Repeated Auctions for Reallocation of Tasks with Pickup and Delivery upon Robot Failure

Bradford Heap and Maurice Pagnucco

ARC Centre of Excellence for Autonomous Systems and NICTA
School of Computer Science and Engineering
The University of New South Wales
Sydney, NSW, 2052, Australia.
{bradfordh,morri}@cse.unsw.edu.au

The task allocation problem with pickup and delivery is an extension of the widely studied *multi-robot task allocation (MRTA) problem* which, in general, considers each task as a single location to visit. Within the robotics domain distributed auctions are a popular method for task allocation [4]. In this work, we consider a team of autonomous mobile robots making deliveries in an office-like environment. Each robot has a set of tasks to complete, and each task is composed of a pickup location and a delivery location. The robots seek to complete their assigned tasks either minimising distance travelled or time taken according to a global team objective. During execution, individual robots may fail due to malfunctioning equipment or running low on battery power.

A common approach for reacting to task execution delays and changes in the system is to repeatedly auction and redistribute tasks that are not completed, either at certain time intervals or upon each single task completion [6, 12]. Reallocating and replanning tasks can be costly in terms of computational power and time. While arbitrary replanning may not be the most efficient approach, knowing when to reallocate and how much of the system should be reallocated is a challenging problem.

In this paper we consider the reallocation of a failed robot's assigned tasks to the remaining operating robots using sequential single-item auctions (SSI auctions) [11, 8]. We consider two different approaches to the reallocation of tasks amongst the remaining operating robots: a) partial reallocation in which the failed robot's uncompleted tasks are auctioned—this results in the remaining operating robots modifying their existing task execution plans to incorporate additional tasks—b) global reallocation of the failed robot's uncompleted tasks plus all remaining tasks yet to be picked up. This results in a re-assignment of the task set across all remaining operating robots and new task execution plans to be generated. Despite a global reallocation requiring more computation, inter-robot communication and time, it can be expected that this approach would produce lower distance and/or task execution times as more task assignment combinations are considered. However, our empirical results show that partial allocations, on average, produce final results that are equivalent to the results for global reallocation. The aim of this paper is to explore this surprising result.

1 Problem Definition

We expand the problem formalisation given by Koenig *et al.* [9] to include tasks with pickup and delivery. Given a set of robots $R = \{r_1, \ldots, r_m\}$ and a set of tasks $T = \{t_1, \ldots, t_n\}$. A partial solution to the MRTA problem is given by any tuple $\langle T_{r_1}, \ldots, T_{r_m} \rangle$ of pairwise disjoint task subsets: $T_{r_i} \subseteq T$ with $T_{r_i} \cap T_{r_{i'}} = \emptyset$, $i \neq i'$, $\forall i = 1, \ldots, m$. Each task subset T_{r_i} is then assigned to a single robot $r_i \in R$. To determine a complete solution we need to find a partial solution where all tasks are assigned to task subsets: $\langle T_{r_1} \ldots T_{r_m} \rangle$ with $\cup_{r_i \in R} T_{r_i} = T$.

When a robot fails, we remove it from the set of operating robots: $R \leftarrow R \backslash \{r_{fail}\}$. As a consequence of this, if $T_{r_{fail}} \neq \emptyset$, the previous complete solution to the problem $\cup_{r_i \in R} T_{r_i} = T$ no longer holds. A new complete solution can be found by re-assigning the set of tasks assigned to the failed robot $T_{r_{fail}}$ to the remaining operating robots. We wish to investigate if it is better for these remaining operating robots to keep their existing commitments or to start from scratch.

Multi-robot routing is considered the standard testbed for MRTA problems [4]. For tasks with pickup and delivery, the structure of each task t is a tuple $t = \langle l_p, l_d \rangle$ of a pickup location l_p and a delivery location l_d. We consider a robot to be executing a task once it has visited its pickup location up until it reaches its delivery location. Robots may have capacity constraints in the number of tasks they are able to execute at any moment in time. This is representative of real robots which may have a fixed maximum number of items they can carry.

Each robot always has private knowledge of its current location and can calculate the cost λ to travel between locations. The cost to travel between any two locations is equal across all robots. The robot cost $\lambda_{r_i}(T_{r_i})$ is the minimum cost for an individual robot r_i to visit all locations T_{r_i} assigned to it. There can be synergies between tasks assigned to the same robot, such that: $\lambda_{r_i}(\{t\}) + \lambda_{r_i}(\{t'\}) \neq \lambda_{r_i}(\{t\} \cup \{t'\})$. This allows robots, when calculating bids for additional tasks, to consider the cost of completing additional tasks relative to their current commitments. A positive synergy is when the combined cost for a robot to complete two tasks is lower than the individual costs for the robot to complete each task: $\lambda_{r_i}(\{t\} \cup \{t'\}) < \lambda_{r_i}(\{t\}) + \lambda_{r_i}(\{t'\})$.

Team objectives are used to provide additional guidance in the search for solutions to the task allocation that meet certain criteria. Lagoudakis *et al.* discusses team objectives in detail and their application to MRTA [11]. In this work we use two commonly considered team objectives:

MiniSum $\min \sum_{r_i \in R} \lambda_{r_i}(T_{r_i})$ that is to minimise the sum of the paths of all robots in visiting all their assigned pickup and delivery locations.

MiniMax $\min \max_{r_i \in R} \lambda_{r_i}(T_{r_i})$ that is to minimise the maximum distance any individual robot travels.

2 Related Work

Market-based distributed auction algorithms are popular in the robotics community for solving MRTA problems [4, 7]. Common auction types include *combinatorial auctions*, *parallel auctions* and *sequential auctions*. In NP-complete single-round combinatorial auctions [1] each robot bids on all subsets of the tasks on offer. This generates optimal allocations of tasks to robots. However, in most scenarios, the computation tends to be intractable and is generally not feasible for any but the most simple problems. In parallel auctions, robots generate bids for each task in isolation, with no consideration given to inter-task synergies, and the auctioneer then allocates the tasks all at once. The computational complexity is minimal but solutions are often extremely sub-optimal [8].

Sequential single-item auctions which allocate tasks over multiple rounds are a popular middle ground [11, 8]. In each auction round, each robot submits a bid for a task of its choosing, and one task is awarded to the lowest bidder. A key strength of SSI auctions is their ability to build upon inter-task synergies during each task bidding round. However, when robots have few tasks allocated, robots bidding for tasks using SSI auctions have a greedy bias towards tasks that are close to their initial locations. This can see two tasks, that in an optimal solution would be allocated to one robot, split and allocated to two different robots. Previous work on repeated auctions has demonstrated the benefits of reallocating tasks during execution [12, 13]. Additionally, a variety of further improvements and extensions to SSI auctions have been studied which trade off allocation time against overall team costs [7].

2.1 Robot Failure and Task Reallocation

A variety of approaches for task reallocation upon robot failure have been studied. Botelho and Alami [2] consider the problem of robot failure in Smith's contract net protocol (CNP) [14]. In this work, when a robot is about to fail, it sends out an emergency distress message to all other robots and one robot will come to its aid and complete the failed robot's task. However, in this work no inter-task synergies are explored as each additional task is allocated only after the completion of a previous task. This approach also means that, regardless of robot failure, an optimal solution to the task allocation problem is unlikely to be achieved. Dias *et al.* [3] consider various forms of robot failure: communication, partial robot malfunction, and robot death. Their approach to task reallocation is to do a partial reallocation of tasks in the system from the robot that has failed. This is followed by a global reallocation of all tasks at a later moment in time. Gerkey and Mataric [6] deal with robot failures by repeatedly auctioning all uncompleted tasks at set time intervals. While this solution works where tasks are single points, it does not work for tasks with pickup and delivery. Robots may be halfway through the transport of one or more tasks and they would not be able to switch to a different task. Nanjanath and Gini [12] consider repeated auctions upon robot delay. Their approach is that, upon each task completion, all uncompleted tasks across all robots are offered up for reallocation.

Robot failure is closely related to the problem of dynamic task insertion. In dynamic task insertion, additional tasks are inserted into a running system resulting in a need to reallocate tasks. Previous work by Schoenig and Pagnucco [13] has considered SSI auctions with dynamically inserted tasks and compared the costs of robots bidding only for the new task versus a full new auction of all uncompleted tasks. Their results show, despite a large trade-off in computation time, a global reallocation of tasks gives the best results.

2.2 Tasks with Pickup and Delivery

In the field of transport logistics, Fischer, Müller and Pischel apply the CNP to transportation scheduling with fixed time windows [5]. In this work trucks bid for tasks from a central controller and can also make one-for-one swaps with other trucks before they begin to execute their plans. During the execution of plans, the trucks may face traffic delays and, as such, they can locally replan their routes or auction their uncompleted tasks. Their results show that global reallocation of uncompleted tasks provides a large reduction in distance travelled. However, Kohout and Erol argue that Fischer, Müller and Pischel's generation of an initial allocation is poor and therefore global reallocation will produce much better results than local replanning [10]. In their analysis they study problems where multiple items can be transported together and additional jobs are announced sequentially. When a new job is announced, each vehicle bids for the job according to the cost of completing the additional job relative to their existing commitments. To avoid problems where inserting additional tasks has large impacts on the completion time of other tasks, upon each task insertion, already scheduled tasks are permitted to be reallocated to other vehicles. In their empirical analysis they compare this approach to a popular operations research based approach [15]. Overall, they show that their distributed approach is statistically equivalent to this centralised technique.

3 Task Reallocation upon Robot Failure

When a robot detects a problem, for instance, low battery power, it should let other robots know and safely shutdown. A failing robot broadcasts a message to all other operating robots containing its present location and the list of its uncompleted tasks. Any tasks that have not been picked up are able to be immediately auctioned. However, tasks that are under execution when the robot fails must be modified. Because the robot has already visited the pickup location of these tasks, other robots must travel to the location of the failed robot and collect the task from it. To do this the pickup location l_p of all initialised tasks $T_{r_{init}} \subseteq T_{r_{fail}}$ must be updated to the present location of the failed robot $l_p = l_{r_{fail}}$. During reallocation robots continue executing their current task.

A partial reallocation only auctions the task set assigned to the failed robot. The remaining operating robots calculate the bids for these tasks taking into consideration their existing task commitments. Using the cheapest insertion heuristic, each robot's existing task execution plan is modified to include any additional task assignments. This approach allows robots to consider inter-task synergies between their existing commitments and tasks they are bidding for that may not have been considered during the previous allocation. For instance, if in a previous allocation the task $t \in T_{r_{fail}}$ was assigned during the very first round of bidding, no other robot would have been able to consider its synergy with other tasks. Partial reallocations also, generally, have smaller communication overheads than a global reallocation of all tasks. In a distributed SSI auction the total number of messages sent between all robots is $|\bar{T}| * |R|^2$. The number of tasks for auction in a partial reallocation will always be $|T_{r_{fail}}| \leq |T|$.

A global reallocation considers all of the tasks from the failed robot and all uninitialised tasks across all remaining operating robots $\bar{T} \leftarrow T_{r_{fail}} \cup T \backslash T_{init}$. Uninitialised tasks are tasks where a robot has not visited the pickup location of the task. Each robot retains the tasks that it has picked up $T_r \leftarrow T_{r_{init}}$. When calculating bids for additional tasks, the completion of these retained tasks is taken into consideration. It is important to note that, the previous task allocations were generated under conditions and constraints in the number of robots available and the tasks available for bidding in each round which have now changed. Allowing robots to give up tasks previously allocated under these prior circumstances enables them to completely regenerate their plans and consider previous unexplored inter-task synergies. As a result, we expect that this approach will generate solutions with lower costs than partial reallocations.

4 Experiments

To contrast the differences between partial and global reallocations, we simulate an office-like environment with 16 rooms (in a 4x4 grid), each containing four interconnecting doors that can be independently opened or closed to allow or restrict travel between rooms. This environment has become the standard testbed in recent literature [9]. In each experiment, the doors between different rooms and the hallway are either open or closed. We test on 25 randomly generated configurations of opened and closed doors with each robot starting in a different random location. Robots can only travel between rooms through open doors and they cannot open or close doors. However, it is guaranteed that there is at least one path between each room and every other room. For each configuration we test with 10 identical robots, 60 tasks, and from two to eight robot failures. We compare these results to an initial cost which is the cost to complete all tasks without any robot failures or reallocations of tasks. Robots fail at random intervals after arriving at a pickup or dropoff location. We test with the MiniSum and MiniMax team objectives and with capacity constraints of 1, 3, and 5.

In both reallocation approaches the total distance travelled decreases as the capacity constraint is increased. This is not surprising as, the larger the capacity constraint, the more flexibility robots have in executing multiple tasks in unison. We also note, as the number of robots failing increases, the distance required for the remaining robots to travel increases.

To further analyse these results we looked at the distribution of the final costs for both reallocation techniques. Fig. 1 is a plot of the distribution of one standard deviation around the mean for the capacity constraint of one. The other two capacity constraints tested follow a similar trend. One can observe in this plot that, as the number of failed robots increases, the standard deviation becomes much larger. This indicates that in some of the configurations tested the final costs remained low despite the large number of robot failures, however, in other configurations the final costs became extremely large. When the number of robot failures remains less than four there is very little difference in means and distributions between partial and global reallocations. However, as the number of robot failures becomes large there is a clear benefit in using partial reallocations.

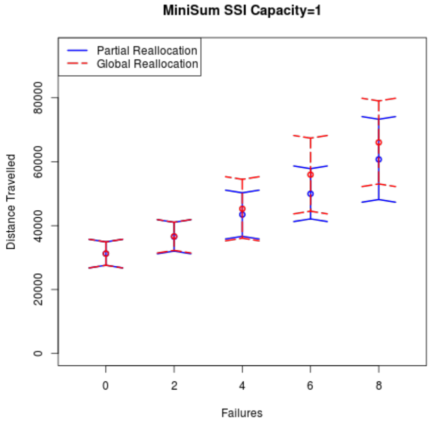

Fig. 1: Distribution of MiniSum team objective results for capacity 1.

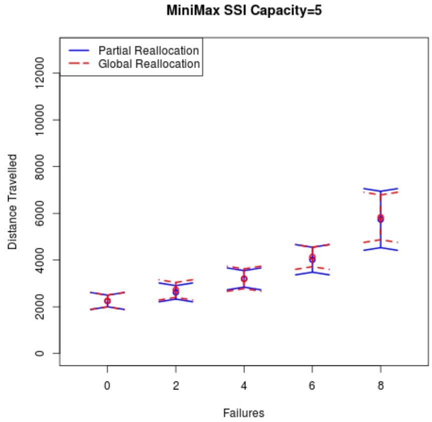

Fig. 2: Distribution of MiniMax team objective results for capacity 5.

t_{r_1}	t'_{r_1}	$\underset{\longleftarrow}{r_1}$	$\underset{\longrightarrow}{r_2}$	t_{r_2}	t'_{r_2}	$\underset{\longrightarrow}{r_3}$	t_{r_3}	t'_{r_3}

Fig. 3: Reallocation example with three robots and six tasks. r_1 is travelling to the left, r_2 and r_3 to the right. Tasks are point locations $l_p = l_d$.

For the MiniMax team objective, again, there are trends that, as the capacity constraint is increased, the cost decreases and, as the number of failed robots increases, the cost increases. The plot in Fig. 2 shows one standard deviation around the mean for the MiniMax team objective with a capacity of five initialised tasks at any one time. This plot shows a different distribution to that of the MiniSum team objective. Our first observation is that the results for both partial and global reallocations completely overlap. At no point does one technique offer an advantage over the other. Our second observation is that the standard deviation remains small in all but the extreme case of eight failed robots. Our logs suggest this is due to robots with lower costs than the robot with the maximum cost taking on additional tasks from failed robots without impacting the overall maximum cost.

Overall, these results are surprising. Our expectations were that global reallocation would outperform partial reallocation. For instance, consider three robots travelling along a horizontal line. The first robot is travelling to the left, the middle robot to the right, and a third robot also travelling to the right (Fig. 3). The first robot then fails. In a partial reallocation the middle robot would need to continue doing tasks to its right and then complete the tasks on its left. However, in a global reallocation the middle robot could give up its tasks to the right and travel to the left and complete the failed robot's tasks. In this situation you would expect that the global reallocation would result in a smaller task cost than the partial allocation.

Table 1: Mean MiniSum Team Objective Computation Time (s) (percentage increase in time after reallocation compared to initial time in brackets).

Capacity	Failures	Initial Allocation	Partial Reallocation	Global Reallocation
1	2	209	250 (19.3%)	330 (57.6%)
1	4	209	265 (25.9%)	439 (110.0%)
1	6	209	280 (33.3%)	510 (143.0%)
1	8	209	302 (44.3%)	579 (176.4%)
3	2	211	261 (23.8%)	339 (60.9%)
3	4	211	284 (36.1%)	419 (100.8%)
3	6	211	314 (48.4%)	493 (133.5%)
3	8	211	340 (61.3%)	553 (162.4%)
5	2	213	269 (25.3%)	343 (59.4%)
5	4	213	307 (44.2%)	433 (103.1%)
5	6	213	358 (68.7%)	509 (140.0%)
5	8	213	398 (86.7%)	615 (188.9%)

Finally, we consider the overall computation time required for generating an initial allocation and for reallocation. Table 1 presents the mean timings for the MiniSum team objective (we omit the MiniMax team objective data as it is nearly identical). These results show that the time required for partial reallocation is much lower than for global reallocation. We note that, for the initial allocation, the capacity constraint has almost no impact on the time required. For partial reallocation, as the capacity constraint increases, there is a smaller increase in the time taken, however, this trend is not seen in global reallocation. As the number of failed robots increase, both reallocation techniques require more computation time. In particular, the time required for global reallocation grows at a very rapid rate as the number of failed robots increases. Overall, from this data and the previous results, we can conclude that partial reallocations are a viable technique for handling robot failure. Their resultant costs are at least equal to global reallocation and they have much faster computation times.

5 Discussion

Our experimental results are unexpected and appear to contradict previous results on reallocation of tasks using auctions. We can classify previous work into two groups, the first being work that presents algorithms for task reallocation [2, 3, 6, 12], and the second dealing with task reallocation upon new task insertion [13, 16]. We are unaware of previous work comparing partial and global reallocation of tasks using repeated auctions.

In our related work section we stated that dynamic task insertion is a very similar problem. Naively, one can assume that adding a new task to the set of tasks: $T \leftarrow T \cup \{t_{new}\}$ and removing a robot from the set of robots: $R \leftarrow R \backslash \{r_{fail}\}$ would affect the task allocation problem in the same way as the complete solution: $\cup_{r_i \in R} T_{r_i} = T$ relies on both R and T. However, a key difference between dynamic task insertion and robot failure is the location of the tasks for reallocation. Most dynamic task insertion approaches assume that the task location is random. However, in the case of a robot failure, despite a robot failure occurring at random, the tasks for auction are not randomly distributed. They are generally geographically close and also contain tight inter-task synergies.

6 Conclusion

This work has studied task reallocation in a robot team upon the failures of teammates. We explored two techniques for task reallocation: partial reallocation which considers only a subset of the total tasks in the system; and, global reallocation which considers almost all tasks in the system. Our empirical evaluations show that, despite global reallocation considering more inter-task synergies, partial reallocations, on average, performed at least as well. Furthermore, partial reallocations require much less computation time.

References

1. Berhault, M., Huang, H., Keskinocak, P., Koenig, S., Elmaghraby, W., Griffin, P.M., Kleywegt, A.J.: Robot exploration with combinatorial auctions. In: IROS. pp. 1957–1962 (2003)
2. Botelho, S., Alami, R.: M+: A scheme for multi-robot cooperation through negotiated task allocation and achievement. In: ICRA. pp. 1234–1239 (1999)
3. Dias, M.B., Zinck, M., Zlot, R., Stentz, A.: Robust multirobot coordination in dynamic environments. In: ICRA. pp. 3435–3442 (2004)
4. Dias, M.B., Zlot, R., Kalra, N., Stentz, A.: Market-based multirobot coordination: A survey and analysis. Proceedings of the IEEE 94(7), 1257–1270 (2006)
5. Fischer, K.: Cooperative transportation scheduling: An application domain for dai. Applied Artificial Intelligence 10(1), 1–34 (1996)
6. Gerkey, B.P., Matarić, M.J.: Sold!: auction methods for multirobot coordination. IEEE Transactions on Robotics 18(5), 758–768 (2002)
7. Koenig, S., Keskinocak, P., Tovey, C.A.: Progress on agent coordination with cooperative auctions. In: AAAI (2010)
8. Koenig, S., Tovey, C.A., Lagoudakis, M.G., Markakis, E., Kempe, D., Keskinocak, P., Kleywegt, A.J., Meyerson, A., Jain, S.: The power of sequential single-item auctions for agent coordination. In: AAAI. pp. 1625–1629 (2006)
9. Koenig, S., Tovey, C.A., Zheng, X., Sungur, I.: Sequential bundle-bid single-sale auction algorithms for decentralized control. In: IJCAI. pp. 1359–1365 (2007)
10. Kohout, R.C., Erol, K.: In-time agent-based vehicle routing with a stochastic improvement heuristic. In: AAAI. pp. 864–869 (1999)
11. Lagoudakis, M.G., Markakis, E., Kempe, D., Keskinocak, P., Kleywegt, A.J., Koenig, S., Tovey, C.A., Meyerson, A., Jain, S.: Auction-based multi-robot routing. In: Robotics: Science and Systems. pp. 343–350 (2005)
12. Nanjanath, M., Gini, M.L.: Repeated auctions for robust task execution by a robot team. Robotics and Autonomous Systems 58(7), 900–909 (2010)
13. Schoenig, A., Pagnucco, M.: Evaluating sequential single-item auctions for dynamic task allocation. In: Australasian Conference on Artificial Intelligence. pp. 506–515 (2010)
14. Smith, R.G.: The contract net protocol: High-level communication and control in a distributed problem solver. IEEE Trans. Computers 29(12), 1104–1113 (1980)
15. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations research 35(2), 254–265 (1987)
16. Van Duin, J., Tavasszy, L., Taniguchi, E.: Real time simulation of auctioning and re-scheduling processes in hybrid freight markets. Transportation Research Part B: Methodological 41(9), 1050–1066 (2007)