Proceedings of the 2007 IEEE/RSJ International
Conference on Intelligent Robots and Systems
San Diego, CA, USA, Oct 29 - Nov 2, 2007

Scheduling for Humans in Multirobot Supervisory Control

Sandra Mau, John Dolan

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, USA
{smau | jmd}@ri.cmu.edu

***Abstract*—This paper describes efficient utilization of human time by two means: prioritization of human tasks and maximizing multirobot team size. We propose an efficient scheduling algorithm for multirobot supervisory control that helps complete a mission faster. The proposed algorithm is superior to existing algorithms by prioritizing human tasks such that robots can regain autonomous control sooner. In simulations of a multirobot area surveying problem, we show that the rate of area coverage is much higher using our algorithm compared to first-in-first-out. We also show that the use of different scheduling algorithms can affect the maximum number of robots a human can manage on a team.**

Another significant finding related to maximum team size is that the size is always the same or higher than an often-cited estimate known as *fan-out* [5]. Since fan-out is derived from an ideal, average case, simulations show that the upper bound on team size is higher than that predicted by the fan-out equation. Fan-out is actually a lower bound on the maximum team size for any practical situation (i.e., where task lengths and periodicity may vary or when robots are heterogeneous).

I. Introduction

Despite the fact that robot autonomy is increasing for multirobot teams performing complex missions, most still operate under human supervisory control. As technology advances, the ideal in supervisory control is to increase the robot-to-human ratio, since more robots working together can potentially decrease the time it takes to accomplish a mission. However, with more robots on a team, a human supervisor's attention becomes increasingly divided. If many robots require human attention simultaneously, it is useful for these tasks to be scheduled for the human efficiently.

Most scheduling problems in real-world applications are unique in that they are highly task-dependent and objective-driven. One particular task characteristic of scheduling for human-multirobot supervisory control is that human tasks become precedence constraints for the robots' remaining tasks – the robot must wait until the human is finished using or repairing it before regaining autonomy and returning to its own tasks. The time during which the robot is waiting for the human to finish is known as *downtime*. One example is when a robot needs human intervention to interpret a situation, such as teleoperation for navigational guidance, or use of a robot as a remote science tool for experts.

Since downtime means that robots must wait for the human, scheduling for the human to lower downtime results in robots regaining autonomy sooner. We have developed an on-line, semi-preemptive, heuristic scheduling algorithm to reduce downtime. This quick and efficient scheduling algorithm, dubbed "double Shifted Shortest Processing Time" (dSSPT), is ideal for human supervisory control. It can be used to prioritize tasks for the human supervisor such that the robot team performs its mission more efficiently, with more robot tasks completed sooner.

Having more robots on the team can allow robots to share tasks, also resulting in more tasks performed sooner. For example, in an area coverage problem, two robots can cover an area faster than just one. However, since human tasks are a precedence constraint for robot tasks, the time it takes for the entire mission (makespan) can be limited by how quickly the human can address robot requests. When a human has no idle time left between any of his tasks, he is saturated. This means additional robots will not lower the makespan any further, since the human cannot address those tasks any faster. The minimum number of robots to reach this saturation point is known as the span-of-control, which is also indicative of the maximum number of robots a human should handle.

Previous work by Goodrich and Olsen [5] has provided an estimate for span-of-control called *fan-out*, which is based on offline averages of interaction and neglect times between human and robots. This paper shows that the fan-out averaging technique does not capture variances observed in real problems, and thus the actual span-of-control is larger than the fan-out estimated for most problems. The scheduling algorithm used for prioritizing human tasks also affects the span-of-control on a team. A comparison is made between the span-of-control for several scheduling algorithms (including dSSPT) and the fan-out estimate for simulated missions.

One contribution of this research is to present an on-line scheduling algorithm, dSSPT, which can more efficiently utilize a human's time. dSSPT prioritizes tasks for the human supervisor such that more robot tasks are completed sooner, as shown in the experiments described below. Another contribution is to analyze factors which affect span-of-control, its implications on maximum team size and why there is a discrepancy from Goodrich and Olsen's fan-out estimate (Equation 1). Specifically, we find that the span-of-control increases with the variance of task length, as well as randomness (or aperiodicity) of task occurrence. The scheduling algorithm used also influences span-of-control.

* This work was supported in part by NASA under Cooperative Agreement No. NNA05CP96A.

II. Related Work

A. Scheduling in Human Multirobot Interaction

Currently, most practical applications of robot teams have a very small robot-to-human ratio. For example, the state of the art in robotic space exploration is the Mars Exploration Rover (MER), which requires a human crew of dozens to monitor and control just two robots[1]. The management of a multirobot team where one operator can oversee many robots is a developing topic of research.

Cummings and Mitchell [2] spell out various management issues in the control of multiple UAVs. They find that humans are not good at selecting the best course of action when there are many complex possibilities, even when given a preview of the upcoming timeline. Also, given an increased workload, the amount of time humans plan ahead decreases, resulting in less efficient plans. This indicates that humans are less reliable when it comes to managing larger groups of robots, which suggests more guidance (e.g., scheduling) for humans can help them make better choices.

In the industrial and manufacturing domain, supervisory control has a longer history. MacCarthy et al. [3] list the history of significant works contributing to human factors studies in planning and scheduling since the 1960's, but conclude that knowledge in this field is still lacking. As a result, no methodical process of designing an effective supervisory control system exists. Although the research in this paper does not attempt to solve this complex issue, it examines human factors in scheduling relating to multirobot metrics and attempts to generalize some observations for efficient use of human time.

B. Span-of-Control in Human Multirobot Interaction

One important metric in human multirobot interaction (HMRI) is *span-of-control*, which denotes the number of robots a human can supervise. The related term *fan-out* (FO), coined by Goodrich and Olsen [5], proposes an upper bound on span-of-control as:

$$FO \le \frac{NT}{IT} + 1 \qquad \textbf{(1)}$$

where IT (interaction time) is the time it takes on average for humans to address a robot's request and NT (neglect time) is the average amount of time a robot can act autonomously without human intervention.

This bound is based on two explicit assumptions: all the robots on the team are homogeneous (Crandall later extends the idea to a feasibility test for heterogeneous teams [1]), and average *NT* and *IT* values yield a valid *FO* value. This suggested upper bound can also be interpreted as: *A team should be of a size such that, on average, only one robot will require human attention at any given time*.

The rationale is that, since humans can only attend to one robot at a time, having more robots requiring help simultaneously means that some will not be addressed; in other words, the human is *saturated*. However, in any practical application, the NT and IT values will vary from the estimated mean and requests will not be as ideally periodic as in Figure 1. Therefore, as team size approaches the upper bound in Equation 1, the probability of more than one robot requiring assistance at the same time (*over-subscription*) is very high. In the worst case, all robots could require human attention simultaneously, despite the fact that the average fan-out is within the bounds of Equation 1. Those robots waiting for human attention experience *downtime*.



Figure 1: *Illustration of team size can be estimated by the number of ITs that can "fit" in NT, plus 1 (from Crandall [1])*

Another implicit assumption of fan-out is that the averaging takes place over an infinite time horizon. The upper bound proposed by Goodrich and Olsen is true on an infinite time horizon, where having more ITs than can "fit" into NT will result in tasks not being addressed. For example, Figure 2 (top) illustrates that if one extra task occurs more than average, on the infinite time horizon one task will always be left unfinished. However, most robotic missions have a bounded time horizon and set goals. It is also impossible for humans to work indefinitely long, and even if there are many humans rotating in shifts, funding for the mission is surely limited. When a mission ends, the unfinished task in the previous example will eventually be addressed, with the difference being that the *makespan*, or time to complete all tasks, becomes IT units of time later than without that task (Figure 2 bottom).

The research described in this paper considers what happens to the span-of-control when variance in IT and NT occur (as would be expected in realistic applications).

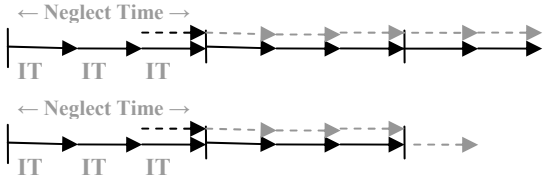


Figure 2: *Top shows 4 tasks arriving in an NT that can only fit 3, so one task never gets addressed if the timeline is infinite. Bottom shows the 4th task addressed at the end for a finite timeline.*

C. Classical Job Shop Scheduling

Scheduling theory has been very well studied for over half a century. The classic scheduling problem is to order a set of tasks such that a certain objective is achieved (usually something is minimized or maximized).

1) Term Definitions

Release Time (r_i): The time at which task *i* arrives to be scheduled.

Processing Time (a_i): Expected time required to complete task *i*.

Interaction Time (IT_i): Similar to processing time, but used in the HMRI application domain to encompass a variety of human interaction, including wait times and processing time [4].

[1] "Mars Exploration Rover Mission: People," NASA, Jet Propulsion Laboratory, Cal Tech. <http://marsrovers.jpl.nasa.gov/people/>.

Idle Time: Time in which the resource is not processing a task.

Completion Time or Makespan: Time it takes to complete all tasks in a schedule.

Wait-time: The time that a task has to wait before being addressed by the resource.

Downtime or Flow Time: The time a task spends in the system, i.e., the wait-time plus the processing time of a task.

Over-subscribed: Having multiple tasks queued.

Preemption: The idea that a task can be interrupted during processing, placed on hold, and then resumed from the point of interruption. When used to describe a scheduling algorithm, it means that such an algorithm can break tasks up into various chunks and schedule the chunks at various times, not consecutively.

On-line vs. off-line: Off-line scheduling occurs when the complete set of tasks and its parameters are known ahead of time. On-line scheduling occurs when there is no prior knowledge of new task arrivals until they are released. The decisions are made based only on the known tasks.

Precedence constraint: A succession relationship between tasks.

2) Some Relevant Scheduling Algorithms

dSSPT is a derivative of the Shortest Processing Time (SPT) family of heuristic algorithms. SPT orders tasks based on processing time to minimize waiting times for other tasks. Previous work in comparing dSSPT to SPT-based algorithms focused mainly on comparing downtime performance [4]. It was shown that dSSPT always outperforms SPT, Shifted-SPT (an online variant) and First-In-First-Out (FIFO) in terms of downtime. The greater the variance in task sizes and the larger the multirobot team size, the greater the improvement of dSSPT, especially over FIFO. Previous analysis was based on a simulation of a uni-processor (i.e., a human operator) scheduling problem where tasks were randomly generated based on given parameters. It did not simulate a full multirobot team problem.

This paper extends that work by simulating the full multirobot problem as an area-coverage for prospecting problem and looking at metrics describing additional team effects, such as rate of area coverage and span-of-control of a team. We also confirm the improved downtime performance of dSSPT over FIFO through this full multirobot simulator. Another novel contribution of this paper is to analyze which human multirobot interaction (HMRI) factors affect span-of-control and how it varies with respect to various parameters and scheduling algorithms.

III. dSSPT Scheduling Algorithm

dSSPT is a scheduling algorithm to prioritize human tasks. Robots are assumed to have a separate task planner that schedules all robot tasks but not the human ones. This is similar to having a decentralized team planner, but with precedence constraints between agents.

A. Assumptions

For simplification, experiments conducted in this paper assume that the human supervisor will take the advice of the schedule provided. All the task parameters, such as processing time and release time, are assumed to be known or accurately estimated by the system. All tasks are equal in priority and each task is independent of the others. The human tasks assumed by this algorithm are semi-preemptive.

B. dSSPT Scheduling Algorithm

Each new task is sorted into the dSSPT-scheduled list by comparing it to existing tasks starting from the end of the list and working towards the beginning. Given a new task j, and an existing task in the list i (with $r_i \leq r_j$), do pair-wise comparisons down the list according to the following:

$$\begin{array}{l} \text{If } a_j + 2 * \Delta t < a_i \text{ is true,} \\ \text{where } \Delta t = \begin{cases} 0 & \text{, if i} \geq 2 \\ r_j - max(a_0, r_i) & \text{, otherwise} \end{cases} \\ \text{then swap tasks } i \text{ and } j. \\ \text{Do this for tasks } i = j-1, j-2, ..., 2, 1 \end{array} \quad \textbf{(2)}$$

Δt can also be expressed as $\Delta t = r_3 - \textit{start time of R2}$, where the *start time of R2* $= max(a_0, r_2)$. The full derivation of this algorithm is given in [4].

C. dSSPT Compared to SPT Variants

dSSPT is essentially a hybrid of an improved Shifted-SPT and the original SPT in the different regimes where those algorithms perform best. For sparse schedules, it uses the improved Shifted-SPT and for dense schedules (three or more tasks queued), it uses SPT. Thus, the drawbacks of dSSPT for dense schedules are like those of SPT in that longer tasks may have to wait for a very long time if there are many tasks queued.

Like the other on-line SPT variant algorithms, dSSPT has time complexity of O(n) and scales well with team size.

D. dSSPT Compared to FIFO

FIFO will schedule to fix the robots in chronological order regardless of potentially long processing times leaving other robots waiting. dSSPT on the other hand, may delay some tasks in order to decrease the number of robots requiring human help simultaneously, but may increase makespan.

IV. Area Coverage Prospecting Problem

A simulation of a human-supervised multirobot team prospecting for minerals was developed. In it, the team has a mission of exploring an area to look for minerals. When a mineral patch is found, it becomes a task for the human supervisor to note and identify it.

The parameters of this problem include map size (the size of the area that has to be explored), density of mineral occurrence, randomness (or periodicity) of mineral distribution, the interaction time required at each mineral patch found, the velocity of each robot, and the number of robots. Since velocity is the only parameter in this simulation specific to robot performance, homogeneous robots will all have the same velocity and heterogeneous robot velocities will differ. Robot tasks are assumed to be preemptable and human tasks are assumed to be semi-preemptable. Figure 3a shows an example of what a

simulation with 7 robots looks like on a timeline plot showing both robot and human tasks.

Any minerals found along a robot's path will produce tasks for the human supervisor, which are then scheduled using a particular algorithm. The results below discuss the use of dSSPT versus FIFO for scheduling human tasks.

The robots on the team can operate in two different task allocation modes for their schedules: reallocatable, or non-reallocatable. For both, all robots are initially allocated an area based on their velocity such that in the absence of minerals, they will all finish at the same time. In the reallocatable mode, if one robot finishes earlier than other robots, the scheduler reallocates the part of an unexplored area from an unfinished robot to the finished one such that both robots will finish simultaneously if no future minerals are found. The non-reallocatable mode is one in which tasks cannot be shared. This can emulate situations where robots are located far from each other (like the Mars rovers).

In relating the parameters of this simulated problem to the HMRI metrics, NT can be expressed as:

$$NT = \frac{\textit{Time to cover area without minerals}}{\textit{Expected number of human mineral tasks}} = \frac{area / velocity}{area * density} \quad (3)$$

$$\therefore \boldsymbol{NT = \frac{1}{velocity * density}}$$

V. Saturation for Area Coverage Problem

Saturation occurs when the human cannot address any more robot tasks (the robot team is large enough that there is not enough idle time left to address more tasks). At saturation, the completion time of the team is limited mainly by the human's task completion time. Thus increasing team size would not lower the mission completion time, since the human cannot work any faster than one task at a time.

The expected minimum time to complete the mission for the prospecting problem is:

$$\textit{Min Completion Time} = \frac{\textit{Time to cover area}}{\textit{Number of Robots}} + \frac{\textit{Number of human tasks} * \textit{Time of each task}}{\textit{Number of Robots}} = \frac{area / velocity + area * density * IT}{\textit{Number of robots}} \quad (4)$$

The curve of this function can be seen as the dotted red line in Figure 3b. Note that this is the expected *minimum* completion time because the equation does not consider the occurrences of overlapping mineral tasks when one robot must wait for the human to address another robot first.

The minimum time for human saturation (or minimum human completion time) is equal to the amount of time it takes to address mineral tasks without consideration of human idle time:

$$\textit{Human Saturation Time} = area * density * IT \quad (5)$$

This threshold for human completion time is the dashed horizontal red line in Figure 3b. This equation is the *minimum* human completion time, in that it does not factor in any human idle time at all.

The point where these two curves intersect is the *minimum saturation point* where the mission completion time is equal to the human completion time. Equating Equations 4 and 5 and solving for the number of robots:

$$\frac{area / velocity + area * density * IT}{\textit{Number of robots}} = area * density * IT \quad (6)$$

$$\therefore \boldsymbol{Number\ of\ robots = \frac{1}{velocity * density * IT} + 1}$$

Substituting NT from Equation 3 we get:

$$\boldsymbol{Number\ of\ robots = \frac{NT}{IT} + 1}$$

This is exactly the same as the upper bound for fan-out in Equation 1. This demonstrates mathematically that fan-out is a minimum estimate that occurs only under the ideal conditions of no variance in NT (i.e., periodic tasks) or IT.

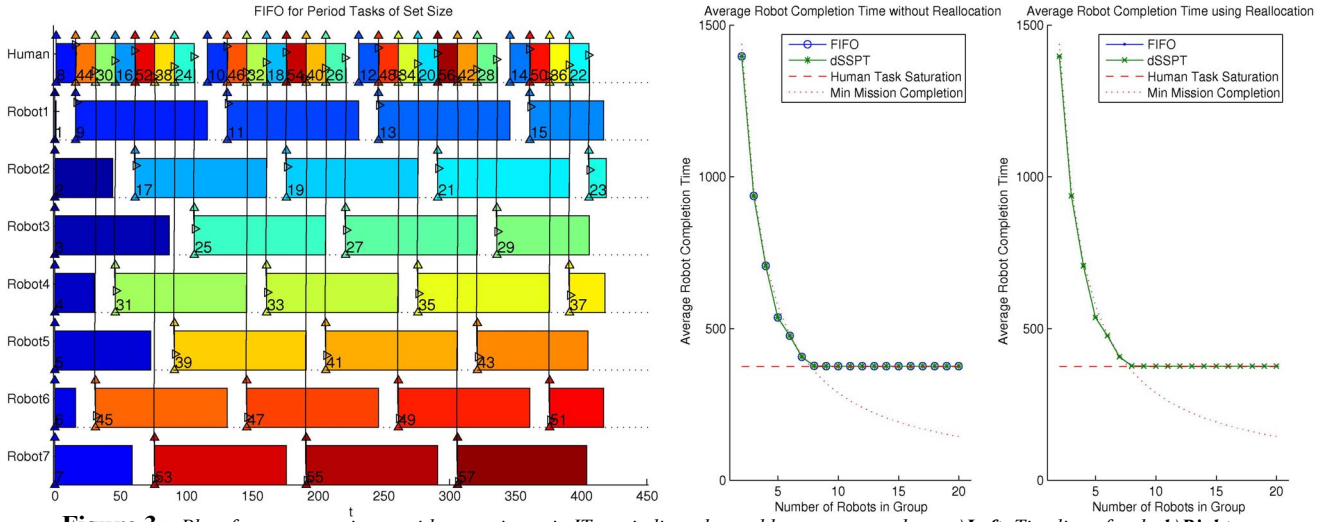


Figure 3: *Plots from an experiment with no variance in IT, periodic tasks, and homogenous robots.* ***a)Left****: Timeline of tasks* ***b)Right****: Average human completion time of trials compared to expected human saturation time (dashed) and expected mission completion time (dotted)*

VI. CHARACTERIZATION OF DSSPT

Simulations comparing the two scheduling algorithms for humans (dSSPT and FIFO) examine trends based on varying the parameters for density, density randomness (aperiodicity of tasks), velocity and IT parameters. In each simulation, these human scheduling algorithms are coupled with one of two possible robot scheduling modes, either with robot task reallocation, or without. Team saturation, downtime and area coverage performance are discussed in this section.

A. Baseline: No Variance, Periodic Tasks

As a baseline to demonstrate the situation of ideal saturation implied by FO (Equation 1), an experiment was run with periodic occurrence of minerals (no randomness), same velocity of 1m/s for all robots, a fixed IT of 15s with no variance and a density of 0.01 (NT = 100s) in a 50x50m^2 map. The results can be seen in Figure 3.

Saturation – The most interesting finding is that the saturation point exactly corresponds to the predicted FO. Based on Equation 1, the expected saturation point is 100/15 + 1 = $7^2/_3$ robots. Figure 3b shows that human saturation does indeed occur before a team size of 8 and remains saturated beyond that size. A team size of 7 resulted in a human completion time close to, but slightly higher than, the expected minimum completion time. Figure 3a shows that at this size, the human still has some idle time and thus is not completely saturated.

Same performance – Since these periodic tasks had the same IT, dSSPT ordered the human tasks in FIFO order. Also, since the tasks were evenly spaced throughout the area, all robots finished at similar times and no robot task reallocation occurred.

B. Non-zero variance, Aperiodic Tasks: Human Saturation

This section compares theoretical span-of-control (FO) to actual span-of-control for simulations. For a fairer comparison, only the simulations with robot reallocation are discussed because this mode dynamically redistributes remaining tasks amongst available robots to lower robot completion times. This means areas are reallocated between robots such that the minerals resulting in human tasks are generally found sooner, thereby reducing human idle time and bringing the human closer to saturation. However, the reallocation mode is an online heuristic algorithm and not guaranteed to minimize completion time. A better robot task reallocation algorithm can potentially decrease the saturation point, but the relative difference in saturation between FIFO and dSSPT should remain the same.

Section VI.A above showed that when tasks occur periodically and all with the same IT, the saturation point (which defines span-of-control) does indeed coincide with the theoretical fan-out. This section shows that when variances occur in IT or NT, the saturation point increasingly drifts away from the FO estimate.

1) Human completion time

When the tasks are periodic but the ITs vary (Table 1, rows 2-4), if the variance is small the saturation is essentially the same as theoretical FO. However, when IT variance is increased, the saturation point gets pushed to larger team sizes. In row 3 with mean IT of 5, 15 or 25s (the mean is randomly selected from those three values) and variance 3, both FIFO and dSSPT's saturation occurs at a team size of 10 instead of the predicted 8. As IT varies even more, as in row 4 with mean IT of 5, 25 or 45s, not only does the saturation occur after the FO estimate of 5, but dSSPT starts to have a slightly larger saturation compared to FIFO (8 compared to 7), as can be seen in Figure 4a.

The reason that saturation occurs later is the varying nature of the tasks (in terms of periodicity and size), as well as the finite timeline of the mission, both of which are expected in real applications. Figure 5b illustrates that when tasks occur aperiodically with the expected average NT, even with the same IT there are instances where the human has idle time, as well as periods where they are over-subscribed. It is in those periods of idle time where potentially more robot tasks can be allotted, thus increasing the span-of-control. Similarly, Figure 5c shows that when tasks occur periodically but have varying IT, there are again periods of idle time and over-subscription for the human.

	Parameters					*Span-of-Control*			*Downtime Ratio (dSSPT/FIFO)*				*Area Coverage*	
	Mineral Periodicity	Mineral Density	Robot Velocity	IT ~ $\mathcal{N}$(mean, var)	$\frac{\text{Mean NT}}{\text{Mean IT}}$+1	FO Upper Bound Estimate **[A]**	FIFO with reallocation **[B]**	dSSPT with reallocation **[C]**	Ratio @ size [A] without reallocation	Ratio @size [A] with reallocation	Ratio @ size [B] with reallocation	Ratio @ size [C] with reallocation	Ratio @ size [C] without reallocation	Ratio @ size [C] with reallocation
1	Periodic	0.01	1	(15, 0)	6.67+1	8	8	8	1	1	1	1	1	1
2	Periodic	0.01	1	(15, 6)	6.67+1	8	8	8	0.99	0.98	0.98	0.98	1.01	1.01
3	Periodic	0.01	1	([5 15 25], 3)	6.67+1	8	10	10	0.92	0.92	0.84	0.84	1.04	1.07
4	Periodic	0.01	1	([5 25 45], 3)	4+1	5	7	8	0.88	0.88	0.88	0.84	1.06	1.10
5	Random	0.01	1	(15, 0)	6.67+1	8	16	16	1	1	1	1	1	1
6	Random	0.005	1	(15, 0)	13.3+1	14	>20	>20	1	1	1	1	1	1
7	Random	0.02	1	(15, 0)	3.33+1	4	8	8	1	1	1	1	1	1
8	Random	0.01	5	([5 15 25], 3)	1.33+1	2	4	4	0.98	0.98	0.96	0.96	1.07	1.08
9	Random	0.01	[1 5 10]	([5 15 25], 3)	1.25+1	2	6	7	1	1	0.98	0.95	1.04	1.08

Table 1: *Summary of results based on varying parameters listed in the first 4 columns. Row 1 is the baseline, rows 2-4 vary IT, rows 5-7 vary mineral density and distribution, rows 8&9 compare homogenous and heterogeneous robot velocities.*

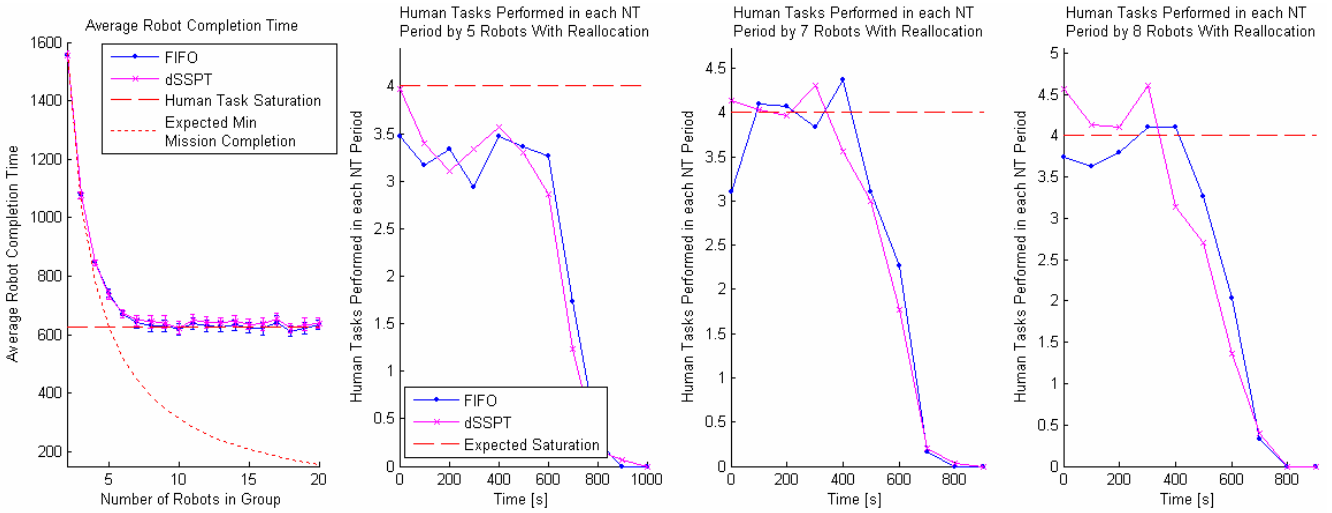
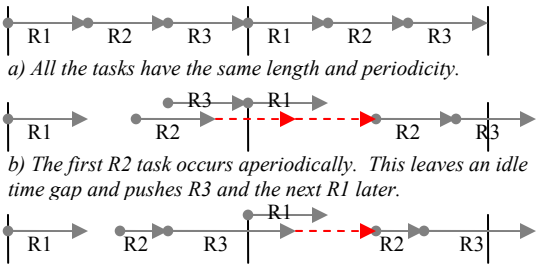


Figure 4: *Experiment with periodic tasks, homogenous velocity of 1m/s, mean IT of 5, 25 or 45.* ***a) Left****: Human completion time compared to expected human saturation and mission completion.* ***b-d) Right****: Number of tasks addressed in each NT period for different team sizes*

When dSSPT reorders non-preemptable tasks, if it reorders the first task in the list, idle time gets inserted into the schedule. This difference occasionally pushes dSSPT to have a slightly larger span-of-control than FIFO.

Therefore, in order for the human to be fully saturated, there must be no idle time at all (or at least not enough to fit an average task). More robots mean that more tasks are released earlier for the human, which reduces idle time.



a) All the tasks have the same length and periodicity.

b) The first R2 task occurs aperiodically. This leaves an idle time gap and pushes R3 and the next R1 later.

c) R1, R2 and R3 have different IT and occur periodically. There are idle times and over-subscribed times.

Figure 5: *Examples of how variance in NT and IT incur idle times and over-subscribed times. This allows more tasks to "fit" in.*

2) NT/IT Over Time

Since the estimated saturation, NT/IT, is an average, the actual number of tasks addressed in any NT period may differ. For example, when there are periodic tasks with mineral density of 0.01 and IT of 5, 25 or 45s with variance 3, the expected FO is 5. However, the previous section demonstrated that the actual saturation did not occur until 7 for FIFO and 8 for dSSPT, as shown in the human completion time graph of Figure 4a.

Viewing this example over intervals of time (Figure 4b-d) shows that indeed at a team size of 5, the expected number of tasks addressed in that time frame (NT/IT) is 4, but the actual number of tasks addressed is much lower at around 3.3 during "steady state" (excluding three points of the tail where the mission is winding up). However, at a team size of 7, we can see that the mean for FIFO is indeed close to 4 at 3.8, and at a team size of 8, dSSPT also has NT/IT at 3.9. This again underlines that the saturation size for a team operating in a dynamic environment is higher than the average-based estimate for FO.

Figure 4b-d show that the number of tasks addressed over time also varies depending on the algorithm. dSSPT schedules more tasks earlier on, so it has a shorter steady-state period compared to FIFO, with a higher NT/IT in the beginning and an earlier decline. This suggests that it might be more efficient to have dynamic team sizes depending on the algorithm used. For dSSPT, it seems that a dynamic team size with a larger team at the beginning and a smaller one as NT/IT decreases would be most efficient. Those robots that are unneeded for latter portions of the mission can potentially move on to start other missions sooner.

C. Downtime Comparison

In every experiment, dSSPT had equal or smaller total downtime compared to FIFO. Table 1 shows the ratios of dSSPT to FIFO downtime for reallocatable and non-reallocatable tasks at various team sizes.

When all human tasks have exactly the same processing time (IT) without variance (Table 1, rows 1, 5-7), dSSPT and FIFO had the same performance with respect to downtime. This makes sense, because if all processing times are constant, then the order will depend only on the release time, according to Equation 2.

If the tasks are periodic but the ITs vary (Table 1, rows 2-4), then dSSPT shows a marked improvement over FIFO, especially if there are many tasks queued and waiting for human assistance. Two types of variances were considered, Gaussian variance around one mean (Table 1, row 2) and having Gaussians around several means (Table 1, rows 3-4) to simulate the idea of variegated tasks with various lengths grouped as short, medium and long, similar to what would be encountered in a real mission. As the variance of IT increases, dSSPT performs increasingly better than FIFO, from a 2% improvement in row 2 to 16% in row 4 for robots with reallocation at the dSSPT saturation size (listed in the rightmost column of Table 1's downtime section).

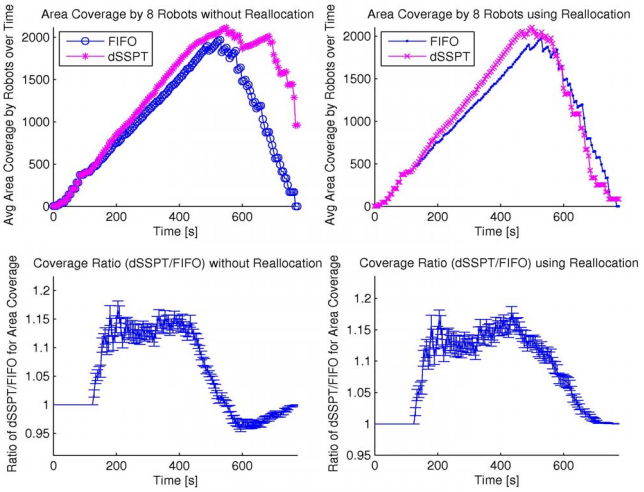


Figure 6: *Area coverage of dSSPT vs. FIFO at team size of 8 for periodic tasks, homogenous velocity 1m/s,mean IT of 5,25 or 45*

D. Area Coverage Over Time

dSSPT outperforms FIFO most of the time. Simulations with IT with no variance yielded no difference between the two algorithms, as expected. Variance in periodicity of minerals and in IT yielded more interesting results. Figure 6 shows area coverage over time for the same experiment discussed in the previous section.

For team sizes at the experimental saturation point, dSSPT always outperforms FIFO using scheduling with robot task reallocation. With reallocation, the minerals are found sooner since there are fewer idle robots and the supervisor is oversubscribed more often. This allows dSSPT to order the overlapping tasks more efficiently to reduce downtime. An example of this is seen in the two graphs on the right side of Figure 6, where FIFO and dSSPT with reallocation are compared. For dSSPT, a team size of 8 is its span-of-control and dSSPT always outperforms FIFO in terms of area coverage (the ratio between the performances is seen in the lower graph to be greater than 1).

Team sizes smaller than the saturation point tend to have larger variances in area coverage performance. When team sizes approach saturation, dSSPT again outperforms FIFO. For team sizes larger than the saturation point, as team size increases, dSSPT's peak performance over FIFO progressively shifts earlier. Later in the mission timelines, FIFO occasionally outperforms dSSPT, as can be seen in the two left graphs of Figure 6, which use no reallocation. This indicates that dSSPT is scheduling many of the smaller human tasks earlier on, thus freeing the robots to cover more area in the beginning, and leaving the longer human tasks until later, so the robots waiting on the human cover less area. This problem is less prominent in scheduling with reallocation because in that case, when the robot is busy waiting for the human, other free robots can take over its remaining coverage area. The oversubscription occurring when the team is larger than the saturation size show the drawback of SPT-based heuristics–long tasks must wait.

E. Summary of Results

- dSSPT outperforms FIFO in terms of lower downtime for aperiodic tasks (e.g., random mineral distribution) of varying IT. This results in a faster rate of area coverage for the simulated prospecting problem; in other words, more human tasks are completed sooner, thus freeing the robots to perform their tasks sooner.
- When there is any variance in either NT or IT, the actual span-of-control becomes larger than the average-based prediction of FO. This means that the upper bound of Equation 1 is actually a lower bound on the maximum team size for any practical situation.
- Looking at tasks performed in each NT period over time is a good gauge of whether a human is saturated. It is also a potentially good method for dynamic team sizing.

VII. Conclusions and Future Work

As supervisory control of multirobot teams becomes more commonly used, there is an increasing need for efficient utilization of the human supervisor's time to maximize team performance. In this paper we suggest prioritizing human tasks such that robots regain autonomy sooner. Due to the nature of human tasks as precedence constraints for robots, downtime minimization through the dSSPT algorithm results in higher team efficiency. Based on a simulated prospecting mission, dSSPT was able to prioritize tasks for humans such that the team generally covered more area in less time.

Analysis of these scheduling algorithms for span-of-control shows that actual span-of-control is greater than the average-based prediction given by the fan-out equation's upper bound. Additionally, analysis of NT/IT over time showed that the number of tasks performed in each NT period dropped off towards the latter part of the mission, with the drop-off occurring sooner for dSSPT than FIFO. This trend suggests that dynamic team sizing would make the most use of robot resources and also that span-of-control is affected by the scheduling algorithm used.

Future work includes human user trials to test the robustness of dSSPT to variance in predicted IT and dynamic team size experiments. Another possible direction is to develop a scheduling algorithm for more diverse tasks including preemptive, semi-preemptive and non-preemptive.

References

[1] J. W. Crandall, M. A. Goodrich, D. R. Olsen, and C. W. Nielsen. "Validating Human-Robot Interaction Schemes in Multi-Tasking Environments." Systems, Man, and Cybernetics Part-A: Systems and Humans. Volume 35, Issue 4, Pages 438-449. 2005.

[2] M. L. Cummings, P. M. Mitchell, "Management of Multiple Dynamic Human Supervisory Control Tasks for UAVs", HCI Intl. Human Systems Integration Conference, 2005.

[3] B. MacCarthy, J. Wilson, S. Crawford, "Human Performance in Industrial Scheduling: A Framework for Understanding", Human Factors and Ergonomics in Manufacturing, Vol.11(4) p286-307, 2001.

[4] S. Mau and J. Dolan, "Scheduling to Minimize Downtime in Human-Multirobot Supervisory Control," Workshop on Planning and Scheduling for Space, October, 2006.

[5] D. Olsen and M. Goodrich, "Metrics for evaluating human-robot interactions," NIST Performance Metrics for Intelligent Systems, 2003