Cooperative Robot Scheduling and Path Planning for D&D Applications

Sebastián A. Zanlongo – DOE Fellow
Alexander C. Wilson – Towson University

FIU Applied Research Center

Abstract

Current robot job allocation and scheduling is often performed in an ad-hoc manner. This is complicated by requiring robots to perform semi-autonomously in partially unknown environments. We present an approach for scheduling and path planning of robots carrying samples or executing some other task in human-denied environments. In particular, we demonstrate a methodology for:

- Allocating jobs to a heterogeneous set of robots
- Designing a safe trajectory for the robots
- Scheduling the jobs and trajectories to minimize runtime

Background and Motivation

Traditional approaches aim to transport materials from one location to another. Many existing approaches assume a completely known environment, or make simplistic assumptions about travel time. Our work allows for navigation in partially-known environments, allows for situations where robots break down, and can dynamically reschedule and design trajectories when new locations are added.

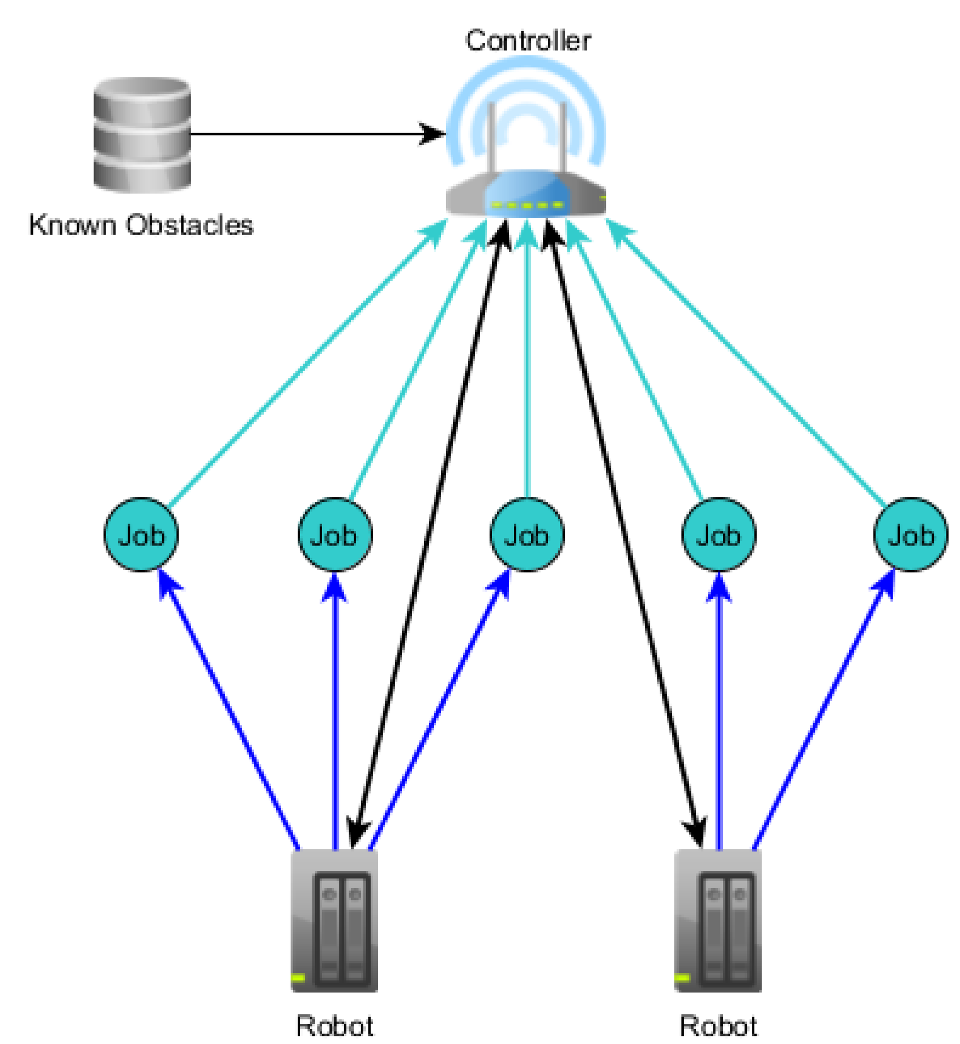
System Architecture

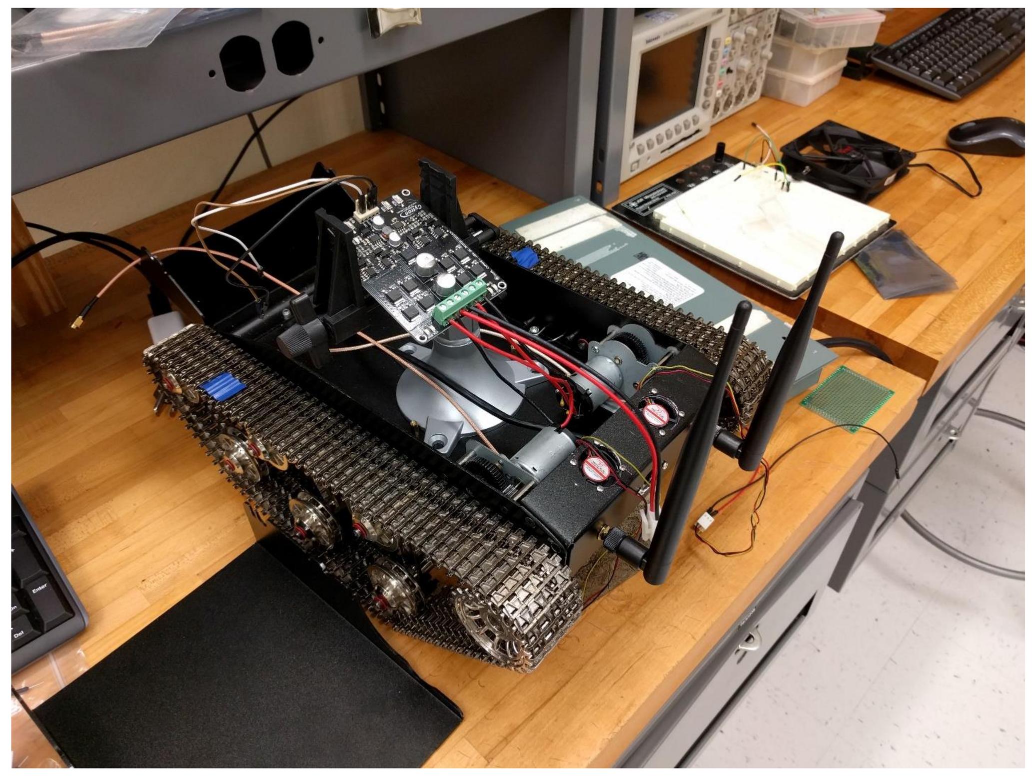
A central controller takes in a map of the environment and a set of jobs that must be executed. Jobs are restricted to robots that are capable of handling them. The scheduler then allocates jobs and designs paths so as to minimize the total run-time.

Controller
- Known Obstacles

Robots
- Specializations
- Locations
- Visit order

Jobs
- Specializations
- Locations
- Run-time
- Deadlines

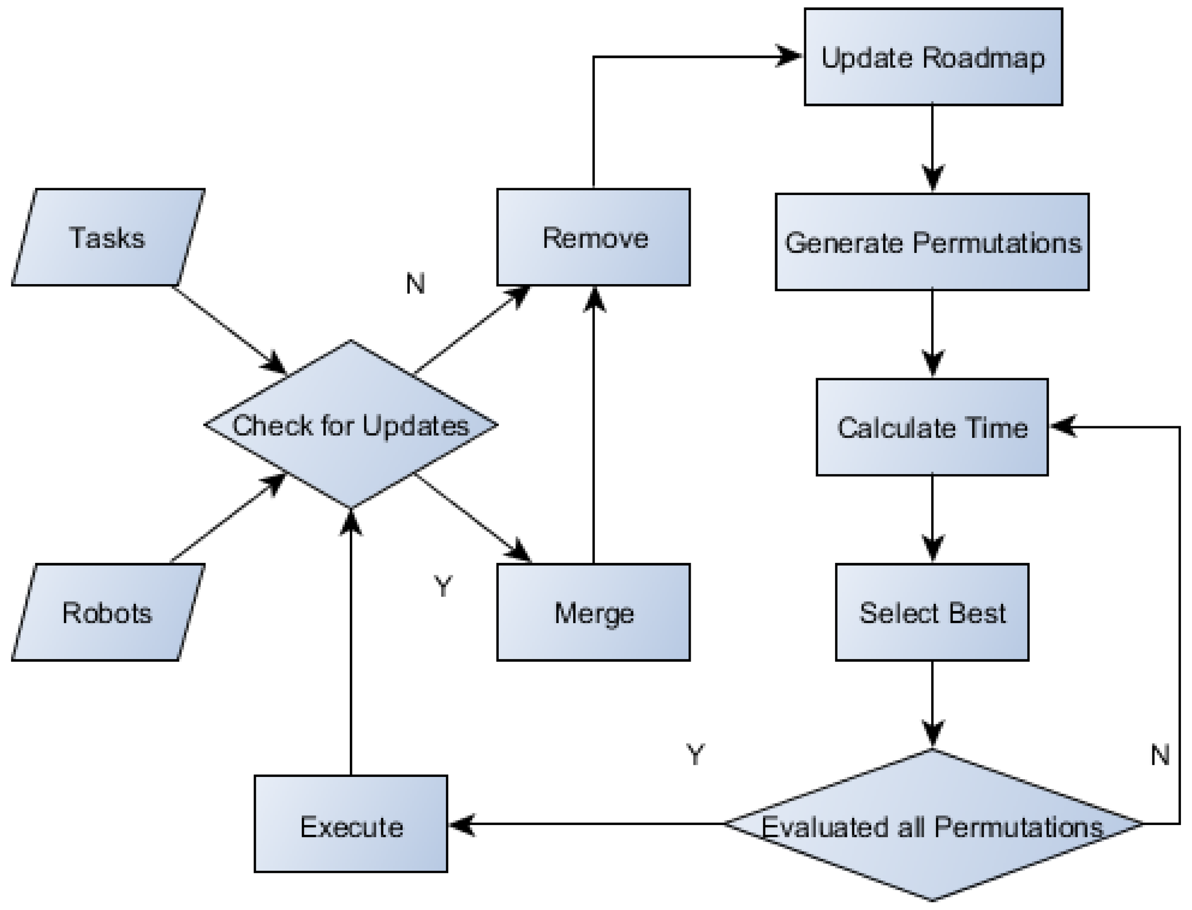




Scheduling

Scheduling is performed as follows:

1. For each robot, we generate multiple new visit orders with the new job in different schedules.
2. Each visit order is then translated into a path, and the total run-time is estimated.
3. Select the job allocation and visit order which results in the least amount of new time and missed deadlines.



Path Planning

Path planning is used to allow robots to navigate in partially unknown environments. This must take into account both obstacle-robot collisions and robot-robot collisions:

$$X^{j}_{obs} = \{x \in X | A^{j}(x^{j}) \cap O \neq \emptyset\}$$

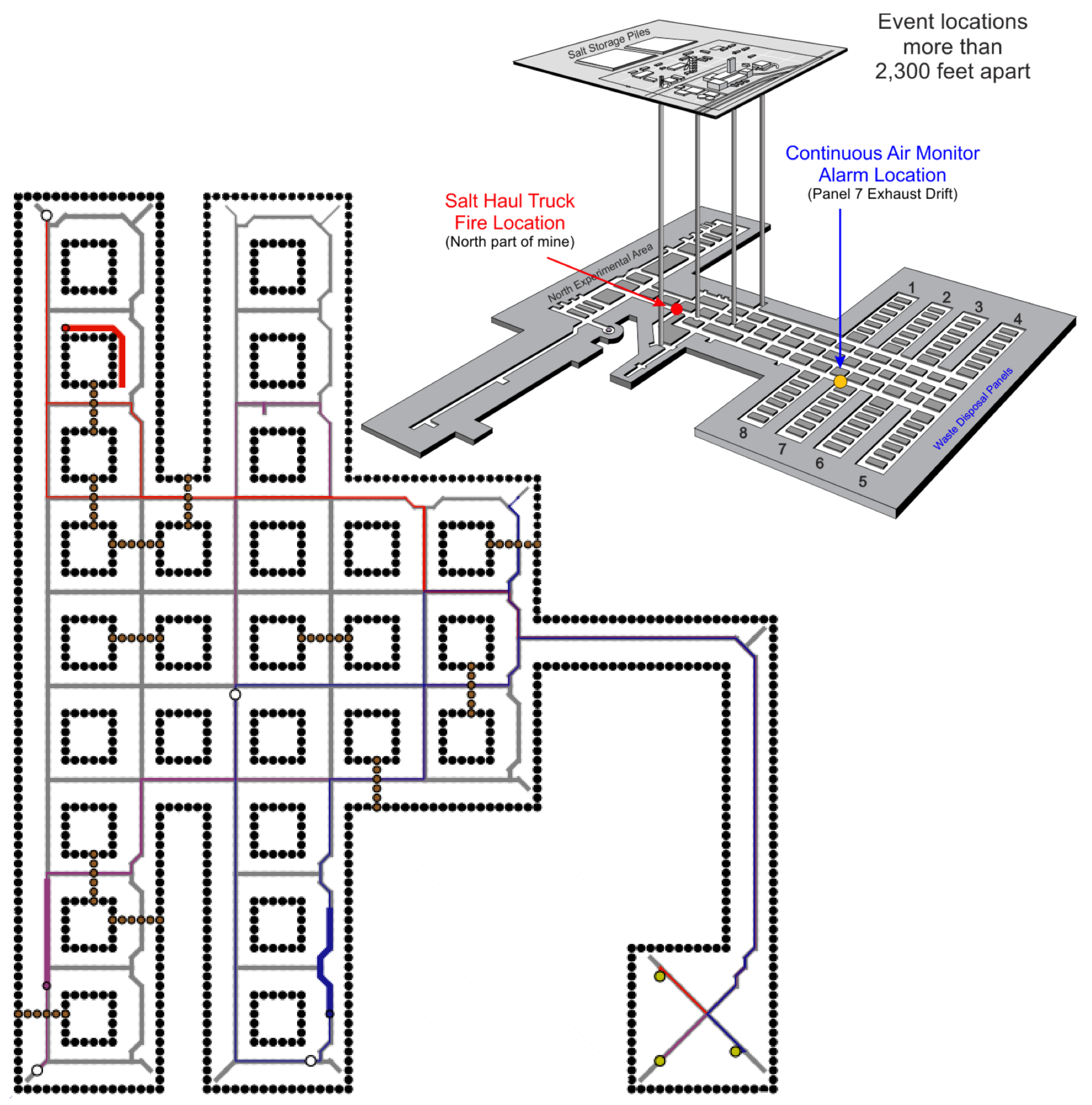
$$X^{jl}_{obs} = \{x \in X | A^{j}(x^{j}) \cap A^{l}(x^{l}) \neq \emptyset\}$$

$$X_{obs} = \left(\bigcup_{j=1}^{m} X^{j}_{obs}\right) \cup \left(\bigcup_{jl, j \neq l} X^{jl}_{obs}\right)$$

1. Based on known obstacles, a Voronoi decomposition is used to create a “roadmap” of safe paths
2. This roadmap is used in the scheduling phase to efficiently estimate travel times.
3. Tentative paths are found through the roadmap between consecutive jobs.
4. If an unknown obstacle is encountered, the robot re-plans around the obstacle by splicing in a new path segment.

Results and Conclusion

- Introduction of path planning capabilities for the robots.
- Works in dynamic, unknown environment.
- The roadmap allows for more reliable estimation of travel times.
- We also allow for jobs to require specializations.



Future Work

- Allow robots to update Controller with new obstacles
- Incorporate payload constraints
- Derive action trajectory $u: T \rightarrow U$
- Carry out physical experiments

Acknowledgements

- Dr. Leonardo Bobadilla – FIU SCIS
- Dr. Tamim Sookoor – G2, Inc.
- Anthony Abrahao – FIU ARC
- DOE-FIU Science & Technology Workforce Development Program

Sebastián A. Zanlongo, DOE Fellow, Computer Science | Applied Research Center | Florida International University | 10555 West Flagler Street, Miami, Florida 33174 | szanlong@fiu.edu