Scheduling to Minimize Downtime in Human-Multirobot Supervisory Control

Sandra Mau and John Dolan

Robotics Institute, Carnegie Mellon University

Introduction

Currently, any robot teams operating in the real world are overseen by human operators. Many multirobot architectures take advantage of the human presence by deferring complicated tasks to the operator for assistance. This type of cooperation with a human in-the-loop is known as supervisory control. As robots become more autonomous and reliable, the number of robots a single human can oversee in a group will increase since they will require less attention. Hence it becomes important to efficiently schedule an operator's time when he has to deal with multiple tasks.

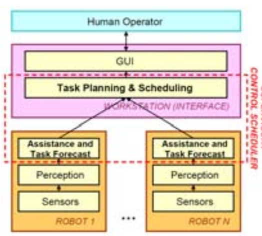
This research proposes the use of on-line scheduling of operator time in human supervisory control of multirobot teams to maximize the number of robots a human operator can manage and overall team productivity.

Motivation

Objective

Maximize robot productivity by fully utilizing human operator time.

- Requires decreasing robot *downtime* (amount of time spent inactive while waiting for human assistance)
- Simultaneously maximizes operator's *span-of-control* (number of robots a human can oversee in a team, such that no more than 1 robot waiting on average.)



Typical system architecture for human-multirobot supervisory control

Applications

Some potential space applications include:

- Future lunar & Mars base missions
- Multirobot science & surveying missions
- Managing time for mission specialists

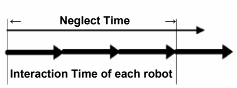
Background

Scheduling

The classic scheduling problem is to order a set of tasks such that certain objectives are achieved. SPT and SSPT are both greedy heuristics with a min downtime objective and FIFO orders things in the order they arrive.

- Limitation of SPT: does not consider asynchronicity
- Limitation of Shifted SPT: better than SPT when sparse, worse when dense
- Limitation of FIFO: effectively the same as no scheduling

Human Multirobot Interaction



A method of averages was proposed by Crandall et al. [2005] to determine span-of-control. It involves determining the ratio of an average *Neglect Time* (a robot's autonomous time before requiring operator help) and *Interaction Time* (the time a human spends addressing the robot task.) The major limitation of Crandall's averaging method is that it is based on static averages obtained through offline experimentation. Thus, variations due to environment, users, and task types are not considered.

Scheduling to Decrease Downtime: dSSPT Algorithm

dSSPT (double Shifted Shortest Processing Time)

Step 1) Append each new task to the dSSPT-scheduled list.

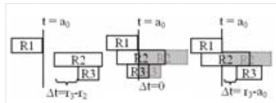
Step 2) Given a new task j, and existing task i in list (with $r_i \leq r_j$), do pair-wise swaps based on the following:

If $a_j + 2*\Delta t < a_i$ is true,

$$\text{where } \Delta t = \begin{cases} 0 & , \text{if } i>=2 \\ r_j - max(a_0, r_i) & , \text{otherwise} \end{cases}$$

then swap tasks i and j.

Step 3) Do Step 2 comparison down the list for tasks $i=j-1, j-2, \ldots, 2, 1$



For three tasks, arriving in the order R1, R2, R3, a pair-wise comparison of R2 and R3 is being made. The figures show 3 possible cases.

Comparing Algorithms

Experiment

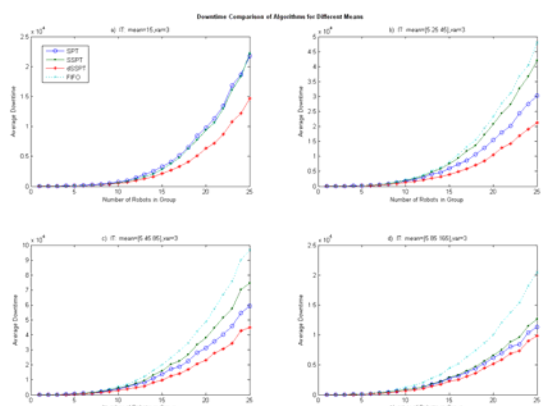
We compare dSSPT to FIFO as a baseline, as well as its precursors SPT and SSPT. The set of experiments shown below simulated tasks of three classes: short, medium and long tasks with a Gaussian variance in length.

Results

- Outperforms FIFO, SPT and SSPT in terms of decreasing downtime
- Improvement over FIFO ranging from 30% to over 50% (the more variation in task lengths the more improvement is shown)

Insights

- dSSPT is basically a hybrid of an improved SSPT and the original SPT in the different regimes where those algorithms perform best. For sparse schedules, it resembles SSPT and for dense schedules SPT.
- Drawback of dSSPT for dense schedules are reminiscent of those for SPT, in that longer tasks may have to wait if there are many tasks queued. However, in the domain of supervisory control dense schedules should be infrequent.



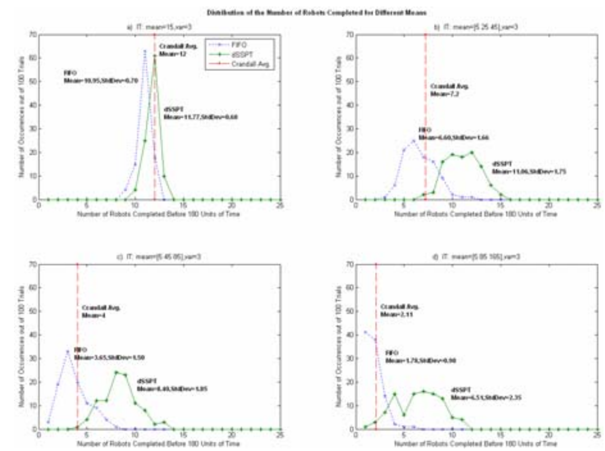
Effect of Scheduling on Team Size

Experiment

We compared dSSPT to FIFO and a static averaging method to determine span-of-control for a certain neglect time with the three classes of tasks.

Results

- FIFO's mean is similar to the average performance method.
- The results of scheduling would yield exactly Crandall's result for the basic case where all robots have the same IT.
- For tasks of increasingly varied lengths (more reflective of reality), dSSPT outperforms both FIFO and the averaging method.



Conclusions

1) A scheduling algorithm, dSSPT, was developed based on a simple, greedy heuristic that tries to minimize downtime for a team. It outperforms the standard FIFO, SPT and SSPT algorithms and manages to do so in polynomial time.

2) When scheduling algorithms are applied in supervisory control, a downtime-efficient, on-line scheduling algorithm can increase the span-of-control in comparison to that predicted using a static performance averaging method or no explicit scheduling at all (i.e. FIFO).

3) Results suggest an online averaging method of determining span-of-control dynamically.

4) dSSPT may also be used in other applications that want to minimize downtime (or also waiting time and processing time).

5) Future work would look at a wider variety of tasks, account for deadlines and learn realistic models for human-robot interaction

This work is partially supported by NASA under Cooperative Agreement No. NNA05CP96A