MATLAB. There are up to 40 markers, representing tasks, at random locations within the testing area. These tasks must be completed by all four robots in the system. Some of these tasks can be completed by any robot. These are referred to as *common* tasks. Other tasks can only be completed by one given robot, known as *private* tasks. The goal of the multi-robot system is to complete all the tasks in as short a time as possible. An example image of the simulation is given in Figure 2. It is assumed that all tasks can be completed in the same amount of time and that they are not inter-related.

At the beginning of each run, the robots are informed of all task locations. When all tasks have been completed, they return home to log details and recharge.

Three scenarios are tested. Scenario 1 involves 28 common tasks and 3 private tasks for each robot, producing 40 total tasks. The task locations are within a 10x10 metre area. This represents a situation where robots operate within the same environment. In Scenario 2 there are also 28 common tasks and 3 private tasks each. The task locations are within a 20x20 metre area, and the private tasks are located in the 10x10 corners. This represents a situation where robots operate near one another, but not strictly in the same environment. Scenario 3 involves 30 common tasks and 10 private tasks for one robot, producing 40 total tasks. The task locations are within a 10x10 metre area. This represents a situation where one robot has greater functionality than the others, but they all operate within the same environment.

Task locations were randomly generated each run according to the scenario rules. This produces some level of unpredictability for each run, so that dynamic task allocation is tested.

The robots are non-holonomic land robots in that they can only move forwards or backwards in the direction they are facing. They move at up to 0.5 metres per second, and can rotate on the spot at up to 1 radian per second.

We compare these solutions to those found from a sequential single-item auction [Lagoudakis et al., 2005]. One task is allocated per auction round, and robots create new bids based on the tasks that they have received. Robots initially have no tasks allocated to them. They then bid on each task. The overall best bid is the winner, and a robot is allocated the task. For each remaining unallocated task, the robot calculates the cost of adding the task to its tour, and bids accordingly. Another auction round is held and the process repeats until all tasks are allocated.

Sequential single-item auctions are expected to produce significantly better solutions over our introduced methods, but our aim is not to outperform them. Rather, our methods are designed for low-intelligent agents with unreliable connectivity, where sequential single-item auctions are unable to be used. We compare results to sequential single-item auctions as a benchmark to illustrate reasonable results if provided with full communication connectivity and path-planning.

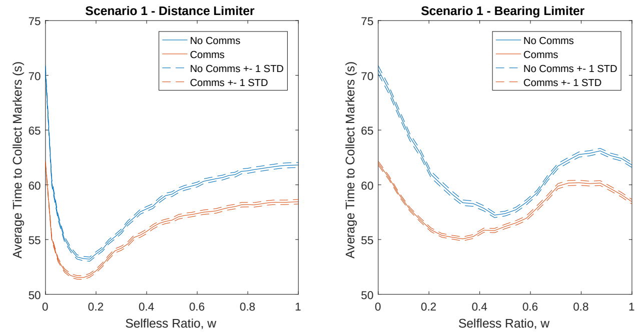

Figure 3: Time to complete all tasks for Scenario 1 as a function of selfless ratio. There are 4 robots, 3 private tasks each, and 28 common tasks within a 10x10 metre area. Selfless ratio was tested in 0.025 increments, and 1000 tests were performed for each. The use of distance limiter has an optimal selfless ratio of 0.175 resulting in a time of 53.2 seconds (no-comms) and a selfless ratio of 0.15 resulting in a time of 51.5 seconds (comms). The yaw limiter has an optimal selfless ratio of 0.45 resulting in a time of 57.2 seconds (no-comms) and a selfless ratio of 0.325 resulting in a time of 55.1 seconds.

6 Results

6.1 Scenario 1

Scenario 1 involved 4 robots, 28 common tasks, 3 private tasks per robot. The tasks are randomly placed in a 10x10 metre area. Simulations are performed 1000 times for each selfless ratio. This is a situation where each robot is specialised to perform certain tasks, and there are also basic tasks that any of the robots can complete.

Figure 3 shows that purely selfless robots ($w = 1$) outperform the selfish robots ($w = 0$) in this scenario for both non-communicative and communicative cases. The opportunistic collaboration algorithms are capable of outperforming the standard selfish and selfless algorithms for a wide range of selfless ratios. The performance functions have local minima, which can be found using relatively simple convergence algorithms, as will be shown later in this section. The Distance Limiter outperforms the Bearing Limiter, and also has a single local minimum, making it a better candidate for optimisation. Some selfless ratios cause the Bearing Limiter to perform worse than the standard algorithms. This can be attributed to the fact that at high selfless ratios (> 0.7) the robots are willing to make large bearing changes ($> 125°$) to complete a common task, but will ignore close tasks that are behind them, acting as a blind spot. It can also be seen that the non-communicating

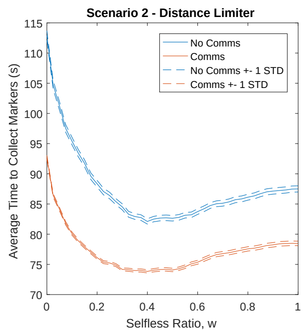

Figure 4: Time to complete all tasks for Scenario 2: 4 robots, 3 private tasks each, 28 common tasks within a 20x20 metre area. Selfless ratio was tested in 0.025 increments, and 1000 tests were performed for each. The use of distance limiter has an optimal selfless ratio of 0.400 resulting in a time of 82.1 seconds (no-comms) and a selfless ratio of 0.400 resulting in a time of 73.8 seconds (comms). The yaw limiter has an optimal selfless ratio of 0.800 resulting in a time of 85.3 seconds (no-comms) and a selfless ratio of 0.800 resulting in a time of 77.4 seconds.

opportunistic collaboration algorithms are able to outperform the communicating standard algorithms.

For this scenario, sequential single-item auctioning completed tasks in an average of 28.6 seconds, which is 56% faster than the best result.

6.2 Scenario 2

In this scenario, the tasks are spaced out into a 20x20 metre region, but all private tasks are within the 10x10 metre corners. There are 28 common tasks, and 3 private tasks for each robot. This acts as a scenario where the robots work in nearby environments rather than the same environment. The results are shown in Figure 4.

In this scenario, the results are similar to the previous scenario. Local minima exist and the Distance Limiter outperforms the Bearing Limiter. The optimal selfless ratios are higher (more selfless) than for Scenario 1, and the use of opportunistic collaboration has less of an impact. This makes sense, as the working environments are spaced apart. Each robot should complete more tasks in their area and collaboration is less applicable.

For this scenario, sequential single-item auctioning completed tasks in an average of 34.9 seconds, which is 47% faster than the best result.

6.3 Scenario 3

In Scenario 3 there were 4 robots, 30 common tasks, one robot has 10 private tasks. The tasks are randomly placed in a 10x10 metre area. This scenario represents a system where three robots can perform most tasks, but one robot has extra features and is able to perform the harder tasks. The results are seen in Figure 5. In this scenario, being selfish vastly outperforms being selfless. This is due to the specialised robot with 10 private tasks. If this robot selflessly completes common tasks then the common tasks are completed quickly, but the private tasks are delayed in their completion.

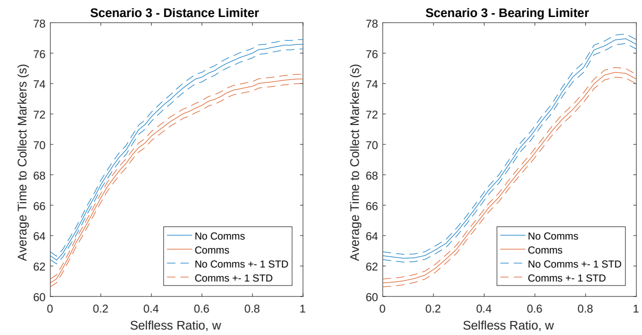

Figure 5: Time to complete all tasks for Scenario 3: 4 robots, one robot has 10 private tasks each, 30 common tasks within a 10x10 metre area. Selfless ratio was tested in 0.025 increments, and 1000 tests were performed for each. The use of distance limiter has an optimal selfless ratio of 0.025 resulting in a time of 62.4 seconds (no-comms) and a selfless ratio of 0.00 resulting in a time of 60.9 seconds (comms). The yaw limiter has an optimal selfless ratio of 0.075 resulting in a time of 62.5 seconds (no-comms) and a selfless ratio of 0.00 resulting in a time of 60.9 seconds.

It can be seen that the opportunistic collaboration algorithms support this scenario where collaboration is not beneficial. The collaborative aspect can easily be deactivated by using a selfless ratio of 0 or 1, making it no worse than the standard solutions.

For this scenario, sequential single-item auctioning completed tasks in an average of 35.2 seconds, which is 58% faster than the best result.

6.4 Dynamic Scenarios

The optimal selfless ratio can be found dynamically in changing scenarios using a simple feedback mechanism. The multi-robot system is treated as a black box, and the selfless ratio is altered depending on performance results. For the scenarios discussed earlier, the optimal ratio has local minima. Therefore, a simple optimisation process known as stochastic gradient descent is used to periodically converge to local minima that change over time. The technique is outlined in Section 4. Figure 6 shows the convergence for $n = 40$ (samples) and $\eta = 0.1$ (step size). These values can be adjusted to juggle between rate of convergence and the precision of convergence.

Figure 6 shows three graphs. In each graph, the task distributions change over time. The ideal selfless ratio

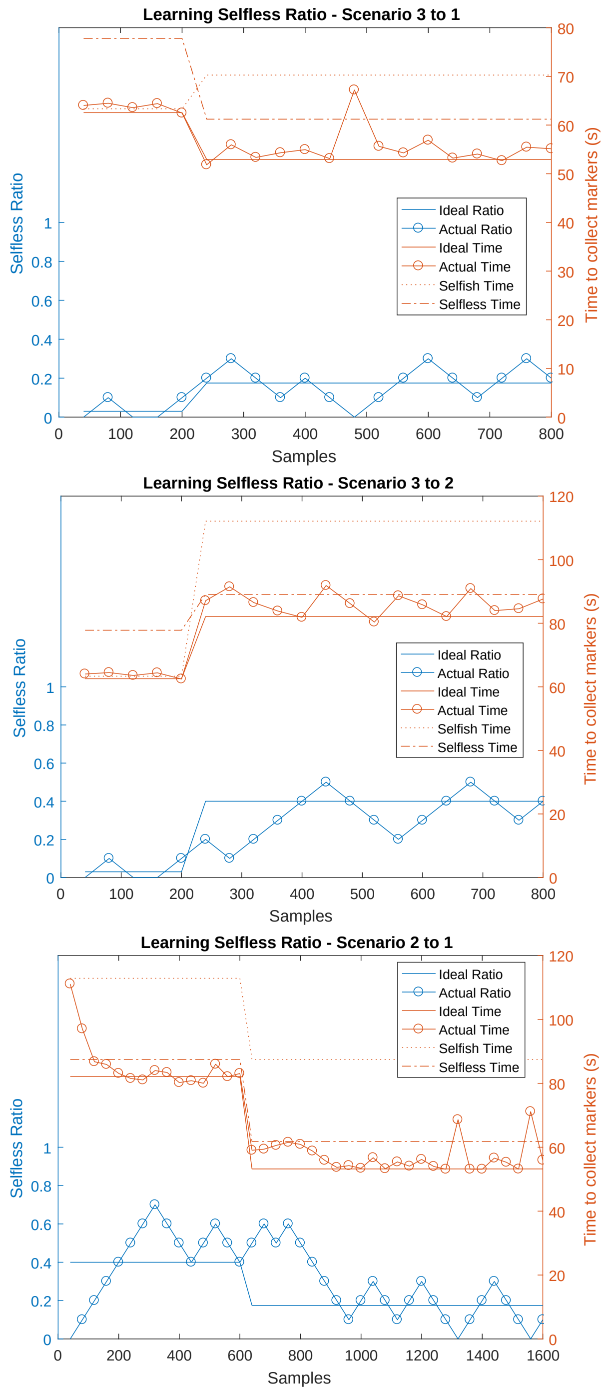

Figure 6: Learning the optimal selfless ratio using stochastic gradient descent. The scenario changes after a number of samples, causing a change in the optimal selfless ratio. The system is able to alter the ratio over time in order to reduce the completion time.

at each point in time is shown by a solid blue line. Tasks completed using the ideal ratio would take the ideal amount of time, as shown by a solid red line. The bounds of time completion (selfish and selfless) are shown by dashed red lines. The actual selfless ratios used are shown by a blue line with circles for each point, which results in the corresponding time shown by red line with circles for each point. We can see that a simple optimisation process is able to converge around the optimal selfless ratio. There is some jitter that occurs around the ideal ratios, this is because the learning system uses 40 samples, rather than the 1000 samples used previously. Using less samples causes higher variance of completion time, but significantly improves convergence time. Similarly, a higher learning rate will improve convergence time at a cost of convergence precision. The process shown here illustrates that even a simple optimisation process on a black box is sufficient for good results.

7 Conclusion

In this paper, we have presented a simple, robust solution to a heterogeneous multi-robot task allocation problem using opportunistic collaboration. The robots selfishly aim to achieve the tasks that they are responsible for and only perform common tasks when they can be completed at little cost. The acceptable cost is dictated by the selfless ratio, which can be altered when robots return to recharge. We also present a learning process to automatically optimise the selfless ratio over time.

Solving the multi robot routing problem optimally is NP-hard, taking a long time to solve for even a small number of tasks. In contrast, the opportunistic collaboration algorithm can be executed very quickly, leaving the robot processors free for performing their intended tasks. This makes it suitable for robots with little processing capability, such as those in swarms.

This algorithm is also suitable to improve the robustness of multi-robot systems operating in outdoor environments, as it can be executed with or without communication. In comparison, decentralised task allocation algorithms require reliable communications in order to perform, making them sensitive to environmental conditions.

Acknowledgements

The Commonwealth of Australia (represented by the Defence Science and Technology Group) supported this research through a Defence Science Partnerships agreement.

References

[Bianchi and Jakubowicz, 2013] Bianchi, P. and Jakubowicz, J. (2013). Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization. *IEEE Transactions on Automatic Control*, 58(2):391–405.

[Braekers et al., 2014] Braekers, K., Caris, A., and Janssens, G. K. (2014). Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. *Transportation Research Part B: Methodological*, 67:166–186.

[Capdevila, 2015] Capdevila, I. (2015). Co-working spaces and the localised dynamics of innovation in Barcelona. *International Journal of Innovation Management*, 19(03):1540004.

[Chopra and Egerstedt, 2014] Chopra, S. and Egerstedt, M. (2014). Heterogeneous multi-robot routing. In *American Control Conference (ACC), 2014*, pages 5390–5395. IEEE.

[Das et al., 2015] Das, G. P., McGinnity, T. M., Coleman, S. A., and Behera, L. (2015). A distributed task allocation algorithm for a multi-robot system in healthcare facilities. *Journal of Intelligent & Robotic Systems*, 80(1):33–58.

[DEmidio et al., 2016] DEmidio, M., Frigioni, D., and Navarra, A. (2016). Characterizing the computational power of anonymous mobile robots. In *Distributed Computing Systems (ICDCS), 2016 IEEE 36th International Conference on*, pages 293–302. IEEE.

[Hardy et al., 2016] Hardy, P., Cazzolato, B., Ding, B., and Prime, Z. (2016). A maximum capture width tracking controller for ocean wave energy converters in irregular waves. *Ocean Engineering*, 121:516–529.

[James et al., 2016] James, T. A., Page, J. S., and Sprague, J. (2016). Promoting interprofessional collaboration in oncology through a teamwork skills simulation programme. *Journal of Interprofessional Care*, 30(4):539–541.

[Jones et al., 2011] Jones, E. G., Dias, M. B., and Stentz, A. (2011). Time-extended multi-robot coordination for domains with intra-path constraints. *Autonomous Robots*, 30(1):41–56.

[Jose and Pratihar, 2016] Jose, K. and Pratihar, D. K. (2016). Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods. *Robotics and Autonomous Systems*, 80:34–42.

[Kartal et al., 2016] Kartal, B., Nunes, E., Godoy, J., and Gini, M. (2016). Monte Carlo tree search with branch and bound for multi-robot task allocation. In *The IJCAI-16 Workshop on Autonomous Mobile Service Robots*.

[Khamis et al., 2015] Khamis, A., Hussein, A., and Elmogy, A. (2015). *Multi-robot Task Allocation: A Review of the State-of-the-Art*, pages 31–51. Springer.

[Kim et al., 2012] Kim, M.-H., Kim, S.-P., and Lee, S. (2012). Social-welfare based task allocation for multirobot systems with resource constraints. *Computers & Industrial Engineering*, 63(4):994–1002.

[Klaučo et al., 2016] Klaučo, M., Blažek, S., and Kvasnica, M. (2016). An optimal path planning problem for heterogeneous multi-vehicle systems. *International Journal of Applied Mathematics and Computer Science*, 26(2):297–308.

[Koubâa et al., 2016] Koubâa, A., Cheikhrouhou, O., Bennaceur, H., Sriti, M.-F., Javed, Y., and Ammar, A. (2016). Move and improve: a market-based mechanism for the multiple depot multiple travelling salesmen problem. *Journal of Intelligent & Robotic Systems*, pages 1–24.

[Lagoudakis et al., 2005] Lagoudakis, M. G., Markakis, E., Kempe, D., Keskinocak, P., Kleywegt, A. J., Koenig, S., Tovey, C. A., Meyerson, A., and Jain, S. (2005). Auction-based multi-robot routing. In *Robotics: Science and Systems*, volume 5, pages 343–350. Rome, Italy.

[Lerman et al., 2006] Lerman, K., Jones, C., Galstyan, A., and Matari, M. J. (2006). Analysis of dynamic task allocation in multi-robot systems. *The International Journal of Robotics Research*, 25(3):225–241.

[Liemhetcharat et al., 2015] Liemhetcharat, S., Yan, R., and Tee, K. P. (2015). Continuous foraging and information gathering in a multi-agent team. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1325–1333. International Foundation for Autonomous Agents and Multiagent Systems.

[Liu et al., 2014] Liu, L., Shell, D. A., and Michael, N. (2014). From selfish auctioning to incentivized marketing. *Autonomous Robots*, 37(4):417–430.

[Lukic et al., 2015] Lukic, M., Barnawi, A., and Stojmenovic, I. (2015). Robot coordination for energy-balanced matching and sequence dispatch of robots to events. *IEEE Transactions on Computers*, 64(5):1416–1428.

[Luo et al., 2015] Luo, L., Chakraborty, N., and Sycara, K. (2015). Distributed algorithms for multirobot task assignment with task deadline constraints. *IEEE Transactions on Automation Science and Engineering*, 12(3):876–888.

[Mathew et al., 2015] Mathew, N., Smith, S. L., and Waslander, S. L. (2015). Planning paths for package delivery in heterogeneous multirobot teams. *IEEE Transactions on Automation Science and Engineering*, 12(4):1298–1308.

[Montoya-Torres et al., 2015] Montoya-Torres, J. R., Franco, J. L., Isaza, S. N., Jimnez, H. F., and Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, 79:115–129.

[Nagy and Anderson, 2016] Nagy, G. and Anderson, J. (2016). Active recruitment mechanisms for heterogeneous robot teams in dangerous environments. In *Canadian Conference on Artificial Intelligence*, pages 276–281. Springer.

[Neapolitan and Naimipour, 2010] Neapolitan, R. E. and Naimipour, K. (2010). *Foundations of algorithms.* Jones & Bartlett Learning.

[Nunes and Gini, 2015] Nunes, E. and Gini, M. L. (2015). Multi-robot auctions for allocation of tasks with temporal constraints. In *AAAI*, pages 2110–2116.

[Öztürk and Kuzucuoğlu, 2015] Öztürk, S. and Kuzucuoğlu, A. E. (2015). Optimal bid valuation using path finding for multi-robot task allocation. *Journal of Intelligent Manufacturing*, 26(5):1049–1062.

[Pereira et al., 2015] Pereira, T., Veloso, M., and Moreira, A. (2015). Multi-robot planning using robot-dependent reachability maps. In *Robot 2015: Second Iberian Robotics Conference*, pages 189–201. Springer.

[Saad, 1998] Saad, D. (1998). Online algorithms and stochastic approximations. *Online Learning*, 5.

[Sundar and Rathinam, 2015] Sundar, K. and Rathinam, S. (2015). An exact algorithm for a heterogeneous, multiple depot, multiple traveling salesman problem. In *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, pages 366–371. IEEE.

[Whitacre and Bender, 2010] Whitacre, J. and Bender, A. (2010). Degeneracy: a design principle for achieving robustness and evolvability. *Journal of Theoretical Biology*, 263(1):143–153.

[Wu et al., 2017] Wu, D., Zeng, G., Meng, L., Zhou, W., and Li, L. (2017). Gini coefficient-based task allocation for multi-robot systems with limited energy resources. *IEEE/CAA Journal of Automatica Sinica.*