

Multi-Robot Assembly Scheduling for the Lunar Crater Radio Telescope on the Far-Side of the Moon

Preston Culbertson
 Department of Mechanical Engineering
 Stanford University
 Stanford, CA, 94305
 pculbertson@stanford.edu

Ashish Goel
 Jet Propulsion Laboratory
 California Institute of Technology
 4800 Oak Grove Drive, Pasadena
 California 91109, USA
 ashish.goel@jpl.nasa.gov

Saptarshi Bandyopadhyay
 Jet Propulsion Laboratory
 California Institute of Technology
 4800 Oak Grove Drive, Pasadena
 California 91109, USA
 saptarshi.bandyopadhyay@jpl.nasa.gov

Patrick McGarey
 Jet Propulsion Laboratory
 California Institute of Technology
 4800 Oak Grove Drive, Pasadena
 California 91109, USA
 patrick.mcgaray@jpl.nasa.gov

Mac Schwager
 Department of Aeronautics and Astronautics
 Stanford University
 Stanford, CA, 94305
 schwager@stanford.edu

Abstract—The Lunar Crater Radio Telescope (LCRT) is a proposed ultra-long-wavelength radio telescope to be constructed on the far side of the moon. The proposed telescope will be constructed by deploying a 1km wire mesh in a 3-5km crater using a team of wall-climbing DuAxel robots. In this work, we consider the problem of generating minimum-time assembly sequences for LCRT, using realistic models of travel speed and lighting. We pose the assembly sequencing problem as a mixed-integer linear program (MILP), which we solve to global optimality using commercial solvers. We present methods for modeling time-varying travel and assembly times, based on variable lighting conditions (including crater shadowing), and show how such time-varying parameters can be incorporated into the MILP. Finally, we present numerical studies of our method, showing how makespan varies with the number of assembly robots.

TABLE OF CONTENTS

1. INTRODUCTION.....	1
2. PRIOR WORK.....	2
3. PROBLEM FORMULATION.....	3
4. MATHEMATICAL MODEL	4
5. PARAMETER GENERATION.....	5
6. RESULTS	6
7. CONCLUSIONS AND FUTURE WORK	7
ACKNOWLEDGMENTS	7
REFERENCES	7
BIOGRAPHY	8

1. INTRODUCTION

The Lunar Crater Radio Telescope (LCRT) [1] is a proposed ultra-long wavelength radio telescope to be constructed on the far side of the moon. This telescope would offer significant advantages over Earth-based telescopes, including the ability to observe the universe at wavelengths greater than 10 meters, and shielding from radio interference (from both Earth-based

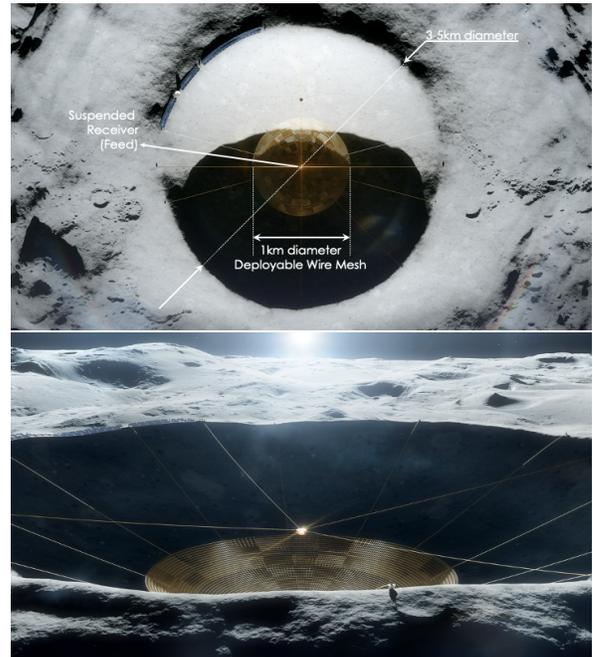


Figure 1: Concept art of LCRT [1] - Top View and Side View. Artist Vladimir Vustyansky.

and solar sources) by the Moon itself. The LCRT, if constructed, would have a 1 kilometer diameter, and would in fact be the largest filled-aperture telescope in the Solar System. See Figure 1 for a conceptual drawing of the telescope.

Such a telescope would allow scientists to observe the early universe in the 10-50m wavelength band, which to date has been inaccessible. These wavelengths are crucial to understanding the cosmological “Dark Ages,” and are not observable from Earth due to its ionosphere. Thus, LCRT could enable scientific discoveries by providing observations of the neutral intergalactic medium at wavelengths inaccessible with current technology.

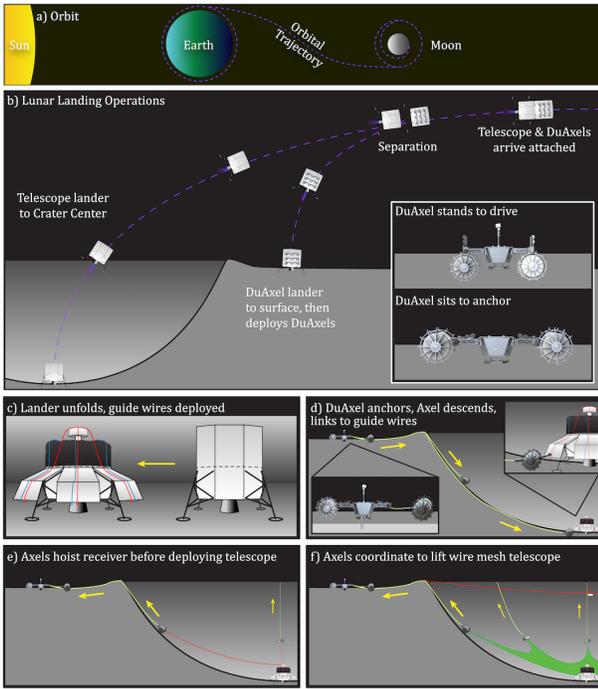


Figure 2: Schematic [1] of the Concept of Operations (ConOps) for LCRT assembly. The LCRT payload will land in two parts; the first, with telescope materials will be located in the crater center. The second, containing the assembly rovers, will land on the crater rim and serve as a depot. The rovers will proceed to retrieve structural cables from the central lander, and anchor them to the crater rim. Once all cables are anchored, the mesh will be deployed along the cables. This paper studies the problem of assembly planning and scheduling for this cable deployment process.

The proposed telescope consists of a 1km-diameter parabolic reflector, which is constructed using a variable-thickness wire mesh, suspended across a 3km-diameter crater. The mesh will be designed such that the reflector maintains its shape passively while suspended, at a variety of temperatures. Figure 2 provides a diagram of the LCRT Concept of Operations (ConOps). To construct the telescope, a team of autonomous DuAxel rovers [2], [3] will be launched with the telescope materials, and work to deploy the lift cables from a central lander. Each rover consists of two tethered robots, which connect to a common base. To fetch each cable, the robots will detach, descend into the crater, attach to the cable, and then ascend to the crater rim using the tether.

However, an important question remains: which cables should be assigned to each rover? Since lunar regolith is abrasive and can easily damage sensitive equipment (such as sensors and actuators essential to basic rover functions), in order to minimize mission risk, we want to find assembly sequences which minimize the total makespan of the telescope. This is difficult for two reasons. First, assembly sequencing is a combinatorial optimization, which is computationally difficult and scales poorly in the number of agents. Second, the parameters of our assembly task (i.e., rover speeds) depend on the lighting conditions of the crater, meaning the time required to tow and anchor a cable depends on whether the job is performed during lunar day or night, or if portions of the route are in shadow.

In this paper, we pose the assembly sequencing problem as a mixed-integer linear program, which can be solved effec-

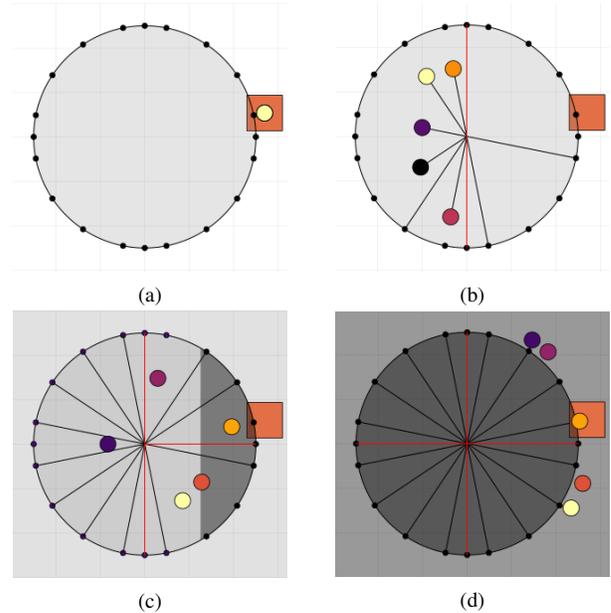


Figure 3: A team of five rovers (shown as dots) assemble the Lunar Crater Radio Telescope (LCRT). Using our proposed scheduler, we generate a minimum-time assembly sequence which assign lift (black) and receiver (red) cables to each robot to be retrieved from a central lander and anchored on the crater rim. Our scheduler reasons about lighting conditions, such as the partial shadowing seen in (c), which greatly affect the rovers’ travel speeds.

tively using commercial solvers. To capture the illumination effects mentioned previously we model the problem parameters as piecewise-linear functions of time, and demonstrate how these parameters can be incorporated into the planning problem. We also detail how the parameter values themselves can be generated from real data for a specific LCRT construction scenario. See Figure 3 for a pictorial illustration of this problem statement.

2. PRIOR WORK

Scheduling Problems

Scheduling problems are an important class of combinatorial optimization problems which lie at the heart of operations research, among other fields. While scheduling problems are deeply important, they are also fundamentally difficult, since such problems are *NP*-hard, i.e., at least as hard as the hardest problems in *NP*. Perhaps the canonical example of such problems is the job shop problem [4], which seeks to optimally schedule jobs in a machine shop to maximize throughput.

The problem considered in this work is more closely related to the Vehicle Routing Problem (VRP), a multi-vehicle generalization of the well-known Traveling Salesman Problem (TSP) [5], [6]. The TSP is to find a single minimum-cost tour (i.e. connected, cyclic path between all vertices) of an undirected weighted graph; it remains a canonical example of a combinatorial optimization. The VRP modifies the TSP to seek a set of N tours (for N different agents) which together cover the graph. This variant has also been of deep importance in the operations research community (see [7], [8] for a comprehensive survey).

The problem considered in this work is quite similar to the time-dependent variation of the VRP, wherein the distances between nodes depend on the time they are being traversed. While numerous papers [9], [10], [11] have considered this variant, most propose heuristic or meta-heuristic solutions to the problem, whereas we pose it as an integer program which can be solved exactly. Further, while most time-varying VRPs model the travel times as step functions, or introduce discrete variables to select “windows” for each job to begin/end, we instead model the time-varying parameters as piecewise-linear functions which can be modeled easily in modern solvers. While the job times considered in this work are deterministic, a number of authors [12], [13], [14] have considered the stochastic VRP, which remains an interesting direction for future work.

This work is also closely related to robotic assembly planning (see [15] for a survey), another class of combinatorial problems which focuses on sequencing physically-feasible assembly operations for a set of components. [16] presents a multi-robot variation of this problem, and shows it can also be posed as a mixed-integer program. [17] presents a method for multi-robot assembly which combines a high-level A* search with a low-level conflict-based search [18] to generate optimal assembly sequences. Our work is also related to multi-robot path finding [19], [20], although in this work we do not focus on collisions between agents (since the workspace is much larger than the rovers, allowing trivial collision avoidance), and instead concentrate on minimizing overall makespan.

Mixed-Integer Linear Programming

A fundamental tool leveraged in this work is mixed-integer linear programming, a variant of linear programming which requires a subset of the decision variables to take on integer values. A mixed-integer linear program (MILP) is given by

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{z}} \quad & \mathbf{c}^T \mathbf{x} + \mathbf{d}^T \mathbf{z} \\ \text{s.t.} \quad & \mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{z} \leq 0, \\ & \mathbf{x} \in \mathbb{R}^q, \mathbf{z} \in \mathbb{Z}^p \end{aligned}$$

where \mathbf{x} are the continuous decision variables and \mathbf{z} are the integer decision variables. This class of problems is quite powerful for modeling decision problems with discrete or logical structure, and have been long-studied in the optimization community. They are commonly solved using branch-and-bound or branch-and-cut techniques (see [5] for an overview), as implemented in a number of open-source and commercial solvers such as Gurobi [21].

3. PROBLEM FORMULATION

In this paper, we consider the problem of minimum-time multi-robot assembly sequencing for LCRT. Figure 4 shows a schematic of the proposed assembly site. A team of n rovers will be used to retrieve, tow, and anchor cables from the central lander. There are a total of $m = m_\ell + m_r$ wires, where m_ℓ is the number of lift wires, and m_r is the number of receiver wires. Each wire is to be anchored at a fixed radius R from the crater center, with desired angles $\theta_1, \dots, \theta_m$.

To model the realistic travel times of each robot, we model their speeds as both lighting- and location-dependent. When on the crater rim, each rover can move with a maximum speed of v_r^ℓ in illuminated sections (i.e., during lunar day) and v_r^d in darkness. Intuitively, we will have $v_r^d < v_r^\ell$, since poor

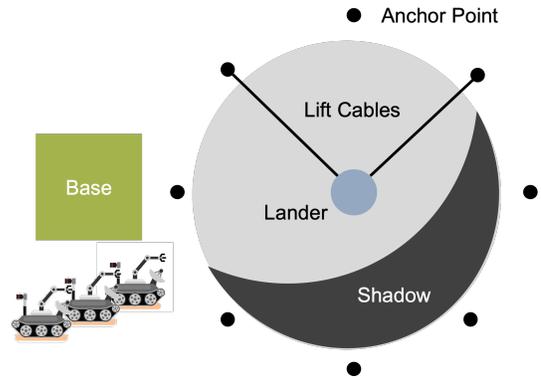


Figure 4: A schematic of the assembly planning problem studied in this paper. A team of rovers are deployed from a base on the crater rim, and fetch lift cables from a lander at the center of the crater. Each rover must tow the cable along the crater wall, and anchor it onto the rim. Rover travel speeds vary based on lighting conditions, including shadowing within the crater.

lighting conditions will require more conservative movement. The rovers also have speeds (v_d^ℓ, v_d^d) while descending along the crater wall, and (v_a^ℓ, v_a^d) when ascending while towing the cable. We note here that some portions of the crater can be shadowed during the lunar day, causing movement speeds to be non-uniform within the crater at a given time.

To fully specify the problem, we must also provide data on the crater’s dimensions and location, as well as the start time of the planned assembly mission. To this end, we assume the planner has access to a CAD model of the crater (in this work we use a model generated from lunar survey data from the Lunar Reconnaissance Orbiter) which can be used to compute path lengths between anchor points and the crater center. The CAD model is also required to compute lighting conditions and shadowing over the course of assembly.

Using this data, our problem is to generate an assembly plan, which is a map \mathcal{A} from wire indices $\{1, \dots, m\}$ to robots $\{1, \dots, n\}$, and a pair of start times s_i and end times f_i for each job. We require that each job duration $f_i - s_i$ be at least the time required to retrieve and anchor the cable, under the time- and location-dependent rover speeds discussed previously. Jobs also must not overlap, meaning a rover must complete a job and travel to the next anchor point before starting another. We seek to minimize the makespan of the telescope, which we define to be the time required for all jobs to be completed, and all rovers to return to the base (so they may be stored and protected from lunar conditions). Thus our problem is given by

$$\begin{aligned} \min_{s_i, f_i, \mathcal{A}} \quad & \text{LCRT makespan} \\ \text{s.t.} \quad & \text{job durations,} \\ & \text{travel times} \end{aligned} \tag{1}$$

The rest of the paper proceeds as follows. Section 4 presents a mathematical model for our optimization (1), which captures the time-varying travel and job times at the core of our problem. Section 5 details how we generate the problem parameters for the optimization using the physical data provided. Section 6 studies how our method performs on a specific LCRT construction scenario, including how the makespan varies with the number of robots n . We conclude in Section 7 with a brief discussion of the method and some next directions for future work.

4. MATHEMATICAL MODEL

Here we present the mathematical form of our assembly sequence optimization (1). We model the problem as a mixed-integer linear program (MILP), a common form for scheduling problems. To do so, we introduce two sets of variables, $\mathbf{z} \in \{0, 1\}^{n \times m \times m}$, which are binary variables with $z_{j,k}^i = 1$ if job j precedes job k for worker i , and $\mathbf{t} \in \mathbb{R}^{n \times m}$ where t_j^i is the start time of job j on worker i .

Time-Invariant Form

To begin, we formulate our problem in the case of time-invariant parameters before generalizing to the full problem. For the time-invariant problem, we consider a set of m jobs which have duration d_j . Each job will be assigned to exactly one robot, with a fixed travel time $\delta_{j,k}$ required to move from job j to job k . We also include $\delta_{0,j}$ and $\delta_{j,0}$ for $j = 1, \dots, m$, which are the travel times from and to the depot, respectively, for each job.

The time-invariant optimization is given by:

$$\begin{aligned} \min \quad & C \\ \text{s.t.} \quad & C \geq t_j^i + d_j + \delta_{j,0} && \forall i, j \quad (2) \\ & z_{j,k}^i \Rightarrow t_k^i \geq t_j^i + d_j + \delta_{j,k} && \forall i, j, k \quad (3) \\ & \neg z_{j,k}^i \Rightarrow t_j^i \geq t_k^i + d_k + \delta_{k,j} && \forall i, j, k \quad (4) \\ & \sum_{i,k} z_{j,k}^i = 1 && \forall j \quad (5) \\ & t_j^i \geq \delta_{0,j} && \forall j \quad (6) \end{aligned}$$

We now discuss the constraints in more detail. Equation (2) requires the objective C to be lower-bounded by the termination time of all jobs. Specifically, the objective must be greater than the start time of all jobs, plus their duration, plus the time required to return to the depot. The constraints (3), (4) define the ordering of the jobs, where we use \Rightarrow to denote an implication constraint (i.e., the constraint is imposed when the logical condition evaluates to true). If $z_{j,k}^i = 1$, then the start of job k on worker i must be greater than the start time of job j , plus the duration of job j and travel time from j to k . Equation (4) imposes the logical inverse. The constraint (5) enforces completeness, i.e., that all jobs are assigned to exactly one worker. Finally, (6) requires the start time of each job must be greater than the time required to travel to it from the depot.

Piecewise-Linear Modeling

We now discuss how to integrate time-varying problem data into our formulation. We seek to model the job durations and travel times as piecewise-linear (PWL) functions of time. For example, let us consider a generic PWL function $d = f(t)$ defined by points $(t_1, d_1), \dots, (t_N, d_N)$. We define the function as

$$f(t) = \begin{cases} d_1, & t \leq t_1, \\ d_i + (t - t_i) \frac{d_{i+1} - d_i}{t_{i+1} - t_i}, & t_i \leq t \leq t_{i+1}, \\ d_N, & t \geq t_N. \end{cases} \quad (7)$$

Thus, we seek a way to express our problem data as PWL functions. To do so, we will use the concept of Special Ordered Sets (SOS) [22], a common tool for handling PWL modeling in MILPs. These sets have two varieties; we say a vector $\mathbf{x} \in \mathbb{R}^N$ is in SOS(1) if only one of the entries x_i

is non-zero. Further, we say \mathbf{x} is in SOS(2) if all entries x_i are non-negative, at most two entries are non-zero, and these entries are adjacent. Support for SOS constraints is commonly included in MILP solver routines, such as Gurobi [21], which we use to generate the results presented in Section 6.

Thus, suppose in our problem we have variables t, d which we seek to constrain such that $d = f(t)$, where f is the PWL function (7) given above. To impose this constraint, we introduce an additional set of decision variables $\lambda \in \mathbb{R}^N$, with constraints

$$\begin{aligned} \sum_{i=1}^N \lambda_i t_i &= t, \\ \sum_{i=1}^N \lambda_i d_i &= d, \\ \sum_{i=1}^N \lambda_i &= 1, \\ \lambda_i &\geq 0, && i = 1, \dots, N, \\ \lambda &\in \text{SOS}(2) \end{aligned}$$

In short, we can interpret the λ_i as interpolation weights between adjacent knot points (t_i, d_i) ; the SOS(2) constraint is key to ensuring the function is interpolated only between adjacent points, rather than across all knot points.

Time-Varying Formulation

Using this technique, we can now introduce time-varying problem data into our formulation. To do this, we introduce a set of T knot points $\theta = [t_1, \dots, t_T]^T \in \mathbb{R}^T$, at which the job durations and travel times will be computed. Let $\mathbf{D}_j \in \mathbb{R}^T$ denote the vector of durations for job j , and $\mathbf{\Delta}_{j,k} \in \mathbb{R}^T$ denote the vector of transfer times between jobs j and k .

We also introduce two sets of non-negative variables, $\lambda \in \mathbb{R}^{n \times m \times T}$, which are interpolation weights for the job start times, and $\gamma \in \mathbb{R}^{n \times m \times T}$, which are weights for the job completion times. To conform to the PWL modeling, we impose

$$\sum_{i=1}^n \sum_{t=1}^T \lambda_{j,t}^i = 1 \quad \forall j \quad (8)$$

$$\sum_{i=1}^n \sum_{t=1}^T \gamma_{j,t}^i = 1 \quad \forall j \quad (9)$$

$$\lambda_{j,1:T}^i \in \text{SOS}(2) \quad \forall i, j \quad (10)$$

$$\gamma_{j,1:T}^i \in \text{SOS}(2) \quad \forall i, j \quad (11)$$

Finally, to incorporate the time-varying problem data, we will make \mathbf{d}, δ real-valued decision variables, and impose the constraints

$$\sum_{t=1}^T \lambda_{j,t}^i \theta_t = t_j^i \quad \forall i, j \quad (12)$$

$$\sum_{t=1}^T \sum_{i=1}^n \lambda_{j,t}^i d_t = d_j \quad \forall j \quad (13)$$

$$\sum_{t=1}^T \sum_{i=1}^n \gamma_{j,t}^i \Delta_{j,k} = \delta_{j,k} \quad \forall j, k \quad (14)$$

$$\sum_{t=1}^T \lambda_{j,t}^i (\theta_t + D_{j,t}) = \sum_{t=1}^T \gamma_{j,t}^i \theta_t \quad \forall i, j \quad (15)$$

$$\sum_{t=1}^T \gamma_{j,t}^i = \sum_{t=1}^T \lambda_{j,t}^i \quad \forall i, j \quad (16)$$

$$\sum_{t=1}^T \lambda_{j,t}^{1:n} \in \text{SOS}(1) \quad \forall j \quad (17)$$

$$\sum_{t=1}^T \gamma_{j,t}^{1:n} \in \text{SOS}(1) \quad \forall j \quad (18)$$

Here the constraints (12)-(14) require λ to represent the job start time, and impose the interpolation scheme discussed

previously for both job and travel times. Further, the constraint (15) requires the job completion time (on the RHS) to equal the start time of the job, plus its duration. With (16) we impose that jobs may have non-trivial final times iff they have non-trivial start times on the same agent. We note that the final two constraints, (17), (18), require the interpolation weights to have non-zero entries for only one agent per job, i.e., that each job is assigned to exactly one agent.

Thus, our full optimization problem is given by the constraints (2)-(6) and (8)-(18). Since \mathbf{z} is a set of binary variables, but all constraints and the objective are linear in the decision variables, the problem is indeed a MILP, for which efficient commercial solvers exist.

Implementation Details

We implement our problem in Gurobi [21], a commercial mixed-integer programming solver, using JuMP [23], a modeling package written in Julia. While the formulation described above is correct (i.e., optimal solutions solve the problem described in Section 3), we found empirically that performance could be improved by adding redundant inequalities to our formulation.

Specifically, we introduce a final intermediate variable $\zeta \in \mathbb{R}^{n \times m \times m \times T}$, and constrain

$$\sum_{k=1}^m \zeta_{j,k,t}^i = \gamma_{j,t}^i \quad \forall i, j, t \quad (19)$$

We also impose the typical PWL constraints on ζ , requiring

$$\begin{aligned} 0 &\leq \zeta_{j,k,t}^i \leq 1, \\ \sum_{i=1}^n \sum_{t=1}^T \zeta_{j,1:m,t}^i &\in \text{SOS}(1) && \forall j, \\ \sum_{i=1}^n \sum_{t=1}^T \zeta_{1:m,k,t}^i &\in \text{SOS}(1) && \forall k, \\ \zeta_{j,k,1:T}^i &\in \text{SOS}(2) && \forall i, j, k \end{aligned}$$

By introducing these intermediate variables, we can now capture when each robot transfers between specific jobs, allowing us to impose the (redundant) inequality constraints

$$\sum_{j=1}^m \sum_{t=1}^T \left(\lambda_{j,t}^i D_{j,t} + \sum_{k=1}^m \zeta_{j,k,t}^i \Delta_{j,k,t} \right) \leq C \quad \forall i \quad (20)$$

Empirically, we found these strengthening inequalities greatly improved the lower bound used by the solver, since the LP relaxation of the original problem typically had an optimal objective of zero.

5. PARAMETER GENERATION

We now turn to the problem of generating the problem data used in our MILP formulation from the more abstract problem specification described in Section 3. While our MILP formulation generates a schedule from job durations and travel times, our problem specification only includes a layout for the telescope and crater, a start time for the assembly process, and lighting-dependent speeds for the rovers. Thus, we need to compute both job durations (i.e., how much time is required to descend into the crater, retrieve the wire, and anchor it to the crater rim) and travel times between anchor locations. Since these times vary based on the time of day, we will compute them at a fixed set of knot points, and then approximate the true, continuous times as piecewise-linear interpolations between these knot points.

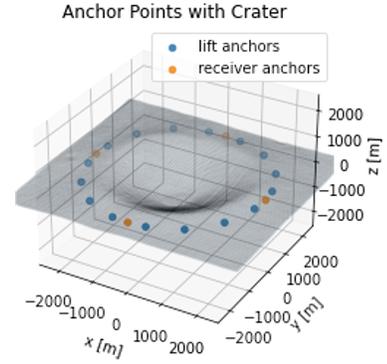


Figure 5: Isometric view of the prospective LCRT crater, with anchor points plotted. We use a “heat method” to reason about distances between the points and the crater center, which must be measured along the (curved) crater surface, instead of a straight line.

Illumination Computation

Our first problem is to compute the illumination of various points of the crater as a function of time. As mentioned in Section 3, we define starting time for the assembly operation as well as the location (latitude, longitude) of the crater on the Moon. Using this data, we can use the SPICE system [24] to compute the exact position and orientation of the crater with respect to the sun. This allows us to compute a relative direction of the sun from the crater normal, which we assume points outward along the ray from the moon’s center to the crater center.

In this work, we use a coarse illumination model, treating the sun as a directional light source (i.e., a point light at an infinite distance, with all light rays aligned along a single direction). Thus, we say a point on the crater is illuminated if the ray from the point to the sun does not intersect the moon, which would cause the point to be shadowed. Note this neglects variations in light intensity which would arise from reflection and scattering; we believe this coarse lighting model is a good first step for reasoning about navigability.

To implement this illumination model in software, we use the software package `trimesh` [25] to compute ray-mesh intersections for each point of interest along the ray pointing toward the sun, as generated by SPICE.

Distance Measurement

Further, while our problem specification includes lighting-dependent speeds for the rovers, we still must compute relative distances between the anchor points and the crater center to compute travel time (both along the rim, and into the crater). While the distance between anchor points can be easily approximated by the length of a planar arc connecting them, measuring distance to the crater center is more challenging. Because the rovers must travel along the crater surface, which is curved as shown in Figure 5, when descending into the crater, they will travel a farther distance than simply the Euclidean (or “straight-line”) distance between the anchor points and crater center. Thus, we seek to measure the “geodesic distance” (see [26] for a more formal discussion) along the crater surface.

This is a common problem in computational geometry, and there exist fast solvers for measuring geodesic distance on triangle meshes. In particular, for this work we use the

“heat method” [27], as implemented in the Python package `potpourri3d` [28]. The heat method uses fast finite-element method to compute geodesic distances along the crater surface, which allows us to generate accurate travel times both along the crater rim and along its walls.

Travel Time Computation

Now, since we can compute illumination of and distance between points on the crater, and since we are given illumination-dependent rover velocities, we should be able to compute the travel time between points.

However, one issue remains: the illumination of points can vary along the path, both as time passes and the rover’s location changes (e.g., moving into shadow as it descends into the crater). Thus, the rover’s velocity is non-constant along the path, which makes the typical travel time computation $\Delta t = \frac{d}{v}$ inaccurate, where d is the distance traveled, v is the (constant) velocity along the path.

Instead, the rover’s position (along the path from its origin to its destination) is governed by a non-autonomous ordinary differential equation (ODE),

$$\frac{dx}{dt} = v(x, t) = \begin{cases} v^\ell, & \mathcal{I}(x, t) = 1, \\ v^d, & \text{otherwise.} \end{cases} \quad (21)$$

where $\mathcal{I}(x, t)$ is an indicator function for if the point x is illuminated at time t , v^ℓ is the velocity of the rover in full illumination, and v^d is the rover velocity in darkness.

Thus, our problem is to solve for t_f such that

$$\int_0^{t_f} v(x, t) dt = d,$$

where d is the distance traveled.

To solve this, we use a technique common in optimal control [29] called “time rescaling.” We introduce a scaled time τ , with $t = \alpha\tau$, where we define $x(\tau = 0) = 0$, and $x(\tau = 1) = d$, and $\alpha > 0$ is the final time t_f .

We can now change variables in our ODE, yielding

$$\frac{dx}{d\tau} = \frac{dx}{dt} \frac{dt}{d\tau} = v(x, \alpha\tau)\alpha.$$

By introducing this scaled time, we can now treat the scaling factor α as a state in our ODE, with trivial dynamics $\frac{d\alpha}{dt} = 0$. Thus, we now have a two-point boundary value problem (2PBVP) in x, α , which we can solve with numerical routines.

Thus, to generate the drive times between points, we solve a 2PBVP, with velocities generated by the illumination checking procedure proposed previously. Practically, we sample a fixed number of points along each path, and a fixed number of timesteps at which we compute the illumination, and perform multilinear interpolation (in both position and time) to approximate illumination values. Computing the rim travel times is straightforward using this procedure; for job durations, we compute two 2PBVP solutions (one for descent, another for ascent), and add a constant factor to capture time required to grasp the cable at the lander, and to anchor it at the rim.

parameter	value
M_ℓ	16
M_r	4
v_r^ℓ	1×10^{-1} m/s
v_r^d	5×10^{-2} m/s
v_a^ℓ, v_a^ℓ	2×10^{-2} m/s
v_a^d, v_a^d	1×10^{-2} m/s
R	2.5×10^3 m

Table 1: Parameter Values for LCRT Construction Scenario

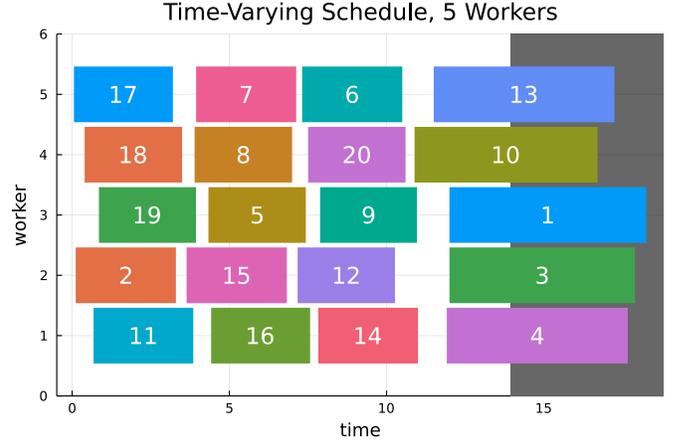


Figure 6: Planned assembly schedule with $n = 5$ agents. Each colored bar shows the start time and duration of a specific wire being anchored. The background is shaded according to the lighting of the crater; lunar night is shown with a dark background.

6. RESULTS

We now present numerical results demonstrating our assembly planner on an example LCRT design. Here we plan an assembly sequence for a telescope with $m_\ell = 16$ lift wires and $m_r = 4$ receiver wires, constructed within a 3km-diameter crater selected as a candidate by the LCRT team. Table 1 lists the parameter values used in the simulation, including velocity values for the rovers. To optimize all schedules, we allow the MILP solver to run for at most 10 minutes before returning the best feasible solution.

Figure 6 plots an example schedule sequenced for a team of $n = 5$ rovers. We can see the proposed schedule does not include any overlapping jobs between agents, and that the jobs that occur partially during lunar night have longer durations, as expected. We can also note that the end times of jobs all roughly align; further, the rovers (4 & 5) which seemingly end early perform jobs at wires located far from the depot (coincident with anchor 1). Thus, the optimized schedule is able to balance workloads effectively across all agents.

We further studied how the LCRT makespan returned by our solver varied with the number of rovers; this is an important data point for mission design, since additional rovers yield shorter makespans (and thus lower mission risk), but increase the mission cost. For this case study, we considered LCRT

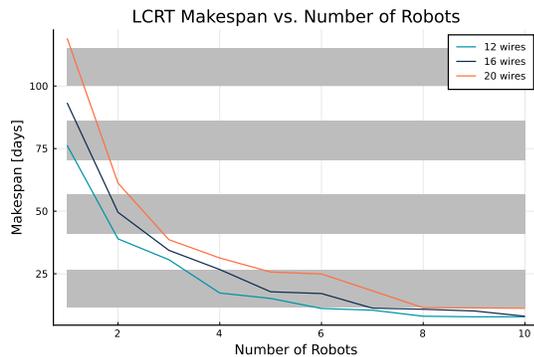


Figure 7: Optimized LCRT makespan plotted for the number of agents n varying between 1 and 10 and a variety of LCRT designs. We can see diminishing returns to the number of agents, with larger marginal speedups when n is small, as expected. Lunar night is represented by the shaded horizontal bars on the plot; teams with $n \geq 8$ can complete construction for all designs over the course of a single lunar day.

designs with $m_\ell = \{12, 16, 20\}$ lift wires and $m_r = 4$ receiver wires. Figure 7 plots the makespan for n varying between 1 and 10. We can observe diminishing returns to the number of agents, with larger marginal speedups (i.e., speedups from adding one extra rover) when n is small. We can also observe that when $n \geq 8$ the makespan is less than a single lunar day for all designs, meaning the team can complete LCRT in full sunlight. This is an important design point, since it means the assembly robots may not need to be designed to survive lunar night. As expected, the makespan increases with the number of wires for small teams, although the makespan converges to similar values as the number of robots grow, since the team can perform tasks in parallel.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a method for multi-robot assembly sequencing for the LCRT. We posed the scheduling problem as a mixed-integer linear program, and demonstrated how to include time-varying data in the problem formulation, by modeling job durations as piecewise-linear functions of time. We also provided an overview of how to generate this timing data from the realistic data (e.g., crater location and mesh, desired LCRT layout) provided as part of the design specification. Finally, we presented some numerical results of running our solver with some realistic parameter values, and observed that the assembly sequences returned were indeed reasonable, and observed diminishing returns in the number of agents n , as expected.

There exist numerous directions for future work. Perhaps the most compelling is finding ways to model stochastic job times and uncertainty in the planning problem. While in this work we considered the job and travel times to be deterministic, and known *a priori*, the real world is uncertain and unpredictable. There has been extensive work on stochastic vehicle routing [13], [12] which could provide interesting ways forward on reasoning rigorously about uncertainty inherent in the problem, such as modeling stops by robots when they cannot plan paths forward and require human assistance. Other interesting directions would include tighter formulations of the MILP; we typically terminate the solver before it returns a certificate of optimality for the schedule. Perhaps tighter formulations exist that would

speed performance, and allow for replanning or solutions on resource-limited computers. Finally, we are interested in performing higher-fidelity simulation of the assembly task in order to verify our modeling assumptions (especially that our interpolation schemes are appropriate).

ACKNOWLEDGMENTS

This work was supported in part by NASA Space Technology Graduate Research Opportunities (NSTGRO) Grant 80NSSC18K1180. Part of this research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. ©2022 All rights reserved.

REFERENCES

- [1] Saptarshi Bandyopadhyay, Patrick McGarey, Ashish Goel, Ramin Rafizadeh, Melanie Delapierre, Manan Arya, Joseph Lazio, Paul Goldsmith, Nacer Chahat, Adrian Stoica, Marco Quadrelli, Issa Nesnas, Kenneth Jenks, and Gregg Hallinan. Conceptual Design of the Lunar Crater Radio Telescope (LCRT) on the Far Side of the Moon. In *2021 IEEE Aerospace Conference (50100)*, pages 1–25, March 2021.
- [2] Issa AD Nesnas, Jaret B Matthews, Pablo Abad-Manterola, Joel W Burdick, Jeffrey A Edlund, Jack C Morrison, Robert D Peters, Melissa M Tanner, Robert N Miyake, Benjamin S Solish, et al. Axel and DuAxel rovers for the sustainable exploration of extreme terrains. *Journal of Field Robotics*, 29(4):663–685, 2012.
- [3] Patrick McGarey, William Reid, and Issa Nesnas. Towards articulated mobility and efficient docking for the duaxel tethered robot system. In *2019 IEEE Aerospace Conference*, pages 1–9. IEEE, 2019.
- [4] Albert Jones and Luis Rabelo. Survey of job shop scheduling techniques, 1998.
- [5] George Nemhauser and Laurence Wolsey. The scope of integer and combinatorial optimization. In *Integer and Combinatorial Optimization*, chapter 1.1, pages 1–26. John Wiley & Sons, Ltd, 1988.
- [6] Gilbert Laporte. The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(2):231–247, June 1992.
- [7] Tolga Bektas. The multiple traveling salesman problem: An overview of formulations and solution procedures. *Omega*, 34(3):209–219, June 2006.
- [8] A. Mor and M. G. Speranza. Vehicle routing problems over time: A survey. *4OR*, 18(2):129–149, June 2020.
- [9] Anders Dohn, Matias Sevel Rasmussen, and Jesper Larsen. The vehicle routing problem with time windows and temporal dependencies. *Networks. An International Journal*, 58(4):273–289, 2011.
- [10] Chryssi Malandraki and Mark S. Daskin. Time Dependent Vehicle Routing Problems: Formulations, Properties and Heuristic Algorithms. *Transportation Science*, 26(3):185–200, August 1992.
- [11] Miguel Andres Figliozzi. The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics. *Transportation Research Part E: Logistics*

- and *Transportation Review*, 48(3):616–636, May 2012.
- [12] Gilbert Laporte, François Louveaux, and H el ene Mercure. The Vehicle Routing Problem with Stochastic Travel Times. *Transportation Science*, 26(3):161–170, August 1992.
- [13] Bruce L. Golden and James R. Yee. A Framework For Probabilistic Vehicle Routing. *AIIE Transactions*, 11(2):109–112, June 1979.
- [14] Marjan van den Akker and Han Hoogeveen. Minimizing the number of late jobs in a stochastic setting using a chance constraint. *Journal of Scheduling*, 11(1):59–69, February 2008.
- [15] P. Jim enez. Survey on assembly sequencing: A combinatorial and geometrical perspective. *Journal of Intelligent Manufacturing*, 24(2):235–250, April 2013.
- [16] Preston Culbertson, Saptarshi Bandyopadhyay, and Mac Schwager. Multi-Robot Assembly Sequencing via Discrete Optimization. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6502–6509, November 2019.
- [17] Kyle Brown, Oriana Peltzer, Martin A. Sehr, Mac Schwager, and Mykel J. Kochenderfer. Optimal Sequential Task Assignment and Path Finding for Multi-Agent Robotic Assembly Planning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 441–447, May 2020.
- [18] Wolfgang H onig, Scott Kiesel, Andrew Tinka, Joseph W Durham, and Nora Ayanian. Conflict-Based Search with Optimal Task Assignment. page 9, 2018.
- [19] Jingjin Yu and Steven M. LaValle. Planning optimal paths for multiple robots on graphs. In *2013 IEEE International Conference on Robotics and Automation*, pages 3612–3617, May 2013.
- [20] Hang Ma, Glenn Wagner, Ariel Felner, Jiaoyang Li, T. K. Satish Kumar, and Sven Koenig. Multi-Agent Path Finding with Deadlines. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 417–423, Stockholm, Sweden, July 2018. International Joint Conferences on Artificial Intelligence Organization.
- [21] Gurobi Optimization, LLC. Gurobi optimizer reference manual, 2021.
- [22] E. Beale and John Tomlin. Special facilities in a general mathematical programming system for nonconvex problems using ordered sets of variables. *Operational Research*, 69:447–454, January 1969.
- [23] Iain Dunning, Joey Huchette, and Miles Lubin. JuMP: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.
- [24] Charles Acton, Nathaniel Bachman, Boris Semenov, and Edward Wright. A look towards the future in the handling of space science mission geometry. *Planetary and Space Science*, 150:9–12, January 2018.
- [25] Dawson-Haggerty et al. trimesh. <https://trimsh.org/>.
- [26] F. Flaherty and M.P. do Carmo. *Riemannian Geometry*. Mathematics: Theory & Applications. Birkh user Boston, 2013.
- [27] Keenan Crane, Clarisse Weischedel, and Max Wardetzky. The heat method for distance computation. *Communications of The Acm*, 60(11):90–99, October 2017.

- [28] Nicholas Sharp. potpourri3d. <https://github.com/nmwsharp/potpourri3d>.
- [29] U. Ascher and R. D. Russell. Reformulation of Boundary Value Problems into “Standard” Form. *SIAM Review*, 23(2):238–254, April 1981.

BIOGRAPHY



Preston Culbertson is a PhD student in the Multi-Robot Systems Lab at Stanford. He obtained an MS in Mechanical Engineering from Stanford in 2020, and a BS in Mechanical Engineering from Georgia Tech in 2016. His research is centered on problems in multi-robot manipulation and assembly, especially in uncertain or unstructured environments. He is specifically interested in enabling teams of robots to solve challenging problems (like manipulation and navigation) under uncertainty about their environment and their teammates’ actions. He is currently supported by the 2018 NASA Space Technology Graduate Research Opportunities (NSTGRO) Award, and received the 2018 ICRA Best Manipulation Paper Award.



Saptarshi Bandyopadhyay is a Robotics Technologist at the NASA Jet Propulsion Laboratory, California Institute of Technology, where he develops novel algorithms for future multi-agent and swarm missions. He was recently named a NIAC fellow for his work on the Lunar Crater Radio Telescope on the far-side of the Moon. He received his Ph.D. in Aerospace Engineering in 2016 from the University of Illinois at Urbana-Champaign, USA, where he specialized in probabilistic swarm guidance and distributed estimation. He earned his Bachelors and Masters degree in Aerospace Engineering in 2010 from the Indian Institute of Technology Bombay, India, where as an undergraduate, he co-founded and led the institute’s student satellite project Pratham, which was launched into low Earth orbit in September 2016. His engineering expertise stems from a long-standing interest in the science underlying space missions, since winning the gold medal for India at the 9th International Astronomy Olympiad held in Ukraine in 2004. Saptarshi’s current research interests include robotics, multi-agent systems and swarms, dynamics and controls, estimation theory, probability theory, and systems engineering. He has published more than 40 papers in journals and refereed conferences.



Ashish Goel is a Research Technologist at the Jet Propulsion Laboratory in the Robotic Surface Mobility group. Ashish received his Masters and PhD in Aeronautics and Astronautics from Stanford University where he worked with Prof. Sigrid Close, developing sensors and techniques for the detection and characterization of meteoroid and orbital debris impacts in space. He received his bachelor’s degree in Engineering Physics from Indian Institute of Technology Bombay. Prior to joining JPL, Ashish also worked as a postdoctoral researcher with Prof. Sergio Pellegrino in the Graduate Aerospace Laboratories at Caltech.

He developed optimized trajectories for a planar formation of space solar power satellites. He also served as a systems engineer and worked on the design of flight electronics for the Autonomous Assembly of a Reconfigurable Space Telescope (AAReST) project.



Patrick McGarey is a Robotics Technologist at JPL in the Robotic Mobility Group. Patrick received his PhD in Aerospace Engineering from the University of Toronto, where he was a visiting Fulbright Scholar. His research is focused on the development of systems and autonomy functions for the exploration of extreme environments throughout the solar system. Currently, Patrick is working

on the Axel/DuAxel rover to enable advanced mobility for the exploration of RSL features on Mars and lava tubes on the Moon.



Mac Schwager is an associate professor with the Aeronautics and Astronautics Department at Stanford University. He obtained his BS degree in 2000 from Stanford University, his MS degree from MIT in 2005, and his PhD degree from MIT in 2009. He was a postdoctoral researcher working jointly in the GRASP lab at the University of Pennsylvania and CSAIL at MIT from 2010 to 2012,

and was an assistant professor at Boston University from 2012 to 2015. He received the NSF CAREER award in 2014, the DARPA YFA in 2018, and a Google faculty research award in 2018, and the IROS Toshio Fukuda Young Professional Award in 2019. His research interests are in distributed algorithms for control, perception, and learning in groups of robots, and models of cooperation and competition in groups of engineered and natural agents.