Multiobjective Waypoint Sequencing for Planetary Rovers with Time-Dependent Energy Constraints

Christopher Cunningham[1], Jonathan Joo[2], Heather L. Jones[1], William L. "Red" Whittaker[1]

[1] *Carnegie Mellon University, Pittsburgh, PA*
e-mail: ccunningham@cmu.edu, hlj@cs.cmu.edu, red@cmu.edu

[2] *California Institute of Technology, Pasadena, CA*
e-mail: joo@caltech.edu

ABSTRACT

Robots will be the first to discover and characterize ices that exist at the poles of some moons and planets. These distinctive regions have extensive, grazing, time-varying shadows that raise significant time and energy constraints for solar-powered robots. In order to maximize the science-value of missions in such environments, rovers must visit as many targets as possible while considering limitations imposed by time-varying shadows and risks associated with traveling long distances. This paper compares a greedy baseline algorithm with two genetic algorithm approaches for selecting and sequencing waypoints to maximize waypoint value while minimizing distance traveled. The value and diversity of solutions from the baseline greedy solution, a single-objective genetic algorithm, and an NSGA-II framework are compared for this multiobjective optimization problem. All genetic solutions are shown to find high value sequences as compared to the greedy algorithm. This research demonstrates that a genetic approach could be utilized to effectively plan future missions for solar-powered rovers in dynamic, shadowed environments.

1 INTRODUCTION

Rover missions to the lunar poles will be shorter and faster-paced than current and previous Mars missions. Due to lunar temperature extremes, a rover will not last for years; instead, missions may even be confined to a two-week lunar sunlight period [1]. This contrast in mission lengths necessitates varied approaches to mission planning. Tactical planning for the Mars rovers involves planning out about a day of operations at a time and then waiting for the rover to execute those operations. Here, strategic planning, which happens on a longer time scale, is decoupled from tactical planning. On the Moon however, available communications and abundant solar power enable nearly 24-7 operation with seconds of latency for the mission duration [2, 1].

However, staying in the sun requires careful spatiotemporal path planning, since oblique illumination angles cause long shadows, which transform with time as the Moon rotates (see Figure 1). Tactical decisions can and must be made quickly, but must also carefully consider how a given action might impact subsequent mission goals. For example, stopping for hours to drill in one location could prevent a solar-powered rover from traveling to another interesting region, as drilling time coincides with the short time window when the connecting path is illuminated. Thus, the lack of constant sunlight availability can make some regions unreachable with a rover's battery capacity.

Missions with such short time frames must maximize their utility by visiting as many high-value science targets as possible. Rover missions also prefer to minimize distance traveled, since longer paths are inherently riskier. Unfortunately, as the number of desired waypoints increases, an exact solution to the question of which waypoints to visit and in what order becomes impractical [3]. To address this problem, this paper evaluates the efficacy of several metaheuristic algorithms for planning waypoint sequences with large numbers of waypoints (~50) that run within a reasonable time frame.

This paper applies two genetic algorithms to the

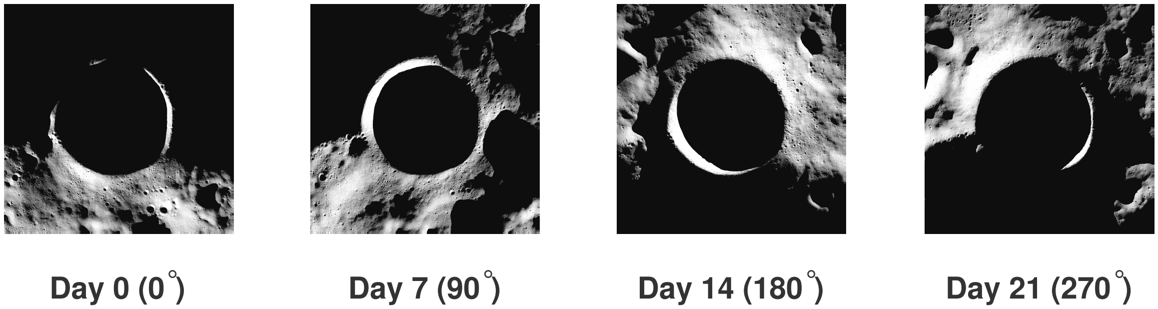


Figure 1: Time-varying lighting conditions on Shackleton Crater, located at the Lunar South Pole.

problem – a single-objective genetic algorithm and a multiobjective NSGA-II [4]. Both algorithms start with populations generated randomly and using deterministic and probabilistic greedy algorithms. These algorithms maximize the value of waypoints while minimizing distance and remaining within energy and time bounds. Heuristics guide the evolution of each generation, which helps minimize runtime while improving solutions. Performance of both the genetic algorithms and the greedy algorithms is tested on simulated lunar polar data and efficacy is evaluated based on runtime, the highest value path, and the solution diversity.

The paper is organized as follows. First, related work and a formal definition of the waypoint sequencing problem are given. Next, the waypoint sequencing algorithms are presented and explained. Experiments are described and analyzed in the Experiment and Results section. The Conclusions section discusses conclusions and directions for future research.

2 RELATED WORK

The problem of waypoint sequencing under time-varying illumination subject to non-monotonic energy constraints is similar to, but distinct from, several variants of the Orienteering Problem (OP), in which the agent maximizes reward while minimizing path length [5]. Variants of the OP include: the Time-Dependent Orienteering Problem (TDOP) [6], the Orienteering Problem with Time Windows (OPTW) [7], the Time-Dependent Orienteering Problem with Time Windows (TD-OPTW) [8], the Sequence Dependent Team Orienteering Problem (SDTOP) [9], and the Generalized Orienteering Problem [10]. In the TDOP, edge costs between two waypoints are dependent on the time at which the path starts from one waypoint to another. In the OPTW, each waypoint is only available during a limited time. The TD-OPTW combines TDOP and OPTW. In the SD-TOP, waypoints have different value depending on the visitation order. Finally, in the GOP, the total value of all waypoints is a non-linear function of a set of waypoint attributes.

Prior researchers have applied the OP to rover planning [11, 12]. Others have proposed numerous approaches to solving the OP and its variants. Gunawan et. al. [13] pose the TDOP as an integer linear programming model and utilizes metaheuristics for a solution. Duque, Lozano, and Medaglia solve the OPTW with recursive depth-first search and a set of pruning methods [14], though this requires a priori knowledge of edge costs. Wang et. al. [10] and Karbowska et. al. [15] use genetic approaches to solve the GOP and OPTW, respectively.

While multiobjective OP variants are often dealt with by constraining one objective (i.e. distance) while maximizing the other (i.e. waypoint value), other algorithms address multiobjective optimization directly, one being NSGA-II, a genetic algorithm variant [4]. Instead of finding a single solution, it finds a set of solutions along the pareto-optimal front. Since mission planers will have many preferences that are difficult to take into account, providing a set of non-dominated solutions to choose from is desirable.

The distinction of this waypoint sequencing problem is the incorporation of rover energy as a non-monotonic resource, making the path cost between waypoints depend not only on start time, but also the sequence of past waypoints. Energy constraints also control the times at which waypoints are available. Thus time windows are not explicitly known a priori but are implicit functions of energy constraints. If, for a period of time, a waypoint requires too much shadowed travel to reach or too much time in shadow to complete an action at the waypoint, energy constraints will prevent travel to that waypoint during that period. Edge costs are not known a priori and are sequence dependent, unlike in the OP variants discussed above. However, the sequence affects the edge costs instead of the value gained from waypoints, as in the Sequence Dependent Team Orienteering Problem (SDTOP) introduced by Mennell [9].

3 PROBLEM DEFINITION

This section provides a formal definition of the problem of waypoint sequencing under time-varying illumination subject to non-monotonic energy constraints. Solar powered rovers expend energy but recharge while illuminated. A rover has maximum and minimum energy levels, E_{max} and E_{min}, respectively. Illumination varies spatiotemporally, so to stay above E_{min}, a rover may take different paths between the same two points given different start times or energies.

Given a set W of N waypoints, located spatially in an environment with spatiotemporal illumination variation, the waypoint sequencing problem seeks to find an ordering for some subset of these waypoints. The rover arrives at waypoint w_i at time a_i with energy e_i and leaves at time l_i with energy ϵ_i For each w_i, there is an associated value, p_i, as well as time and energy costs that the robot must incur to complete the action, τ_i and $\delta E_i^{a_i,e_i}$, respectively.

The path from waypoint w_i to waypoint w_j has some distance, $d_{ij}^{l_i,\epsilon_i}$, takes some time, $\tau_{ij}^{l_i,\epsilon_i}$, and causes some change in energy, $\delta E_{ij}^{l_i,\epsilon_i}$, which all depend l_i and ϵ_i. The rover starts at a waypoint, w_1, at time t_{start} and must reach an ending waypoint, w_N before time t_{end}. The variable $x_{ij} = 1$ if the rover visits waypoint w_i immediately before waypoint w_j. The objective is to maximize the total value, $P = p_1 + \sum_{i=1}^{N-1} \sum_{j=2}^{N} x_{ij} p_j$, of waypoints visited, while minimizing total distance traveled, $D = \sum_{i=1}^{N-1} \sum_{j=2}^{N} x_{ij} d_{ij}$, and satisfying the constraints in Eqs. 1-6.

Following a formulation similar to [5], Eq. 1 enforces the start and end waypoint, ensures path continuity and makes sure each waypoint is visited at most once. Eq. 2 and Eq. 4 ensure that costs τ_i, $\delta E_i^{a_i,e_i}$, and $\delta E_{ij}^{l_i,\epsilon_i}$ are met and that energy does not exceed E_{max}. Eq. 5 enforces energy bound E_{min}. Eq. 6 enforces time bounds t_{start} and t_{end}.

$$\sum_{j=2}^{N} x_{1j} = \sum_{i=1}^{N-1} x_{iN} = 1, \ \sum_{i=1}^{N-1} x_{ik} = \sum_{j=2}^{N} x_{kj} \leq 1 \quad (1)$$

$$l_i - a_i \geq \tau_i \quad (2)$$

$$\epsilon_i = \min(\delta E_i^{a_i,e_i} + e_i, E_{max}) \quad (3)$$

$$e_j = \min(\delta E_{ij}^{l_i,\epsilon_i} + \epsilon_i, E_{max}) \quad (4)$$

$$E_{min} \leq \epsilon_i, E_{min} \leq e_i, \forall i = 1, ..., N \quad (5)$$

$$t_{start} + l_1 - a_1 + \sum_{i=1}^{N-1} \sum_{j=2}^{N} x_{ij} \left(\tau_{ij} + l_j - a_j \right) \leq t_{end} \quad (6)$$

4 WAYPOINT SEQUENCING

Two genetic algorithm approaches were tested: a single-objective genetic algorithm, which optimizes P but, given two paths of equivalent value, chooses lower D; and the NSGA-II algorithm, which optimizes both P and D simultaneously [4]. For both genetic approaches, the population is seeded with a set of feasible paths generated using randomized greedy algorithms. All algorithms call an energy-aware point-to-point planner to get path costs between waypoints.

4.1 Waypoint-to-Waypoint Planning

Point-to-point planning for solar-powered rovers under time-varying illumination has been addressed both by Tompkins [16] and more recently Cunningham et al. [17]. Both developed planners that considered energy and time constraints while using deterministic A*-based planning. Otten et al. also addressed long-duration point-to-point path planning on the lunar poles but did not explicitly address energy constraints [2].

This research directly follows the approach of Cunningham et al. for point-to-point planning [17]. An energy-aware waypoint-to-waypoint planner is used to find $d_{ij}^{l_i,\epsilon_i}$, $\tau_{ij}^{l_i,\epsilon_i}$, and $\delta E_{ij}^{l_i,\epsilon_i}$ for all paths between waypoints as well as the energy cost for visiting a waypoint, $\delta E_i^{a_i,e_i}$.

In order to represent the dynamic nature of the environment at the lunar poles, the world is discretized into a time series of shadow maps. Each shadow map represents the state of the world for a discrete time interval, T. A graph is used to encode the world in a representation that a planner can easily interpret. A node of the graph, $V = (x, y, T)$, corresponds to a position, (x, y), at a specific time interval, T. Edges are created using directed edges in an 8-connected grid. Nodes are connected to other nodes at the same time interval, T, and at the next time interval, $T + 1$. For each pair of connected nodes, there is a set of different velocities that the rover can use that allows different energy and time transitions. Energy and time costs for each edge are calculated using solar power derived from the shadow maps and physics-based rover models.

The point-to-point planner's A*-based search algorithm minimizes time while keeping all nodes in the path within predefined energy bounds derived from the battery capacity. Energy, E, and time, t, are continuous variables also considered as part of the search state and must stay within predefined constraints. The relevant energy and time values for a state are computed during the search and are not known a priori. Consequently, the planner does not plan directly over the graph nodes. Instead, it dynamically creates a new state $S = (x, y, T, t, E)$ every time it adds to the open list. Multiple (x, y, T, t, E) states correspond to the same (x, y, T) node in the graph but with different energy and time values. State dominance is used to prevent this method from creating an unbounded number of states.

Energy is a non-monotonic resource cost. It can either increase or decrease over an edge. Falling below E_{min} causes states to be pruned, and no more energy can be added to the battery above E_{max}.

When the point-to-point planner is asked to plan from a starting state to a waypoint, it plans to a set of goal nodes at that (x, y) location using Euclidean distance divided by maximum speed as the heuristic. Because each waypoint has a specified time cost T_i, when the planner opens a state at the goal location, it tests to see whether it can complete its objective at that location and still remain within energy and time constraints before determining that it has completed the search.

4.2 Greedy and Probabilistic Greedy

The greedy algorithm iteratively selects the next-best waypoint to form a full sequence, beginning with the start waypoint and time. At each step, it checks if the end can be reached from the current waypoint, and uses a heuristic function to measure the cost to visit each remaining waypoint. If the end cannot be reached before t_{end}, then the current waypoint is removed and replaced with the final waypoint. Otherwise, the algorithm greedily chooses the next waypoint with the highest heuristic value. The heuristic value function for traveling from w_i to w_j is:

$$H_{i \to j} = \frac{p_j}{\tau_{ij}^{l_i,\epsilon_i}} - \frac{\log(p_j)}{p_j + \tau_{ij}^{l_i,\epsilon_i}} \tag{7}$$

The heuristic function prioritizes higher value locations while penalizing locations that take a long time to reach. Several other functions were evaluated, but this produced the best results in practice. In the worst case, the greedy algorithm examines every waypoint and at every step evaluates the path to each remaining waypoint, resulting in a time complexity of $O(n^2)$. The probabilistic greedy algorithm is the same as the greedy algorithm, but instead of always choosing the best waypoint to add next, it randomly picks the next waypoints with higher probabilities given to those with higher heuristic values.

4.3 Genetic Algorithms

For both genetic algorithms, an initial population is constructed of ρ_{size} randomly generated, γ_{size} probabilistic-greedy generated, and one deterministic-greedy generated waypoint sequences. During each generation, the best parents are selected and paired to produce children, which then undergo mutations and become the next generation. In order to directly compare the single-objective and NSGA-II, both loops stopped after the best child according to the single-objective had not changed for G_{num} generations. Both genetic algorithms are elitist. The single-objective always keeps the best parent from the previous generation, and the NSGA-II considers all parents and children when selecting the next generation.

Both methods ensure that the "best" individuals remain in the current population.

4.3.1 Fitness Selection

A subset of size σ_{size} is chosen from the population to undergo mating. The single-objective algorithm uses tournament grouping selection: the sample population is divided into σ_{size} groups; then the best sequence is added to the mating pool from each group. Tournament grouping selection was chosen because it is less likely to become stuck in local minima than simply selecting the best σ_{size} [18]. The NSGA-II genetic algorithm uses the standard non-domination and crowding distance sorting to select the mating pool in order to preserve solution diversity. Crowding distance is computed in objective space [4]. The final new population size after the entire evolution step is $(4 * (\sigma_{size} - 1)) + 1$.

4.3.2 Evolution

The evolution step turns a mating pool into a new generation using crossover, mutation, and cleaning.

Crossover: The crossover phase utilizes random mating selection and Edge-Recombination Crossover, which has the advantage of maintaining valid path segments of parents [10].

Mutation: The mutation step modifies child sequences generated in the crossover step to encourage varied exploration. Each child from the evolution step is copied, and the copy is mutated. Mutation occurs in three ways:

1. Addition: a random waypoint is added to the sequence with higher probibility of inserting close to another waypoint that is close in either time or Euclidean distance.
2. Delete: a random waypoint is removed.
3. Swap: two random waypoints are swapped.

All mutations are more likely to act on waypoints in the sequence that are connected to edges with high time cost. This heuristic guides mutations to improve sections of the sequence that cause the robot to wait or travel long distances, which are more likely to be suboptimal.

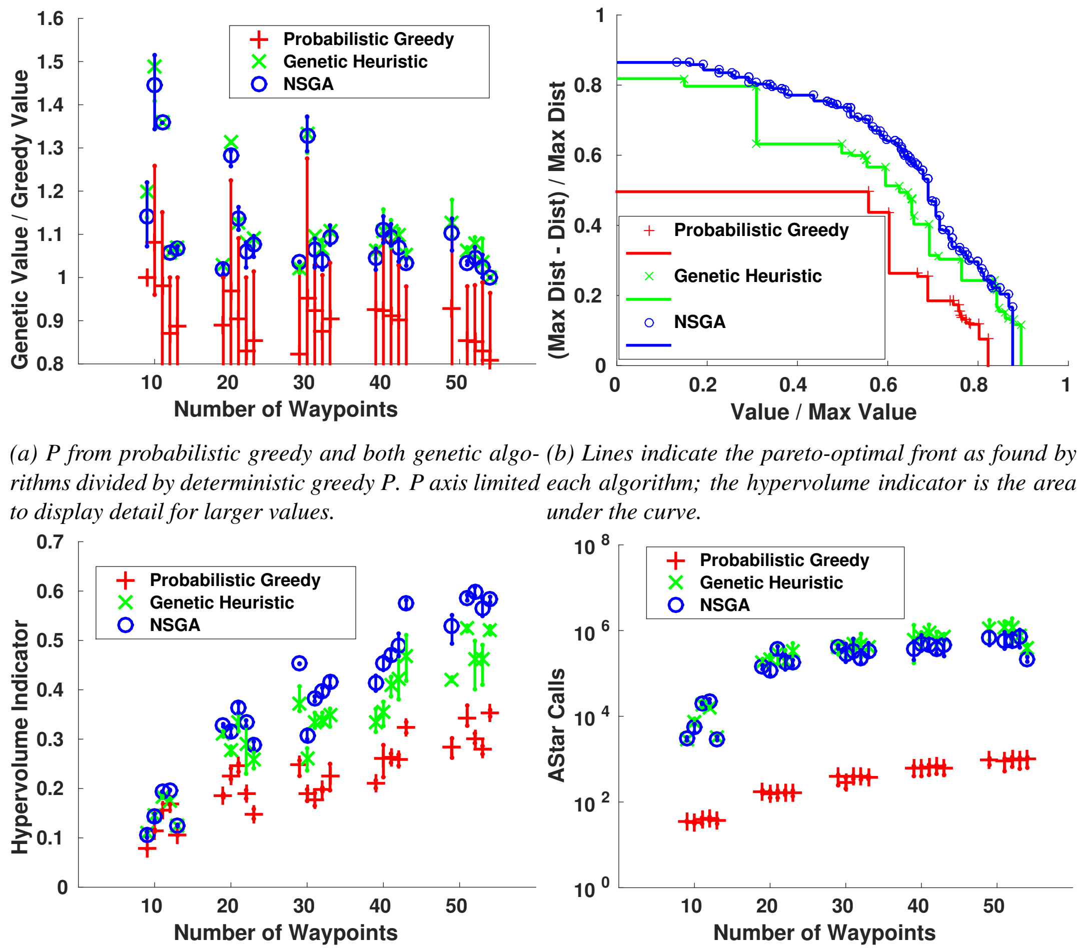
Cleaning: Because some children may not be viable paths, the cleaning step iteratively removes waypoints from a sequence (excluding the beginning and end waypoints) until a viable path exists through the sequence that satisfies all constraints. Heuristics are also added to cleaning, greatly improving the speed and resulting paths. Waypoints are removed randomly with greater probabilities assigned to sections of the path with high time cost. There is also a higher chance of removing either the last waypoint before the point-to-point planning failed or the waypoint it was planning to. Compared to fully random removal, these heuristics remove fewer waypoints, keeping the sequence quality high.

5 Experiments and Results

Experiments were run to test both the quality of paths returned and the computational efficiency for each waypoint sequencing algorithm. All experiments were run using 35 synthetically-generated shadow maps of Shackleton Crater on the lunar south pole, corresponding to several Earth days. Sets of waypoints were randomly generated within the map area. A graph of vertex locations and edge costs for the graph used in waypoint-to-waypoint planning was precomputed.

The number of waypoints, $\{10, 20, 30, 40, 50\}$, and the locations (within the map area) and values (from 1 to 10) of the waypoints were varied between tests. 10 waypoints was the minimum tested since fewer waypoints become trivial for a brute force solution. For each number of waypoints, five different test cases were generated with different locations and values. Values ranged between 1 and 10 for each waypoint. The planner was run on each test case four times. For both genetic algorithms, the mating population size, σ_{size}, for N waypoints was $(N/2)^3$ with a max of 500. The initial number of randomly generated sequences, ρ_{size}, was the same as σ_{size}. The initial number of greedily generated sequences, γ_{size}, was 35. The algorithms stopped after $G_{num} = 20$ generations with no change.

Algorithms were compared based on P value of the highest value sequence as a fraction of deterministic greedy value, the hypervolume indicator, and computational complexity. Highest value is compared because higher value is generally the more important goal. Minimum distance is a trival



(a) P from probabilistic greedy and both genetic algorithms divided by deterministic greedy P. P axis limited to display detail for larger values.

(b) Lines indicate the pareto-optimal front as found by each algorithm; the hypervolume indicator is the area under the curve.

(c) Hypervolume indicator for each algorithm.

(d) A calls for each algorithm.*

Figure 2: Results comparing each algorithm with four runs each on five different cases for each number of waypoints. Mean, min, and max are indicated where appropriate.

case in which the rover travels directly from start to finish without visiting any intermediary waypoints. Since the true highest possible value path is unknown, computed values are normalized by dividing by the value from deterministic greedy. Figure 2a shows the value comparison for the probabilistic greedy, single-objective genetic, and NSGA-II waypoint sequencing algorithms. The mean, minimum, and maximum for each of 5 test cases at each number of waypoints are shown. In most cases, the single-objective genetic algorithm obtains the highest value path. However, the NSGA-II algorithm is not far off in value. Both genetic algorithms improve upon their initial greedy populations in 93% of test cases. These is more value gain with fewer waypoints. This shows that at higher numbers of waypoints, the algorithms are likely not adequately exploring the solution space and higher populations are needed.

The hypervolume indicator, calculated as described in [19], is used to measure the diversity of solutions and the quality of non-dominated solutions found. Solutions with higher hypervolume indicators get a higher diversity of solutions than those with lower values. Waypoint sequence value and distance are normalized to the range $[0, 1]$ for this computation ($\frac{val}{max\ observed\ val}$ and $1 - \frac{dist}{max\ observed\ dist}$). Figure 2b illustrates the hypervolume indicator computation for one run. Note that probabilistic greedy does not explore the space very efficiently, single-objective genetic does reasonably well with the highest value, and NSGA-II explores the space most evenly. For each algorithm, Figure 2c shows the mean, maximum, and minimum hypervolume indicators computed for each of 5 test cases for each number of waypoints. Note that NSGA-II gets the highest hypervolume indicator fairly consistently, single-objective genetic is a close second, and probabilistic greedy does worst, especially as the number of waypoints increases. This result is expected, since the NSGA-II is explicitly seeking all pareto-optimal solutions.

The total number of point-to-point planner calls was used as a measurement of complexity because the point-to-point planner is called each time a waypoint-to-waypoint cost is requested. Evaluating a single waypoint sequence of N waypoints requires $N - 1$ planner calls. Figure 2d shows the number of point-to-point planner calls, mean, minimum, and maximum, that each algorithm makes for each of 5 test cases for each number of waypoints. Note that the log scale on this plot obscures much of the variability for individual test cases. The probabilistic greedy algorithm has a very low number of planner calls, while both genetic approaches are considerably higher, as expected. NSGA-II is slightly lower than single-objective genetic for large numbers of waypoints. The genetic algorithms are both significantly faster than a brute force solution. Instead of the time complexity being factorial in the number of waypoints, it is instead roughtly $O(NG\sigma_{size})$ where G is the number of generations. The solution is now linear in the number of waypoints. The quality of the solution is strongly affected by the number of generations and the population size. These two parameters can be controlled to either provide a very good solution with longer computation time or a faster solution that might be slightly less desirable.

6 CONCLUSION

This research has demonstrated that metaheuristic algorithms can be used to generate mission plans for energy-constrained rovers in an environment with significant time-varying shadows. These algorithms have polynomial time complexity, making them feasible for large numbers of waypoints. Both the greedy and probabilistic greedy algorithms can be used to quickly obtain relatively low-quality solutions. However, given more time, both genetic algorithms improve the initial greedy population in 93% of test cases. The single-objective algorithm more often finds a higher value solution, but NSGA-II often obtains a better variety of plans that mission planners could evaluate. Heuristics are used to great effect to guide the genetic algorithms, enabling better solutions in less time than purely random exploration.

It is likely that the NSGA-II algorithm would eventually approach an equal or higher value solution than the single-objective case but, because it explores evenly across the pareto-optimal front, it explores more slowly in the high-value region and prematurely reaches the ending condition. The NSGA-II algorithm could be modified by weighting the crowding distance more heavily in favor of value [20], increasing the population size and number of generations, or by changing the stopping conditions. In addition, current implementations are single-threaded and unoptimized. Future work could vastly decrease the runtime and increase population size through parallelization, which would further improve solutions.

Acknowledgement

This work was supported by the NASA Small Business Innovation Research program under contract NNX13CA55P. Two of the authors are also supported by NASA Space Technology Research Fellowships.

References

[1] J. George, G. Mattes, K. Rogers, D. Magruder, A. Paz, H. Vaccaro, R. Baird, G. Sanders, J. Smith, J. Quinn, *et al.*, "Resolve mission architecture for lunar resource prospecting and utilization," in *Lunar and Planetary Science Conference*, vol. 43, 2012, p. 2583.

[2] N. D. Otten, H. L. Jones, D. S. Wettergreen, and W. L. Whittaker, "Planning routes of continuous illumination and traversable slope using connected component analysis," in *International Conference on Robotics and Automation*, May 2015.

[3] C. P. Keller, "Algorithms to solve the orienteering problem: A comparison," *European Journal of Operational Research*, vol. 41, no. 2, pp. 224–231, 1989.

[4] K. Deb, A. Pratap, S. Agarwal, and M. T., "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182–197, 2002.

[5] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden, "The orienteering problem:

A survey," *European Journal of Operational Research*, vol. 209, no. 1, pp. 1–10, 2011.

[6] F. V. Fomin and A. Lingas, "Approximation algorithms for time-dependent orienteering," *Information Processing Letters*, vol. 83, no. 2, pp. 57 – 62, 2002

[7] M. G. Kantor and M. B. Rosenwein, "The orienteering problem with time windows," *The Journal of the Operational Research Society*, vol. 43, no. 6, pp. 629–635, Jun. 1992.

[8] A. Garcia, O. Arbelaitz, P. Vansteenwegen, W. Souffriau, and M. T. Linaza, "Hybrid approach for the public transportation time dependent orienteering problem with time windows," in *Hybrid Artificial Intelligence Systems*. Springer, 2010, pp. 151–158.

[9] W. K. Mennell, "Heuristics for Solving Three Routing Problems : Close-Enough Traveling Salesman Problem, Close-Enough Vehicle Routing Problem, Sequence-Dependent Team Orienteering Problem," PhD Thesis, University of Maryland, 2009

[10] X. Wang, B. L. Golden, and E. A. Wasil, "Using a genetic algorithm to solve the generalized orienteering problem," in *The vehicle routing problem: latest advances and new challenges*. Springer, 2008, pp. 263–274.

[11] D. E. Smith, "Choosing Objectives in Over-Subscription Planning." in *Intl. Conf. on Automated Planning and Scheduling*, 2004

[12] L. Pedersen, D. E. Smith, M. Deans, R. Sargent, C. Kunz, D. Lees, and S. Rajagopalan, "Mission Planning and Target Tracking for Autonomous Instrument Placement," in *Aerospace Conference*, Big Sky, MT, 2005, pp. 34–51.

[13] A. Gunawan, H. C. Lau, and Z. Yuan, *A Mathematical Model and Metaheuristics for Time Dependent Orienteering Problem*. Helmut-Schmidt-Univ., Univ. der Bundeswehr Hamburg, 2014.

[14] D. Duque, L. Lozano, and A. L. Medaglia, "Solving the orienteering problem with time windows via the pulse framework," *Computer & Operations Research*, vol. 54, pp. 168–176, 2015.

[15] J. Karbowska-Chilinska and P. Zabielski, "Genetic algorithm solving the orienteering problem with time windows," in *Advances in Systems Science*. Springer, 2014, pp. 609–619.

[16] P. Tompkins, "Mission-directed path planning for planetary rover exploration," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 2005.

[17] C. Cunningham, H. L. Jones, J. Kay, K. M. Peterson, and W. L. R. Whittaker, "Time-dependent planning for resource prospecting," in *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, June 2014.

[18] P. Zabielski, J. Karbowska-Chilinska, J. Koszelew, and K. Ostrowski, "A genetic algorithm with grouping selection and searching operators for the orienteering problem," in *Intelligent Information and Database Systems*. Springer, 2015, pp. 31–40.

[19] E. Zitzler, D. Brockhoff, and L. Thiele, "The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds. Springer Berlin Heidelberg, 2007, vol. 4403, pp. 862–876

[20] T. Friedrich, T. Kroeger, and F. Neumann, "Weighted preferences in evolutionary multi-objective optimization," *International Journal of Machine Learning and Cybernetics*, vol. 4, no. 2, pp. 139–148, 2012.