Telescope Management for Satellite Tracking: A Decentralized Approach

(Extended Abstract)

Feyza Merve Hafızoğlu
University of Tulsa, Tulsa, OK, USA
feyza-hafizoglu@utulsa.edu

Roger Mailler
University of Tulsa, Tulsa, OK, USA
mailler@utulsa.edu

ABSTRACT

The rate that the space debris is increasing has now outpaced our ability to build additional sites. Another solution is to use large numbers of low-cost optical sensors, which can be easily deployed at a fraction of the cost of a traditional tracking station. In this paper, we describe the satellite tracking problem, a complex coordination problem that is subject to time and position constraints that must be solved in a communication limited environment. We adopt scheduling by a central facility as an initial solution and compare central and distributed repair mechanisms in terms of tracks completed.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms, Experimentation

Keywords

distributed constraint satisfaction, satellite tracking

1. INTRODUCTION

Currently, it is estimated that over 100 million pieces of space junk exist in Low-Earth Orbit (LEO) with over 21,000 objects being larger than 10 cm. To protect important assets, such as the Hubble telescope, the International Space Station (ISS), and many of our communication satellites, the U.S. Space Surveillance Network (SSN) monitors the trajectories of large orbiting objects using 29 sites located around the world. Due to increasing number of countries and private organizations that gain access to space, the job of tracking the artifacts they deposit has become more difficult. A more cost-effective solution than establishing new sites is using large numbers of low-cost optical telescopes for improving both the number and the fidelity of tracks.

Many of LEO satellites are bright enough to be seen by the naked eye as they streak across the night sky. For a telescope to observe a satellite, two constraints should be satisfied: (1) telescope should be in twilight time (the satellite must be illuminated by the sun, yet the background be dark) and (2) the satellite must be within the observation range of the telescope. We address all factors that prevent tracking, such as weather and hardware failures, as telescope failures. We assume that one day is divided into fixed length slots, e.g., two minutes. The daily schedule of a telescope consists of tasks that are arranged one task per slot.

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.

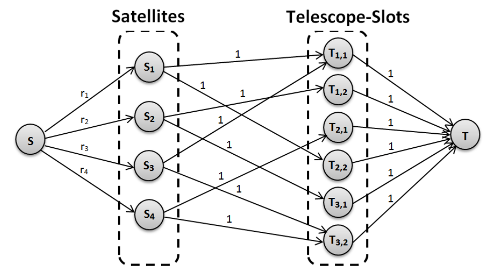

Figure 1: Max-flow in bipartite graph with telescope-slots

The scheduling problem can be formalized as a resource allocation problem to assign limited number of resources (telescope-slots) to a set of tasks (tracking satellites) for maximizing the utility (completed tracks). Given a set m telescopes $\{t_1, \ldots, t_m\}$, a set n satellites $\{s_1, \ldots, s_n\}$, and telescope-slots, x^t_{ij} (x^t_{ij} has a value of 1 if t_i can track s_j at slot t, and 0 otherwise), the utility of s_j, $U_j \longmapsto \mathbb{R}$, is

$$U_j(x^1_{1j}, \ldots, x^k_{mj}) = r_j \left[\sum_{t=1}^{k} \sum_{i=1}^{m} x^t_{ij}/r_j \right],$$

where k is the number of slots in a day and r_j is the required number of tracks. The goal is to maximize the total utility, i.e., $max \sum_{j=1}^{n} U_j(x^1_{1j}, \ldots, x^k_{mj})$, given two constraints:

$$(1)\ \forall_{i\in m, j\in n, t} x^t_{ij} \in \{0,1\} \text{ and } (2)\ \forall_{i\in m,t} \sum_{j=1}^{n} x^t_{ij} \leq 1.$$

This problem can be solved by using a max-flow algorithm on the bipartite graph formed by having the satellites on one side and the telescope-slots on the other connected by edges with capacity 1 (see Figure 1). An edge is added between a satellite and a telescope-slot node, only if the satellite can be tracked by the telescope in this slot, i.e., $x^t_{ij} = 1$.

2. EVALUATION

Deriving an optimal schedule using distributed techniques can be a time consuming and cost prohibitive. Therefore, we assume that a central facility computes the initial schedules at the beginning of each day as shown in Figure 1. Both the central facility and the telescopes can only directly communicate with the entities within their range. Otherwise, a

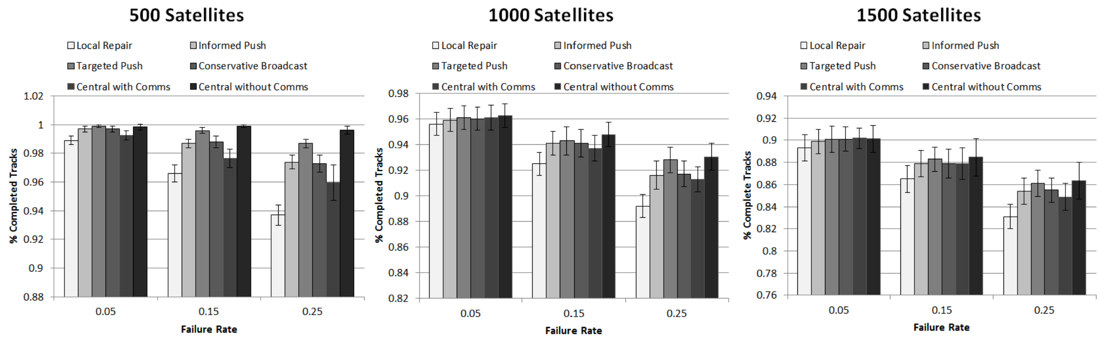

Figure 2: Percentage of completed tracks by using 200 telescopes

multi-hop communication network model is adopted. Once the allocation is made, the execution should be monitored to recover from failures. Monitoring and adjustment need explicit communication to gather information and redistribution of schedules in case of failures.

Test Environment: We conducted experiments with 200 telescopes in a 360x180 cylindrical area representing the Earth. The telescopes were randomly placed. We then add a fixed number of satellites (500, 1000, or 1500) with random initial trajectories and then following an orbital path. Each satellite is required to be observed three times a day. The twilight period is set to 48 minutes. A track has a failure rate (5%, 15%, or 25%). Each test simulates a 24 hour period and the average of 100 runs are reported. We tested four different solution repair techniques in our study.

Centralized Repair Mechanism: Telescopes report failures to a centralized facility, which reconstructs the bipartite graph by adapting to the current state. The new schedules for the rest of the day are recalculated and sent back to the telescopes. Communication delays will occur based on the distance between the central facility and the telescope, and the speed of the communication medium. Two issues are (1) recomputing a new solution can cause considerable solution instability and (2) in environments where communication is intermittent, information cannot be communicated fast enough to keep up with the changes.

Informed Push: This method is inspired by distributed max-flow algorithms based on push relabeling [1, 2]. In case of a failure, the telescope generates a push message by attaching the missed task, a "candidate list" of telescopes that can complete the task (sent with the original tasking by the central facility), and "message path", which is a list of telescopes that have seen the message to prevent cycles. A telescope, who receives a push message, adds the task to its schedule if it's possible. Otherwise, the push message is sent to a neighbor (preferably a neighbor that is in the candidate list) after appending its name to the message path. The process continues until either the track finds a new home, or it reaches a dead end and dies.

Targeted Push: Along with the initial schedule, the central facility sends information about the set of telescopes that can track each satellite and the communications network structure. When a failure occurs a directed message is sent to a telescope that might be able to perform the task.

Conservative Broadcast: The failed telescope broadcasts a message to its neighbors, and the neighbors broadcast the message recursively out to distance three. A receiver of the message, who can perform the task, sends a message back to the initiator telescope but does not reserve the slot yet. The initiator telescope can receive several availability messages and reserves the first availability message.

Results: Figure 2 shows the averages and standard deviations of the percentage of completed tracks. Central repair without communication delay (a fully connected network) mechanism is adopted as the optimal solution. The distributed mechanisms work fairly well in repairing failed tracks when compared to both the "optimal" solution and the central with delay method. Local repair mechanism, where the telescopes reschedule the task within their own schedules if possible, has severe drops in performance with increasing failure rate and/or number of satellites, while the performance of optimal solution slightly decreases. The targeted push mechanism outperforms the other non-optimal mechanisms in general. In fact, when the problem is not tightly constrained by the ratio of telescopes to satellites, e.g. in case of 500 satellites, the informed push and conservative broadcast mechanisms outperform the centralized with delay method. In case of 1000 satellites, the centralized with delay method barely catches the performance of these mechanisms for failure rate 0.05. The results indicate that the distributed repair mechanisms remarkably outperform central repair mechanism because the communication delay prevents central approach from being able to react in a timely enough manner.

3. ACKNOWLEDGMENTS

This material is based on research sponsored by the Air Force Research Laboratory, under agreement number FA8750-11-2-0066. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government.

4. REFERENCES

[1] J. Marberg and E. Gafni. An o($n^2m^{1/2}$) distributed max-flow algorithm. Technical report, University of California, Los Angeles, 1987.

[2] T. Pham, I. Lavallee, M. Bui, and S. H. Do. A distributed algorithm for maximum flow problem. In *4th International Symposium on Parallel and Distributed Computing*, pages 131–138, July 2005.