

Planning to fail—Using reliability to improve multirobot task allocation

Stephen B. Stancliff^a and John M. Dolan^b

^aEnergetics Technology Center of Southern Maryland, La Plata, USA;

^bThe Robotics Institute, Carnegie Mellon University, Pittsburgh, USA

ABSTRACT

The reliability of individual robots influences the success of multirobot missions. When one robot fails, others must be retasked to complete the failed robot's tasks. This increases the failure likelihood for these other robots. Existing multirobot task allocation systems consider robot failures only after the fact, via replanning. In this paper we show that mission performance for multirobot missions can be improved by using knowledge of robot failure rates to inform the initial task allocation.

Keywords: Mobile robots, multirobot systems, reliability, task allocation, mission planning

1. INTRODUCTION

In order for multirobot systems to operate successfully in the real world, they must be able to deal with the inevitable failures of team members. Reference 1 summarizes historical failure data for small field robots and reveals that the robots were either broken or under repair approximately half of the time. Given such poor reliability, it seems likely that any multirobot planning system that does not consider robot failures will perform poorly for real-world missions.

There has been a substantial amount of work in the area of detecting and recovering from robot failures (e.g., Refs 2–4), and several multirobot mission planning systems provide mechanisms for reallocation of tasks among surviving team members after a robot failure (e.g., Refs 5–7). However, all of these methods are reactive rather than predictive, dealing with failure only after it occurs. While it is important in dynamic real-world environments to be able to replan after robot failure, we hypothesize that it should also be useful to consider the probabilities of robot failures when making initial task assignments. As an abstract example, one would not want to assign a robot that has a high chance of failure to a critically-important task.

The only prior work we are aware of that addresses the use of robot reliability to improve multirobot task allocation is Ref. 8. That paper demonstrated that brute-force multirobot task allocation can be improved by using knowledge of robot failure rates in order to consider the effects of backup plans on mission performance. While the results in that paper are significant, their applicability is limited, since real-world multirobot problems are usually too complex for brute-force planning methods. In this paper we extend that work by examining whether similar improvements in plan selection can be obtained when using a greedy planner.

Our results show that, indeed, our method can produce substantial improvements in plan selection even with a greedy planner, but that the amount of improvement depends strongly on the nature of the planner being modified. In particular, highly-greedy planners may prune the search space too much for the additional reliability information to provide much benefit.

A key feature of our method for incorporating reliability is that we do not introduce arbitrary reliability requirements into the mission specifications, but instead show that reliability needs to be considered in order to optimize an *existing* utility metric. We therefore conclude that robot reliability needs to be considered in multirobot task allocation even when reliability is not itself an explicit performance metric for the mission.

Contact: stancliff@etcmd.com, jmd@cs.cmu.edu

2. BACKGROUND

Mission planning for multirobot systems is an active research area with substantial literature describing many different mission planners. One measure by which the performance of multirobot mission planning methods can be compared is how they deal with robot failures. Reference 6, for example, describes a mission planning system that is able to recover from robot failure because tasks can be reallocated. In this system, tasks are “auctioned” to the robot with the highest bid (or lowest, depending on the utility function). This system allows for task reallocation during the mission when new information changes task valuations. For instance, if a robot suffers a component failure that impairs its ability to perform its assigned tasks, it will change its valuation for those tasks, and it can then subcontract tasks to another robot that has a better valuation for those tasks.

While it is important to recover from robot failures, it would be better to minimize the likelihood of such failures in the first place. The mission planning system can influence these likelihoods since the initial assignment of tasks to robots plays a role in determining the probabilities of robot failures during the mission. For example, assigning a robot with a damaged drive motor to a task that requires it to travel a long distance results in a higher probability of that robot failing than if it were assigned to a task that required less travel.

One way to incorporate such reliability concerns into multirobot mission planning would be to introduce a reliability component into the utility function used by the planner. Reference 9, for example, provides methods for predicting probabilities of failure for mobile robots during mission tasks. Incorporating such probabilities directly into the planner utility function is unsatisfying for two reasons. The first is the incommensurability of different components of a utility measure—how do we combine dollars spent, meters travelled, and probability of failure into a single metric? The second is that establishing a numeric reliability requirement is itself a difficult problem that has been minimally explored for the mobile robot domain—i.e., how do we decide if the reliability requirement for a mission should be 95% rather than 96%?

In order to avoid these difficulties, we take a different approach: rather than devising utility metrics that incorporate reliability, we instead look at how robot reliability affects the utility metrics already being used. In plain language, we are *not* taking “Find the solution with the shortest time” and turning it into “Find the solution with the shortest time that also meets reliability level X ” but instead turning it into “Find the solution with the shortest *expected* time” where the expected time takes into account the alternative outcomes that occur when robots fail.

3. ILLUSTRATING EXAMPLE

Consider a simple multirobot exploration mission with two identical robots and two locations to be visited (Fig. 1). The goal of the mission is for all target locations to be visited in any order by any robot in the shortest total mission time. Time is assumed here to be proportional to distance traversed.

Each robot is defined by an (x, y) location and a reliability P_t , which is the probability of surviving a one-unit traverse. Each target location is defined by its (x, y) location. The robot and target parameters used for this example are listed in Table 1.

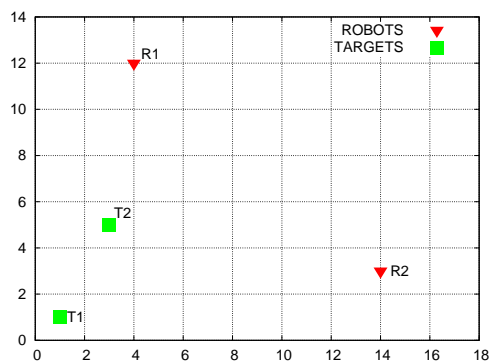


Figure 1. Exploration mission.

Table 1. Robot and target parameters.

	x	y	P_t
Robot 1	4	12	0.99
Robot 2	14	3	0.99
Target 1	1	1	—
Target 2	3	5	—

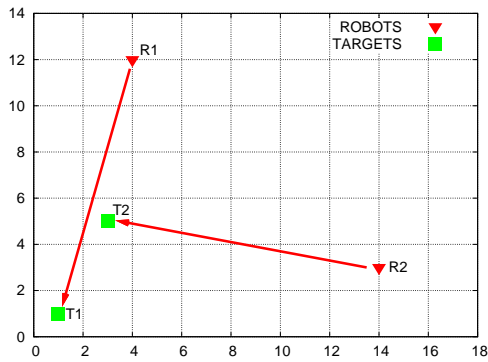
Table 2. Plan durations.

Plan	$d(R1)$	$d(R2)$	d_{plan}
A ($R_1T_1 + R_1T_2$)	15.9	0	15.9
B ($R_1T_1 + R_2T_2$)	11.4	11.2	11.4
C ($R_2T_1 + R_1T_2$)	7.62	13.2	13.2
D ($R_2T_1 + R_2T_2$)	0	17.6	17.6
E ($R_1T_2 + R_1T_1$)	11.5	0	11.5
F ($R_2T_2 + R_2T_1$)	0	15.7	15.7

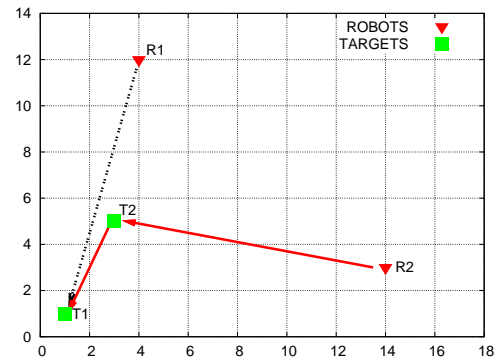
For a small number of robots and tasks, it is feasible to exhaustively enumerate the possible task assignments and then calculate the distance that each robot must traverse to accomplish each plan (Table 2). The plan duration (d_{plan}) is equal to the greatest distance that any robot travels during that plan. The plan with the smallest duration is then chosen. In this example Plan B (Fig. 2a) would be chosen.

Now consider what happens when a robot fails while executing this plan. If Robot 1 fails, then Robot 2 is assigned to visit Target 1 after visiting Target 2 (Fig. 2b). If Robot 2 fails, then Robot 1 is assigned to Target 2 after visiting Target 1 (Fig. 2c). We assume here that tasks are not interrupted, so that new targets are assigned to surviving robots only after the completion of their current tasks.

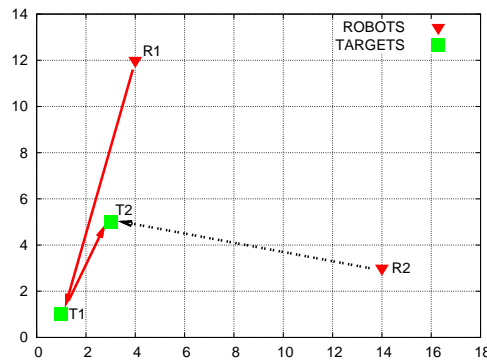
Examining Fig. 2, we see that while the backup plan in the first case is reasonable, the second one is inefficient



(a) Naively-chosen plan (Plan B).



(b) Backup for Plan B when Robot 1 fails (dashed line represents robot failure).



(c) Backup for Plan B when Robot 2 fails.

Figure 2. Chosen plan and backups.

because the optimal plan for Robot 1 to visit both locations is to visit Target 2 first. This inefficiency is what is being missed when reliability is not considered in the original task allocation.

In order to take into consideration both primary and backup plans, we calculate an expected duration (d_{exp}) for each primary plan by multiplying the duration and probability for each alternative outcome and then adding these together. Here we assume that the duration for a failed robot–task pairing is equal to the entire duration of that task. In other words, replanning only happens after a robot has failed to meet a deadline.*

For Plan B this gives:

$$\begin{aligned}
&\text{Primary plan duration} = 11.4 \\
&\text{Backup 1 (Fig. 2b) duration} = 15.7 \\
&\text{Backup 2 (Fig. 2c) duration} = 15.9 \\
&P(R_1T_1 \text{ success}) = (P_t)^{11.4} = 0.891 \\
&P(R_2T_2 \text{ success}) = (P_t)^{11.2} = 0.894 \\
&P(\text{Primary plan}) = P(R_1T_1) \times P(R_2T_2) = 0.797 \\
&P(\text{Backup 1}) = P(\overline{R_1T_1}) \times P(R_2T_2) = 0.097 \\
&P(\text{Backup 2}) = P(R_1T_1) \times P(\overline{R_2T_2}) = 0.095 \\
\\
d_{exp} &= \frac{\sum P_i \times d_i}{\sum P_i} \\
&= \frac{(0.797)(11.4) + (0.097)(15.7) + (0.095)(15.9)}{(0.797 + 0.097 + 0.095)} \\
&= 12.3
\end{aligned}$$

The denominator in the last equation normalizes the expected duration to account for the fact that we are ignoring those cases where both robots fail. Our result is therefore an expected duration *assuming that the mission succeeds*.

Repeating these calculations for each primary plan gives the results shown in Table 3. We see here that a different plan (Plan E—Fig. 3a) has the shortest expected duration. Looking at the backup plans for Plan E (Fig. 3b–3c), we see that these plans are optimal plans for the surviving robot in each case. It is this optimality of backup plans that is being missed when reliability is ignored in the initial task allocation.

Table 3. Naive and expected durations.

Plan	d_{naive}	d_{exp}
A ($R_1T_1 + R_1T_2$)	15.9	15.8
B ($R_1T_1 + R_2T_2$)	11.4	12.3
C ($R_2T_1 + R_1T_2$)	13.2	13.2
D ($R_2T_1 + R_2T_2$)	17.6	16.8
E ($R_1T_2 + R_1T_1$)	11.5	11.9
F ($R_2T_2 + R_2T_1$)	15.7	15.2

*This assumption fails when all of the following conditions are met: (1) A robot has a failure that prevents completion of its task but does not prevent communication with other robots; (2) the failed robot is capable of recognizing that a failure has occurred; (3) the mission requirements allow the robot to communicate its failure (i.e., not a mission where terrain blocks communication, not operating under radio silence, etc.). For most real-world missions, these conditions will not all be met, so the failure of one robot will be detected when it fails to communicate success at some time greater than or equal to the expected task completion time. For realistic missions, therefore, the assumption is conservative.

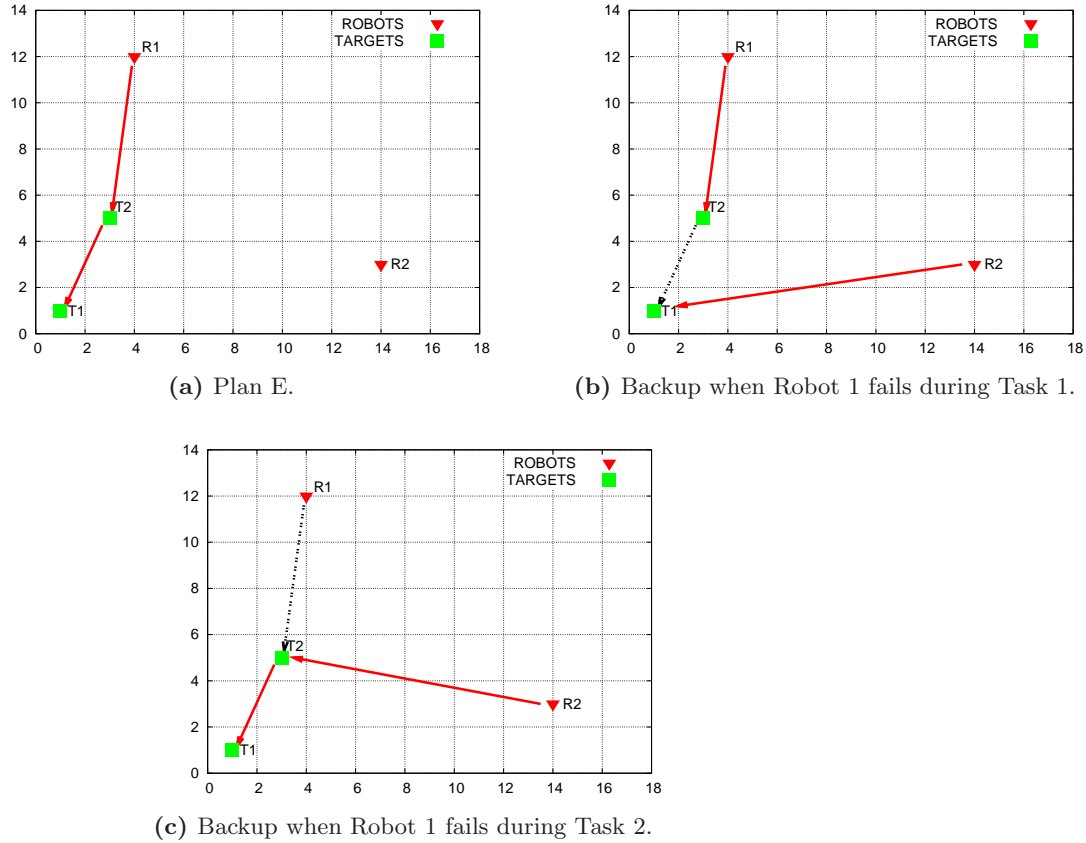


Figure 3. Plan with shortest expected duration (Plan E), and backup plans.

4. SIMULATION RESULTS

The preceding example demonstrated a case where the plan chosen by a reliability-naïve planner is not the optimal plan. In order to determine whether this is an anomalous case or if it is the norm, we implemented the methods described in the preceding section in software. This software was used to evaluate mission configurations with varying world sizes, team sizes, target counts, and robot reliabilities. Our work here is similar to that in Ref. 8, except that in that paper we used a complete planner. Here we examine whether similar benefits can be gained from incorporating reliability knowledge into an incomplete planner.

4.1 Greedy Planner

The planner used here makes use of two heuristics in order to prune the number of plans considered. First, the number of task orderings is limited to a (configurable) fixed number of random orderings. Second, the assignment of robots to tasks is performed greedily, with tasks being assigned sequentially and independently.

As an example, consider a two-robot, two-task exploration mission. A complete planner would evaluate each of the six task allocations:

$$R_1T_1 + R_1T_2$$

$$R_1T_1 + R_2T_2$$

$$R_1T_2 + R_1T_1$$

$$R_1T_2 + R_2T_1$$

$$R_2T_1 + R_2T_2$$

$$R_2T_2 + R_2T_1.$$

If the planner is configured to consider one task ordering and randomly selects the ordering T_1T_2 , then it would first evaluate the possible allocations for T_1 :

$$\begin{aligned} R_1T_1 \\ R_2T_1, \end{aligned}$$

and then evaluate the possible allocations for the second task given the allocation already chosen for the first task. For instance, if R_1T_1 is chosen above, then the following allocations will be considered:

$$\begin{aligned} R_1T_1 + R_1T_2 \\ R_1T_1 + R_2T_2. \end{aligned}$$

In this example, our planner evaluates two partial plans and two complete plans, whereas the complete planner evaluates six complete plans.

4.2 Incorporating Reliability

In order to incorporate reliability into this planner we perform an expected value calculation on each complete plan considered by the basic planner. For the two-robot, two-task mission, we first compare

$$\begin{aligned} \text{val}(R_1T_1) \\ \text{val}(R_2T_1), \end{aligned}$$

then (with R_1T_1 chosen)

$$\begin{aligned} \text{expval}(R_1T_1 + R_1T_2) \\ \text{expval}(R_1T_1 + R_2T_2). \end{aligned}$$

In earlier experiments, we used *expval()* in both steps but found that considering backup plans for partial plans led to poorer plans as often as it led to better ones. Considering expected values only for complete plans guarantees that the reliability-enhanced planner will never choose a worse plan than the baseline planner since both are choosing from the same set of complete plans.

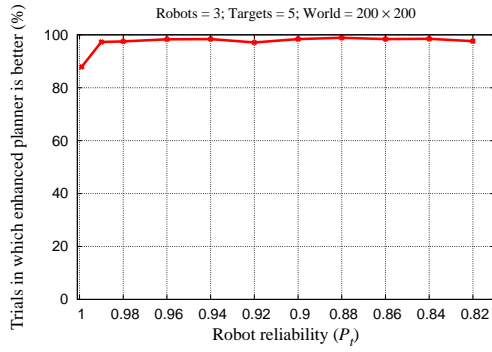
4.3 Results

For each mission configuration (number of robots; target count; robot reliability; world size), numerous trials were run, with robot and target locations randomized for each trial. The two outputs of this simulation are the percentage of trials in which the reliability-enhanced planner chooses a better plan (Fig. 4a) and the average increase in mission duration when reliability is used (Fig. 4b). We then combine these values together to obtain an average expected mission duration when reliability is not used (Fig. 4c).

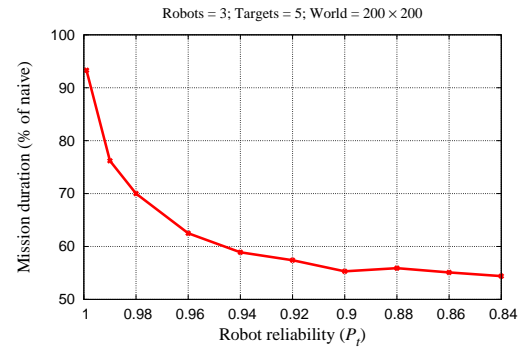
Fig. 4c shows that even for a fairly uncomplicated mission (three robots, five tasks, a small world) with fairly reliable robots ($P_t = 0.999$), incorporating backup plans provides about 5% average improvement in mission duration, and that the improvement increases to almost 50% for less-reliable robots.

Fig. 5 shows similar results for a more complicated scenario (four robots, eleven tasks) with varying world sizes and robot reliabilities. These results make intuitive sense because both larger world size and lower robot reliability increase the likelihood of robot failure during the mission, and with higher likelihood of robot failure the influence of backup plans on the expected duration is greater.

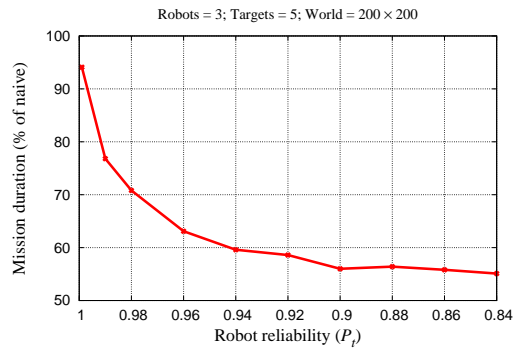
Fig. 6a shows the effect of increasing target count. This graph is interesting because of the nonmonotonicity, which appears to show that reliability information is less useful as the number of targets increases beyond some point. However, this graph reveals a limitation of the underlying planner rather than of the reliability augmentation. What is happening here is that the planner prunes the search space too much in the higher-complexity cases, which leaves the reliability-augmented planner to choose from a set of poor solutions. Fig. 6b demonstrates what happens when the number of task orderings is increased—the valley shifts to the right because the number of task orderings is now sufficient to handle a larger number of tasks. This phenomenon is discussed further in the next section.



(a) How often reliability-enhanced planner chooses a better plan.



(b) Mission duration when reliability chooses a better plan.



(c) Expected mission duration when reliability is used.

Figure 4. Relative performance of reliability-unaware and reliability-augmented planners.

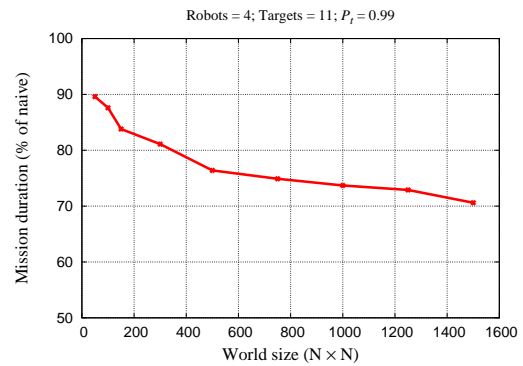


Figure 5. Effects of robot reliability and world size on expected improvement.

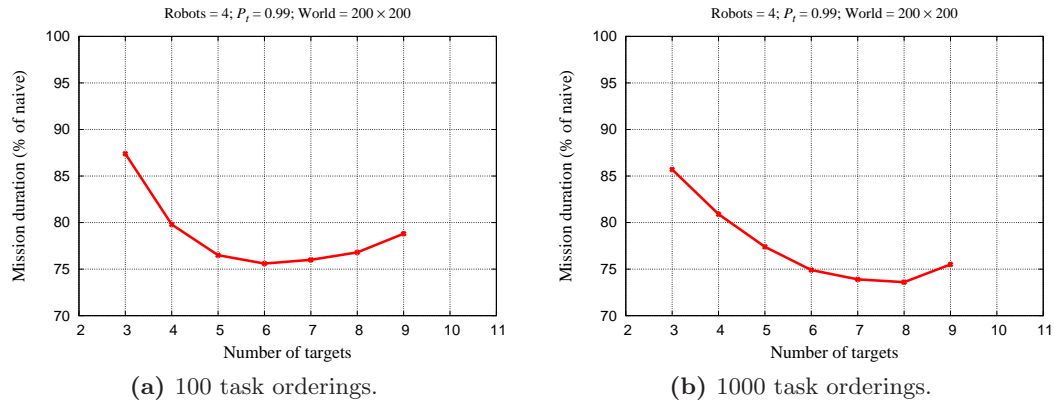


Figure 6. Expected mission duration vs. task count.

4.4 Effect of Planner Greediness

Earlier experiments using a highly-greedy planner that considered only a single task ordering showed a very small improvement when backup plans were considered. The current planner allows us to vary the number of task orderings that the planner evaluates in order to investigate the relationship between the number of task orderings considered, the relative improvement in plan duration, and the computational time required (Fig. 7).

This figure shows that the benefit of using reliability information is strongly related to the number of possible allocations that are explored. However, the improvement in planner performance levels off above some point, while the computational complexity continues to increase steeply, so there is a point of diminishing returns. The important observation here is that if only a small number of possible plans are explored, considering backup plans will provide relatively little improvement.

This also explains the nonmonotonic behavior seen in Fig. 6. When the number of task orderings considered is too small relative to the total number of possible task orderings, the effectiveness of incorporating reliability is reduced. So in Fig. 6 where a fixed number of task orderings was used, the relative improvement from considering reliability begins to decrease once the complexity of the mission scenario increases above some point.

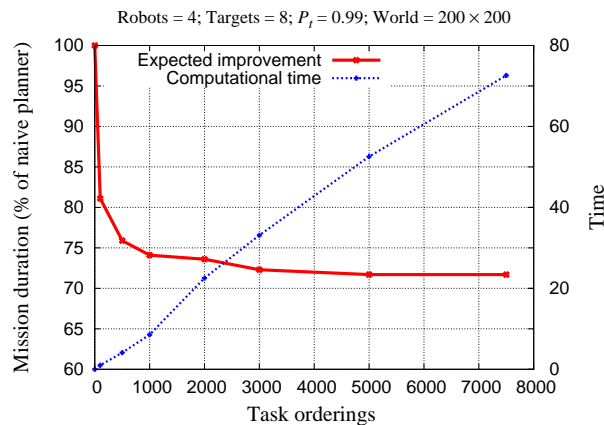


Figure 7. How the number of plans tested affects performance[†]

5. SUMMARY

In this paper, we demonstrated that knowledge of robot reliability can be used to improve multirobot task allocation when using greedy planners. Our results show that the amount of improvement, as well as the computational complexity of achieving that improvement, are highly dependent on the nature of the underlying planner being augmented.

A key feature of these results is that we have slipped reliability “in through the back door” in that we have not introduced arbitrary reliability requirements into the mission specifications but have instead shown that reliability needs to be considered in order to optimize an *existing* utility metric (in these examples, total mission duration). We conclude that robot reliability needs to be considered in generating multirobot task allocations even when reliability is not itself an explicit performance metric for the mission.

Two observations which we can make based on these experiments are:

- Backup plans should be considered only for complete plans. Considering expected values of partial plans is just as likely to worsen results as improve them.
- Considering backup plans does not provide much improvement when the underlying planner only considers a small subset of the possible plans.

Our results also suggest when it may not be useful to consider backup plans. A completely distributed planner, for instance, where no single agent has knowledge of multiple complete plans, would not lend itself to the method employed here. However, it might be possible to take advantage of reliability information in a different way for such a planner, such as incorporating reliability into the utility metric.

Future work suggested by these results is to attempt to further classify the kinds of planners that can and cannot benefit from this method of incorporating reliability and to identify alternative methods for incorporating reliability knowledge into those planners for which the current method provides poor results.

REFERENCES

- [1] Carlson, J. and Murphy, R., “Reliability analysis of mobile robots,” *Proc. IEEE Int. Conf. on Robotics and Automation*, 274–281 (Sep. 2003).
- [2] Parker, L., “Alliance: An architecture for fault tolerant multirobot cooperation,” *IEEE Trans. Robotics and Automation*, 220–240 (Apr. 1998).
- [3] Kaminka, G. and Tambe, M., “Social comparison for failure detection and recovery,” *Proc. Intelligent Agents IV*, 127–141 (1997).
- [4] Verma, V., Gordon, G., Simmons, R., and Thrun, S., “Real-time fault diagnosis [robot fault diagnosis],” *IEEE Robotics and Automation Magazine*, 56–66 (June 2004).
- [5] Brumitt, B. and Stentz, A., “GRAMMPS: A generalized mission planner for multiple mobile robots in unstructured environments,” *Proc. IEEE Int. Conf. on Robotics and Automation*, 1564–1571 (May 1998).
- [6] Dias, B., Zinck, M., Zlot, R., and Stentz, A., “Robust multirobot coordination in dynamic environments,” *Proc. IEEE Int. Conf. on Robotics and Automation*, 3435–3442 (May 2004).
- [7] Gerkey, B. and Mataric, M., “Sold!: Auction methods for multirobot coordination,” *IEEE Trans. Robotics and Automation*, 758–768 (Oct. 2002).
- [8] Stancliff, S., Dolan, J., and Trebi-Ollennu, A., “Planning to fail – Reliability needs to be considered *a priori* in multirobot task allocation,” *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics* (Oct. 2009).
- [9] Stancliff, S., Dolan, J., and Trebi-Ollennu, A., “Mission reliability estimation for multirobot team design,” *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2206–2211 (Oct. 2006).

[†]Time is represented nondimensionally because the relative computational cost is what matters, not the actual time that our suboptimal implementation requires.