FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 50 (2025) No. 3
ISSN 0867-6356
sciendo DOI: 10.2478/fcds-2025-0012 e-ISSN 2300-3405

A Mixed Integer Programming Approach to the Rechargeable Rover Routing Problem on Mars

Wojciech Burzyński *, Mariusz Kaleta †

Abstract. In this paper, we introduce a novel variant of the Vehicle Routing Problem (VRP), the Rechargeable Rover Routing Problem (RRRP), which addresses the routing of energy-constrained autonomous electric rovers for Martian missions. We formulate a graph-based representation of the problem and propose an initial formulation as a mixed-integer non-linear program (MINLP). To enhance computational efficiency, we demonstrate how the model can be linearized. The resulting mixed integer linear model is evaluated on small-scale test cases, and its computational complexity is analyzed for larger problems with up to 30 Points of Interest (PoIs). Our experiments show that the problem can be solved to optimality for problem sizes anticipated in upcoming Mars expeditions. However, for future missions involving swarms of rovers, the development of more efficient heuristic or approximation algorithms will be necessary.

Keywords: task scheduling, electric VRP, VRP with profits, rover routing problem, mars exploration, MILP

1. Introduction

Since the 1990s, several missions have been undertaken to explore the surface of Mars using rovers. The first successful mission, *Pathfinder*, was launched on December 4, 1996. Its rover, *Sojourner*, landed on Mars on July 4, 1997, and operated for 85 Earth days. Subsequent missions included *Spirit* and *Opportunity*, launched in 2003. These missions significantly exceeded their expected operational lifetimes, lasting 8 and 14 years, respectively. Other successful missions followed, such as the Mars Science Laboratory (*Curiosity*) in 2012 and *Zhurong* in 2020, the first non-NASA rover. The most recent mission, Mars 2020, featured

*Warsaw University of Technology, Doctoral School, Pl. Politechniki 1, 00-661 Warsaw, Poland
wojciech.burzynski.dokt@pw.edu.pl, ORCID: 0009-0003-8668-7754
†Warsaw University of Technology, Faculty of Electronics and Information Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland
mariusz.kaleta@pw.edu.pl, ORCID: 0000-0002-2225-8956

the *Perserverence* rover and the *Ingenuity* helicopter, showcasing the potential for cooperative rover-drone operations. However, these missions highlight the high cost of advanced rover technologies and the risks associated with relying on a single rover, which could fail. Future missions are expected to mitigate these risks by deploying multiple smaller, simpler, and more cost-effective rovers. These fleets could continue missions even if some units are lost or damaged.

Inspired by past Mars missions, we address the problem of scheduling a fleet of rovers to traverse the Martian surface and perform a series of scientific tasks. Specific points of interest (PoIs) on the planet's surface are identified as potential sites for rovers to visit. At each PoI, a rover can perform research tasks such as taking pictures or conducting soil analysis. Given the limited number of rovers and their finite operational lifetimes, decisions must be made about which PoIs to visit and which to skip. This problem is akin to the classical Vehicle Routing Problem (VRP), where PoIs represent customers, and the Martian surface forms the road network. However, this scenario introduces unique constraints: not all PoIs need to be visited, rovers may cease operation after exceeding their lifetimes, and each rover must recharge periodically.

While the problem we consider could, in principle, be applied to other celestial bodies, this paper focuses exclusively on Mars. Mars is currently the primary target of active rover missions, and several upcoming missions—including multi-rover deployments—are planned. Additionally, our work is part of a broader research initiative at the Warsaw University of Technology, which includes the development of a physical environment simulating Martian conditions, along with real rovers designed for Mars missions. Since our goal is to develop methods applicable to the Martian environment, we remain within the context of Mars, despite the broader applicability of the proposed problem and approach.

Reactive approaches dominate planning and task allocation research for autonomous planetary vehicles, addressing challenges like coordination of homo- or heterogeneous agents, motion planning, and communication in known or partially known environments [20]. These studies, rooted in Multi-Robot Motion Planning (MMP), explore both decentralized [8, 21, 26] and centralized [6, 16, 27, 28] methods, often prioritizing collision avoidance, which may be overly detailed for planetary exploration. Offline roadmap computation for motion control is highlighted as a key efficiency gain [15].

Additional research perspectives include robotic swarms for mission longevity [22], human-robot collaboration for maximizing scientific data [5], and multi-robot cooperation in challenging terrains [11]. Notable approaches include task assignment using policy-driven swarm robots [10], coordination through centralized and decentralized schemes [4], and adaptive path planning under planetary constraints [3, 25]. Systems like ASPEN [23] and frameworks like CAMPOUT [11] exemplify operational advancements.

In the literature, the scheduling and routing of vehicles are commonly modeled as the Vehicle Routing Problem (VRP) [7]. The classic VRP minimizes route costs while visiting all customers exactly once. Variants such as the Vehicle Routing Problem with Profits (VRPP) or the Team Orienteering Problem relax the requirement to visit every customer [18]. Another relevant variant, the Electric Vehicle Routing Problem (EVRP), incorporates battery constraints and charging operations [14]. However, significant differences exist between charging electric cars on Earth and charging solar-powered rovers on Mars. Unlike electric cars, Martian rovers can recharge anywhere but may depend on favorable weather

and light conditions if they utilize solar panels.

Over the last decade, research on VRP-like problems has primarily focused on metaheuristic algorithms, such as genetic algorithms and Memetic algorithms [13]. In [30], the authors proposed the use of hybrid evolutionary algorithms to solve the VRP with a heterogenous fleet. Similarly, in [29], a genetic algorithm was applied to schedule robots tasked with collecting samples. Nevertheless, exact algorithms continue to play an important role [17]. For instance, in [24], a branch-and-cut algorithm was employed to solve the Electric Vehicle Routing Problem. Most inexact approaches aim to benchmark their performance against the best-known or optimal solutions, often obtained using exact algorithms [9].

To the best of our knowledge, the only work addressing energy-constrained electric vehicles with profits and limited vehicle lifetimes is our previous study on the Solar Powered Rover Routing Problem [2]. This paper extends that work by further developing the problem formulation and presenting a numerical evaluation of the proposed model.

The aim of this work is to develop a linear model and evaluate the feasibility of accurately solving the problem of planning rover routes on Mars. Our contributions can be summarized as follows:

- We formulate a novel problem, the Rechargeable Rover Routing Problem (RRRP).
- We propose a mixed-integer nonlinear programming (MINLP) model for the RRRP and demonstrate that it can be reformulated as a mixed-integer linear programming (MILP) problem.
- We show that very small-scale instances can be solved to optimality using standard MILP solvers based on branch-and-bound methods.
- Since slightly larger instances cannot be solved to optimality within 20 minutes, we suggest that the exact model can serve as a useful baseline. Further research is needed to develop other approaches, possibly heuristic or metaheuristic methods for solving larger instances efficiently.

Section 2 defines the problem, introduces a mathematical non-linear model, and presents its linearized version. Section 3 provides numerical evaluations of the model. Section 4 concludes with a discussion of the findings and potential future research directions.

2. Problem statement and model variants

2.1. Rechargeable Rover Routing Problem (RRRP)

The problem addressed in this paper involves scheduling the tasks of rovers in a fleet K, consisting of homogeneous units. Each rover has a limited expected lifetime and a battery that can be recharged, either via a radioisotope thermoelectric generator (RTG) or solar cells. The rovers are equipped with tools enabling them to conduct various scientific research tasks.

The possible tasks include traveling between two Points of Interest (PoIs), conducting research, or recharging at a given PoI. By the recharging task, we refer to a dedicated period

during which the rover focuses exclusively on the recharging process, with all other activities stopped. We assume that each task is atomic, meaning it must be completed without interruption; a task (including movement) cannot be paused in order to initiate the recharging task. However, this does not imply that recharging cannot occur concurrently with other activities—in practice, recharging may still take place during certain tasks, but such cases are not modeled as standalone recharging tasks within our formulation (see section 2.3 for the energy model).

Initially, all rovers are located at a starting PoI. A rover can perform only one task at a time, meaning it can either travel, conduct research, or recharge. The aim is to determine the most profitable task schedule that forms a route for each rover $k \in K$.

Let O denote the set of tasks (operations) scheduled in the solution. Only a subset of all possible tasks is included in the solution, as some PoIs may remain unvisited or certain routes between PoIs may not be traversed by any rover. For each task $o \in O$, we determine its starting and finishing times, s_o and e_o, respectively. Each task o must be assigned to a rover $k \in K$, represented by the decision variable x_{ok}, which equals 1 if rover k performs task o, and 0 otherwise.

Each task $o \in O$ yields a profit r_o of performing this task. Although this general formulation allows any task to have a profit, in most cases, only tasks related to conducting research are expected to have non-zero profits.

The Rechargeable Rover Routing Problem (RRRP) can be formulated as follows:

Rechargeable Rover Routing Problem (RRRP)

$$\max_{O,s_o,e_o,y_{ok},x_{ok}} \sum_{o\in O,k\in K} r_o \cdot y_{ok} \tag{1}$$

s.t.

$$\texttt{task logic and successions constraints} \tag{2}$$

$$\texttt{route logic constraints} \tag{3}$$

$$\texttt{energy constraints} \tag{4}$$

Constraints (3) ensure that tasks are scheduled in a logical sequence. For instance, conducting research at a specific PoI must be preceded by the rover's arrival at that PoI. Route logic constraints guarantee that routes are consistent, that is, each PoI is visited at most once, each rover starts from its designated initial location, and no cycles occur in the solution. Finally, energy constraints (4) account for energy consumption and the recharging process, ensuring that the schedule remains feasible within the energy limitations of each rover.

2.2. Mixed-integer non-linear model of RRRP

VRP problems are typically represented using graph-based representations. These models generally assume that each vehicle is either traveling between two nodes or stationed at a node where a specific task, like loading or unloading, is being performed. However, in the **RRRP** problem, a vehicle at a given PoI may be engaged in different activities, such as

conducting research or recharging. For this reason, obtaining a graph representation requires special treatment.

Let $Z = \{M, R, C\}$ denote the set of task types performed by a rover, where M, R, C correspond to *Movement*, carrying out *Research* and *Charging* batteries, respectively. To model the problem using a graph $G = (V, E)$, we follow the well-known task-on-arc representation commonly used for scheduling problems [1].

Unlike traditional VRP graph models, where arcs represent only traversal between nodes, in our model, each arc is assigned a specific task type, while vertices represent events. An event at a vertex can correspond to the start or end of a task, depending on whether the arcs flow into or out of the vertex. Task execution times and energy expenditures are associated with the arcs of the graph. The nodes, as representations of events, do not have a duration attribute but rather denote the time at which the event occurs.

Since we consider three types of tasks, each PoI is represented by three nodes in G. Figure 1 illustrates the graph construction. The left part of the figure shows a high-level network view, where each PoI is represented by a single node, and all rovers start at the initial node 0. In contrast, the right part of the figure shows the detailed graph G, where each PoI p is modeled as a cluster of nodes A_p, B_p, C_p representing different states of a vehicle visiting the PoI.

A rover visits PoI p by traversing an arc M leading to node A_p, representing the movement from the preceding PoI. At A_p, the rover can either:

1. Conduct a research task, represented by an arc of type R leading to node B_p, or
2. Recharge its battery, represented by an arc of type C leading to node C_p.

From node B_p, the rover may either recharge (arc C) or move to another PoI (arc M). Similarly, from node C_p, the rover may conduct research (arc R) or leave PoI p (arc M).

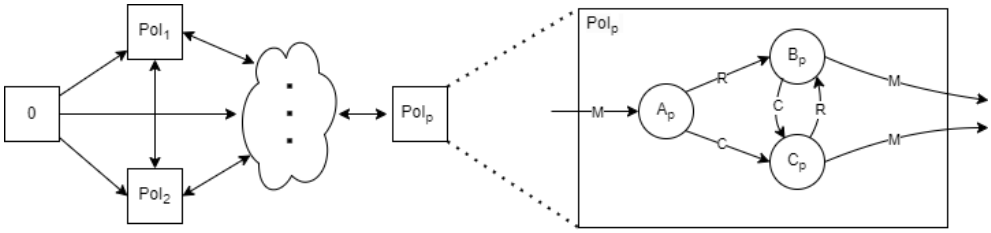

Figure 1. Graph model of the **RRRP** problem.

Based on the graph model, we construct a mixed-integer non-linear programming problem. Let us introduce the following decision variables:

- x_{ij}^k: Binary variable equal to 1 if rover k traverses edge (i, j), and 0 otherwise.
- y_j^k: Binary variable equal to 1 if rover k visits node j, and 0 otherwise. Setting y_j^k and x_{ij}^kcorrespond to executing task j.
- b_i^k: Battery energy level of rover k at event i.

- t_i^k: Time of occurrence of event i for vehicle k.

The objective maximizes the total reward from completing tasks, where r_{ij} is the reward assigned to task (i, j):

$$\max \sum_{(i,j)\in E} \sum_{k\in K} r_{ij} x_{ij}^k \tag{5}$$

$$s.t. \tag{6}$$

1. Starting conditions:

$$y_0^k = 1 \qquad \forall k \in K \tag{7}$$

$$\sum_{j\in V\setminus\{0\}} x_{0j}^k = 1 \qquad \forall k \in K \tag{8}$$

Constraint (7) ensures that all vehicles start from node 0, while constraint (8) guarantees that each vehicle leaves node 0 exactly once.

2. Task allocation

$$\sum_{i\in V} x_{ij}^k = y_j^k \qquad \forall j \in V \setminus \{0\}, \forall k \in K \tag{9}$$

$$\sum_{i\in V} x_{ji}^k - \sum_{i\in V} x_{ij}^k \leq 0 \qquad \forall j \in V \setminus \{0\}, \forall k \in K \tag{10}$$

$$\tag{11}$$

Constraint (9) ensures that if a rover is assigned to a task at node j, it must enter node j. Ensuring that a rover that leaves a node must first enter it is guaranteed by (10). However, rovers do not need to leave a node if it is their last task.

3. Node assignment

$$\sum_{k\in K} y_i^k \leq 1 \qquad \forall i \in V \tag{12}$$

$$\tag{13}$$

The above constraint ensures each node is visited by at most one vehicle.

4. Time constraints

$$t_0^k = T_0 \qquad \forall k \in K \tag{14}$$

$$t_j^k = \sum_{i\in V} x_{ij}^k (t_i^k + \tau_{ij}) \qquad \forall j \in V \setminus \{0\}, \forall k \in K \tag{15}$$

$$t_0 \leq t_i^k \leq t_{max} \qquad \forall k \in K, \forall i \in V \tag{16}$$

Constraint (14) sets the starting time for all vehicles. Tracking the time at each node j based on the time at the preceding node and the traversal time τ_{ij} is modeled by constraint 15. Constraint 16 ensures time constraints are respected, bounded by the mission start time t_0 and maximum allowable time t_{max}.

5. Battery constraints

$$b_j^k \leq \sum_{i \in V} x_{ij}^k (b_i^k + \Delta E(t_i^k, \tau_{ij}, z_{ij})) \qquad \forall j \in V \setminus \{0\}, \forall k \in K \tag{17}$$

$$0 \leq c_j^k \leq B \qquad \forall j \in V \setminus \{0\}, \forall k \in K \tag{18}$$

Constraint (17) tracks battery levels b_j^k, accounting for energy changes due to traversal, task execution, or recharging, where $\Delta E(t_i^k, \tau_{ij}, z_{ij})$ is energy change due to execution of task (i, j). Constraint (18) ensures that battery levels remain within allowable limits, bounded by 0 and the battery capacity B.

6. Integrality

$$y_i^k \in \{0, 1\}, x_{ij}^k \in \{0, 1\} \qquad \forall i \in V, \forall j \in V, \forall k \in K \tag{19}$$

The constraint enforces binary values for the decision variables x_{ij}^k and y_j^k.

Constraints (9) and (10) and (15) inherently prevent sub-tours, as time tracking enforces consistent route progression and eliminates unnecessary loops.

The non-linearity of the model arises from two sources. First, in constraints (15), there is a multiplication of variables, as the binary task assignment variable x_{ij}^k interacts with the continuous-time variable t_i^k to compute the starting time of subsequent tasks. Secondly, the energy function $\Delta E(t_i^k, \tau_{ij}, z_{ij})$ is inherently non-linear, as it depends on multiple factors, including the start time of the task t_i^k, its duration τ_{ij}^k, and the task type z_{ij}. This function often incorporates complex relationships, such as variable efficiency or environmental influences like solar flux for solar-powered rovers, adding to the non-linearity of the problem. These factors collectively make the problem computationally challenging and may require approximations or advanced optimization techniques for efficient solutions.

2.3. Modeling energy changes

The result of performing task (i, j) is a change in the energy $\Delta E(t_i^k, \tau_{ij}, z_{ij})$ in a battery of rover k. Let $P(z_{ij}, t) = P_+(z_{ij}, t) + P_-(z_{ij}, t), z_{ij} \in Z, t \in \mathbb{R}$ be the total power of the rover while performing task type z_{ij} vehicle at time t.

$P_-(z_{ij}, t)$ is the power consumed during the execution of a task of type z_{ij} at time t, and $P_+(z_{ij}, t)$ is power gained. Typically $P_+(z_{ij}, t)$ will be independent of z_{ij}, but it can correspond to engaging a power generator or unfolding solar panels.

The energy cost of a task type z_{ij} starting at t and lasting τ_{ij} is defined as follows:

$$\Delta E(t, \tau_{ij}, z_{ij}) = \int_t^{t+\tau_{ij}} P(z_{ij}, t) \mathrm{d}t = \tag{20}$$

$$\int_t^{t+\tau_{ij}} P_+(z_{ij}, t) + \int_t^{t+\tau_{ij}} P_-(z_{ij}, t) \tag{21}$$

Although the power consumption may vary while executing the task, we simplify function $P_-(z_{ij}, t)$ as follows:

$$P_-(z_{ij}, t) = \begin{cases} -a & z_{ij} = M \\ -b & z_{ij} = R \\ -c & z_{ij} = C \end{cases} \tag{22}$$

where a, b, c are constant power costs defined for each type of task. This simplifies the notation without loss of the generality since the function can be substituted with any differentiable function.

The function $P_+(z_{ij}, t)$ depends on the technology of recharging and environmental conditions like weather. For instance, a simplified model of solar panels may assume a periodical square wave function with a period equal to one sol (martian day). Assuming that a daylight period starts at $t = 0$ and lasts until $\frac{T}{2}$, where $T = 1$ sol is a period of the function, then for charging ($z_{ij} = C$), the function can be defined as follows:

$$P_+(z_{ij}, t) = \begin{cases} \alpha & t \in [0, \frac{T}{2}) \\ 0 & \in [\frac{T}{2}, T) \end{cases} \tag{23}$$

For RTG, a constant function can be a sufficient approximation of a short planning horizon.

To include aspects like dependence on time of day or weather, it is enough to change the definition of the function $P_+(z_{ij}, t)$, and that does not limit our approach. However, some phenomena, like the heating-up or the geographical position of a rover, may have some impact, and our approach may have some limitations in such cases.

2.4. Time discretization

The model, as presented in Sections 2.2 and 2.3, is nonlinear and, therefore, not directly solvable using MILP techniques. In particular, the battery constraint (17) involves the calculation of energy, which is a nonlinear function of time. To address this, we employ a standard technique that approximates nonlinear functions using piecewise-linear functions. This approach involves discretizing the domain of the function—in this case, discretizing time—and approximating the nonlinear function with a linear one within each time segment.

Let $T = \{t_0, \ldots, t_{max}\}$ represent the set of discrete time intervals. We introduce a binary variable s^k_{il}, which equals 1 if event i occurs at time l for rover k. The following constraints ensure proper time assignment and consistency:

$$t^k_i = \sum_{l \in T} l \cdot s^k_{il} \qquad \forall i \in V, \forall k \in K \tag{24}$$

$$\sum_{l \in T} s^k_{il} = y^k_i \qquad \forall i \in V, \forall k \in K \tag{25}$$

$$\sum_{i \in V} s^k_{il} \leq 1 \qquad \forall k \in K, \forall l \in T \tag{26}$$

Equation (24) ensures that the timestamp t_i^k for event i corresponds to the weighted sum of binary variables s_{il}^k. Constraint (25) ensures that a timestamp is assigned to event i only if it is visited (i.e., $y_i^k = 1$). Constraint (26) ensures that at most one event occurs at any given time for each rover.

Time discretization also requires rounding the traversal times τ_{ij} to align them with the discretized time scale. Consequently, we use adjusted traversal times τ'_{ij} in the time-tracking equation (15), which is updated as follows:

$$t_j^k = \sum_{i \in V} x_{ij}^k (t_i^k + \tau'_{ij}) \qquad \forall j \in V \setminus \{0\}, \forall k \in K \tag{27}$$

Let F_l^+ denote the total energy gained (e.g., from solar panels or RTGs) during time interval l. Since time intervals and necessary conditions such as weather are forecastable, the values of F_l^+ can be precomputed. Consequently, constraint (17) can be replaced with the following set of constraints:

$$c_j^k = \sum_{i \in V} x_{ij}^k (c_i^k + \kappa_{ij}) \qquad \forall j \in V \setminus \{0\}, \forall k \in K \tag{28}$$

$$g_j^k = \sum_{l=0}^{T} s_{jl}^k F_l^+ \qquad \forall j \in V, \forall k \in K \tag{29}$$

$$b_j^k = g_j^k - c_j^k \qquad \forall j \in V \setminus \{0\}, \forall k \in K \tag{30}$$

$$0 \leq b_j^k \leq B \qquad \forall j \in V \setminus \{0\}, \forall k \in K \tag{31}$$

where κ_{ij} represents the energy expenditure of the task (i, j). The variable c_j^k denotes the cumulative energy expenditure of all tasks performed from the beginning to the time of event j inclusively. This is analogous to the time-tracking constraint (15). The variable g_j^k represents the total energy gained from beginning to event j. Constraint (29), together with (24), links the discretized time to energy gain.

2.5. Linerarization of constraints

Constraints (15) and reformulated (28) both contain non-linear terms. However, they share a similar structure: the non-linearity arises from the multiplication of binary and non-binary variables. This issue can be addressed with a standard approach based on the big-M method.

Constraint (15) can be linearized as follows:

$$\begin{aligned} & t_j^k = \sum_{i \in V} x_{ij}^k (t_i^k + \tau_{ij}) = \\ & \sum_{i \in V} x_{ij}^k t_i^k + \sum_{i \in V} x_{ij}^k \tau_{ij} = \\ & \sum_{i \in V} d_{ij}^k + \sum_{i \in V} x_{ij}^k \tau_{ij}, \qquad \forall j \in V \setminus \{0\}, \forall k \in K \end{aligned} \tag{32}$$

where an auxiliary variable $d_{ij}^k = x_{ij}^k t_i^k$ is introduced. The following constraints must be added

to the model:

$$d_{ij}^k \leq T_{max} x_{ij}^k \qquad \forall i \in V, \forall j \in V \setminus \{0\}, \forall k \in K \tag{33}$$

$$d_{ij}^k \leq t_i^k \qquad \forall i \in V, \forall j \in V \setminus \{0\}, \forall k \in K \tag{34}$$

$$d_{ij}^k \geq t_i^k - (1 - x_{ij}^k) T_{max} \qquad \forall i \in V, \forall j \in V \setminus \{0\}, \forall k \in K \tag{35}$$

$$d_{ij}^k \geq 0 \qquad \forall i \in V, \forall j \in V \setminus \{0\} \tag{36}$$

Here, T_{max} is used as a substitute for the typical big-M value, representing an upper bound on time.

Similarly, constraint (28) can be rewritten in a linearized form:

$$\begin{aligned} c_j^k &= \sum_{i \in V} x_{ij}^k (c_i^k + \kappa_{ij}) = \sum_{i \in V} x_{ij}^k c_i^k + \sum_{i \in V} x_{ij}^k \kappa_{ij} = \\ &= \sum_{i \in V} e_{ij}^k + \sum_{i \in V} x_{ij}^k \kappa_{ij}, \qquad \forall j \in V \setminus \{0\}, \forall k \in K \end{aligned} \tag{37}$$

where an auxiliary variable $e_{ij}^k = x_{ij}^k c_i^k$ is introduced. The following constraints ensure linearity:

$$e_{ij}^k \leq B x_{ij}^k \qquad \forall i \in V, \forall j \in V \setminus \{0\}, \forall k \in K \tag{38}$$

$$e_{ij}^k \leq c_i^k \qquad \forall i \in V, \forall j \in V \setminus \{0\}, \forall k \in K \tag{39}$$

$$e_{ij}^k \geq c_i^k - (1 - x_{ij}^k) B \qquad \forall i \in V, \forall j \in V \setminus \{0\}, \forall k \in K \tag{40}$$

$$e_{ij}^k \geq 0 \qquad \forall i \in V, \forall j \in V \setminus \{0\} \tag{41}$$

Here, B serves as a substitute for the big-M value, representing the maximum possible energy capacity.

2.6. Final model

Based on the presented reformulations, the final model, incorporating time discretization and linearization, is as follows:

$$\max \sum_{(i,j) \in E} \sum_{k \in K} r_{ij} x_{ij}^k \tag{42}$$

s.t.

$$y_0^k = 1 \qquad \forall k \in K \tag{43}$$

$$\sum_{j \in V \setminus \{0\}} x_{0j}^k = 1 \qquad \forall k \in K \tag{44}$$

$$\sum_{i \in V} x_{ij}^k = y_j^k \qquad \forall j \in V \setminus \{0\}, \forall k \in K \tag{45}$$

$$\sum_{i \in V} x_{ji}^k - \sum_{i \in V} x_{ij}^k \leq 0 \qquad \forall j \in V \setminus \{0\}, \forall k \in K \tag{46}$$

$$\sum_{k \in K} y_i^k \leq 1 \qquad \forall i \in V \tag{47}$$

$$t_0^k = T_0 \qquad \forall k \in K \tag{48}$$

$$t_j^k = \sum_{i \in V} d_{ij}^k + \sum_{i \in V} x_{ij}^k \tau_{ij} \qquad \forall j \in V \setminus \{0\}, \forall k \in K \tag{49}$$

$$d_{ij}^k \leq T_{max} x_{ij}^k \qquad \forall i \in V, \forall j \in V \setminus \{0\}, \forall k \in K \tag{50}$$

$$d_{ij}^k \leq t_i^k \qquad \forall i \in V, \forall j \in V \setminus \{0\}, \forall k \in K \tag{51}$$

$$d_{ij}^k \geq t_i^k - (1 - x_{ij}^k) T_{max} \qquad \forall i \in V, \forall j \in V \setminus \{0\}, \forall k \in K \tag{52}$$

$$d_{ij}^k \geq 0 \qquad \forall i \in V, \forall j \in V \setminus \{0\}, \forall k \in K \tag{53}$$

$$t_i^k = \sum_{l \in T} s_{il}^k l \tag{54}$$

$$\sum_{l \in T} s_{il}^k = y_i^k \qquad \forall i \in V, \forall k \in K \tag{55}$$

$$\sum_{i \in V} s_{il}^k \leq 1 \qquad \forall i \in V, \forall k \in K, \forall l \in T \tag{56}$$

$$c_j^k = \sum_{i \in V} e_{ij}^k + \sum_{i \in V} x_{ij}^k \kappa_{ij} \qquad \forall j \in V \setminus \{0\}, \forall k \in K \tag{57}$$

$$e_{ij}^k \leq B x_{ij}^k \qquad \forall i \in V, \forall j \in V \setminus \{0\}, \forall k \in K \tag{58}$$

$$e_{ij}^k \leq c_i^k \qquad \forall i \in V, \forall j \in V \setminus \{0\}, \forall k \in K \tag{59}$$

$$e_{ij}^k \geq c_i^k - (1 - x_{ij}^k) B \qquad \forall i \in V, \forall j \in V \setminus \{0\}, \forall k \in K \tag{60}$$

$$e_{ij}^k \geq 0 \qquad \forall i \in V, \forall j \in V \setminus \{0\}, \forall k \in K \tag{61}$$

$$g_j^k = \sum_{l=0}^{T} s_{jl}^k F_l^+ \qquad \forall j \in V, \forall k \in K \tag{62}$$

$$b_j^k = g_j^k - c_j^k \qquad \forall j \in V \setminus \{0\}, \forall k \in K \tag{63}$$

$$0 \leq b_j^k \leq B \qquad \forall j \in V \setminus \{0\}, \forall k \in K \tag{64}$$

$$0 \leq t_i^k \leq T_{max} \qquad \forall k \in K, \forall i \in V \tag{65}$$

$$y_i^k \in \{0, 1\}, x_{ij}^k \in \{0, 1\} \qquad \forall i \in V, \forall j \in V, \forall k \in K \tag{66}$$

3. Numerical evaluation

3.1. Model validation

To validate the model, we prepared a simple test case involving one rover and two Points of Interest. The graph representation of the problem is shown in Fig. 2. Each task requires one unit of time ($\tau_{ij} = 1$), and energy consumption for each task is detailed in Table 1. The profit for each research task is $c_{ij} = 1$, where $z_{ij} = R$.

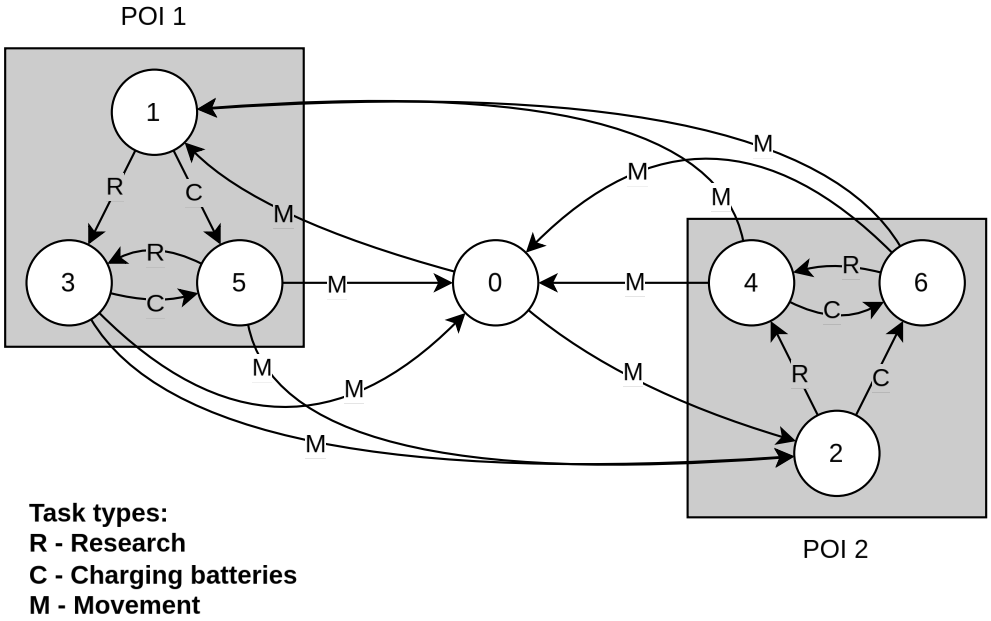

Figure 2. The graph model of the test problem. Edge labels indicate task types, while vertex labels represent task completion event identifiers. Gray rectangles highlight the grouping of nodes corresponding to the same Point of Interest (PoI). For instance, a tour 0-1-3-2-6-4-0 means conducting research at PoI1 (1-3), then moving to PoI2 (3-2), followed by recharging at PoI2 (2-6), conducting research at PoI2 (6-4).

Table 1. Energy consumption in the test problem.

Task type	Cost
M Base-PoI 1	6
M Base-PoI 2	10
M PoI-PoI	4
R	5
C	1

We assume that charging the battery is possible in time slots 1, 2, 4, and 5. Charging is not possible in the middle of the considered time horizon to simulate changing weather conditions. The definition of function $P_+(z,t)$ is illustrated in Fig. 3.

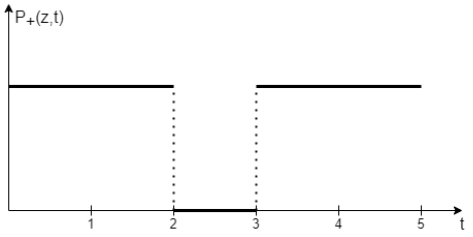

Figure 3. Function of power $P_+(z,t)$ in the test problem.

To observe the behavior of the optimal results, we varied the battery capacity installed on the vehicle. For a battery capacity of 5, no routes are found, as this capacity is insufficient to travel to any PoI. A battery capacity between 6 and 7 allows the rover to reach PoI 1, but no research can be conducted. For capacities between 8 and 13, the model schedules the rover to conduct research at PoI 1. Finally, for a battery capacity of 14 or higher, the rover is able to visit both PoIs and complete the associated research tasks.

To further validate the model's performance with multiple vehicles, a set of six PoIs arranged in a hexagonal pattern was prepared. It was assumed that traveling from the base to any PoI takes three units of time. Fleets of 1, 2, and 6 rovers were tested under different time constraints. When the available time was limited to 4 units, each rover could reach only one PoI, and only the fleet with six rovers was able to visit all PoIs. Increasing the time limit to 8 units allowed three rovers to reach all PoIs. When the time was extended to 12 units, only two rovers were required to visit every PoI. The results are summarized in Table 2. Example cases were shown on Figures 4 and 5.

Table 2. Results for varying fleet size and time limit

		Time limit		
		4	**8**	**12**
Fleet size	**1**	1	2	3
	2	2	4	6
	3	3	6	6
	6	6	6	6

3.2. Numerical complexity

To estimate the model's numerical complexity, we prepared datasets containing different numbers of PoI.

Datasets for the experiment were generated under the following assumption:

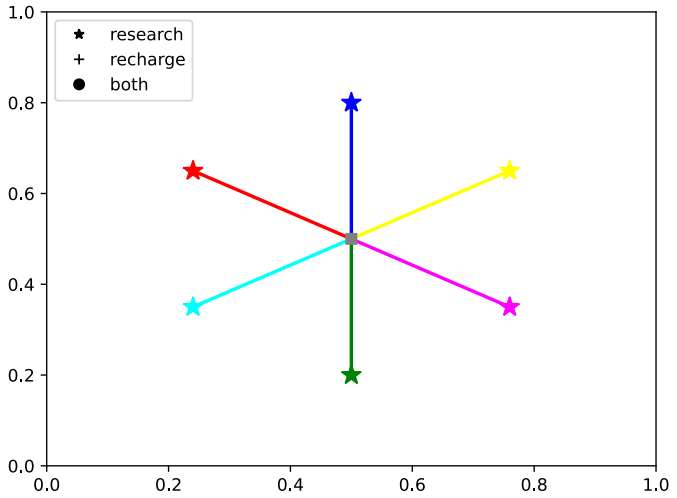

Figure 4. Visualization of test case for 6 rovers and 4 units of time. Each color represents a different vehicle.

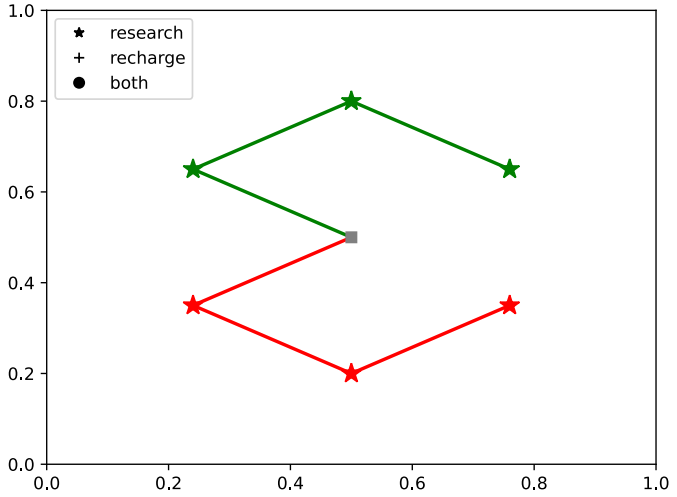

Figure 5. Visualization of test case for 2 rovers and 12 units of time. Each color represents a different vehicle.

- Assumed rover properties:
  - Battery capacity set to $B = 80$.
  - Speed set to 100 (used for further calculation of traveling times).
  - Research energy usage κ_{ij} set to 2 for $z_{ij} = R$.
  - Charge energy usage κ_{ij} set to 1 for $z_{ij} = C$.
  - Move energy usage (κ_{ij} for $z_{ij} = M$) has been calculated using the following formula: travel time × unit move energy cost, where travel time has been calculated based on the shortest path and assumed rover speed, and unit move energy cost is set to 3.
  - $F+$ from RTG - calculated as a linear function of time for each timestep.
- Generation of the map representing the environment was based on the following steps:
  - Preparation of the environment – an empty map of size 1km x 1km was filled with rectangle-shaped obstacles with random sizes and positions. The positions of the obstacles were sampled uniformly at random from the distribution $\mathcal{U}(\sim [0, 1] \times [0, 1])$. The sizes of the obstacles were also sampled uniformly at random from the distribution $\mathcal{U}(\sim [0, 0.1] \times [0, 0.1])$.
  - Points were placed on the map randomly. The positions were sampled uniformly at random from the distribution $\mathcal{U}(\sim [0, 1]\times[0, 1])$. One of the points was selected to be a base, while the rest were selected to be PoIs.
  - Algorithm PRM* was used to generate an intermediate graph containing base, all PoIs, and random intermediate points. For the details of the algorithm, see [12]. Because the result of this algorithm is a graph, a graph pathfinding algorithm is necessary to calculate the actual path. Algorithm A* was used to determine the shortest paths between each base and all PoIs and between every pair of PoIs within the intermediate graph. Probabilistic algorithms such as PRM, RRT, and their derivatives, as well as A*, are commonly used in robotics to plan a trajectory for the vehicles to follow. This combination serves not only as a means to generate a dataset but also to demonstrate how scheduling results can be tied with lower-level motion control procedures.
  - Travel times between PoI were calculated using the following formula: rover speed × shortest path length.
  - A final graph used in the evaluation is obtained by using PoIs and base as its vertices. Travel times are used as τ. κ is assigned using a procedure of calculating move energy usage. Additional vertices and edges are added to the graph to obtain the topology visible in figure 1.
- Profits from visiting PoI is set to 1 for each research task, and 0 for other tasks ($r_{ij} = 1$ for $z_{ij} = R$).
- Total mission time horizon: t_0 is set to 0, t_{max} set to 40, timestep in discretization is 1.

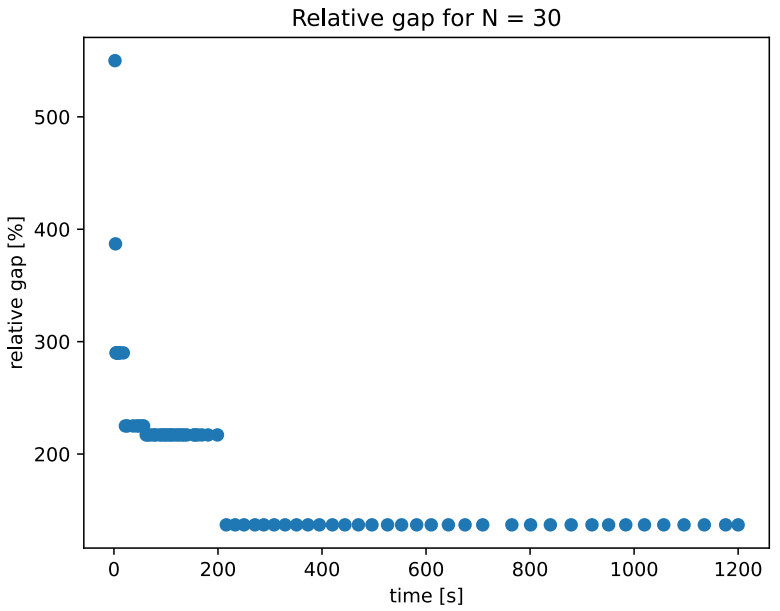

Figure 6. Gap measurement for n = 30 PoI

We generated datasets with the number of Points of Interest (PoIs) ranging from 4 to 40, considering a single rover. To solve the problem instances, we used AMPL with the Gurobi solver, setting a computation time limit of 20 minutes for all runs. The algorithm used by Gurobi is generally Branch-and-Bound-based, which is an exact algorithm. The algorithm provides information on the integrality gap (GAP), the difference between the current upper and lower bounds. If the computations are not stopped prematurely, the algorithm guarantees that GAP converges to 0, proving the optimality. Therefore, whenever we report that the GAP equals 0, it is guaranteed that the optimal solution has been found. The results are summarized in Table 3.

For instances with up to 8 PoIs, the solver successfully found optimal solutions within the time limit. However, for instances with more than 8 PoIs, it was not possible to obtain optimal solutions within 20 minutes. For instances with fewer than 14 PoIs, the integrality gap remained below 60%, whereas larger instances exhibited significantly higher gaps.

During the optimization process, the integrality gap typically decreased rapidly in the initial stages but showed only marginal improvements afterward. Figure 6 illustrates the progression of the gap for an instance with 30 PoIs. After approximately 200 seconds of computation, the gap stabilized at around 137% and showed minimal improvement over the subsequent 26 minutes of optimization.

Figures 7 and 8 present two selected solutions obtained from the optimization process. Every visited PoI and edge is marked in red. If the research was conducted at a PoI, a smaller green circle is drawn inside the red circle. Additionally, if the rover recharged its batteries at

Table 3. Best integers and GAPs for different numbers of PoIs.

No. of POI	Best Integer	Gap
4	3	**0%**
5	4	**0%**
6	5	**0%**
7	6	**0%**
8	7	**0%**
9	7	14.3%
10	8	12.5%
11	7	42%
12	8	57%
13	8	50%
14	7	160%
15	8	75%
16	7	114%
17	6	167%
18	6	183%
19	6	200%
20	5	280%
21	6	217%
22	9	111%
23	8	125%
24	8	125%
25	9	111%
26	9	111%
27	8	137%
28	7	172%
29	10	137%
30	8	137%

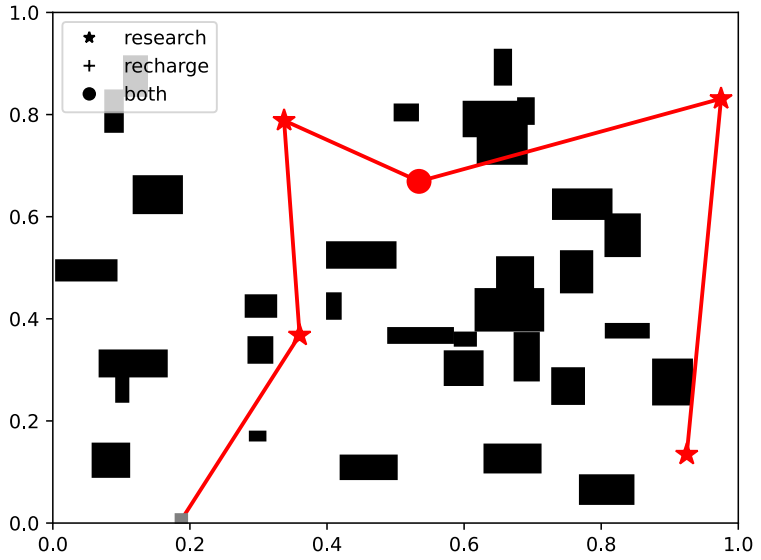

Figure 7. Optimal solution of an instance 5 PoIs.

a PoI, a blue circle is added. Each edge is labeled with associated τ and κ values. PoIs are labeled with the indices of visited vertices, as well as the battery level and time of arrival.

In Figure 7, the rover visited all PoIs and conducted research at each of them. Recharging occurred only once, at PoI number 5. In contrast, in Figure 8, PoI number 7 remains unvisited.

4. Summary

In this paper, we introduced a new problem, the **Rechargeable Rover Routing Problem (RRRP)**, which resembles classical vehicle routing problems, particularly in its variant with profits, but incorporates unique energy constraints. We developed a graph-based representation for the problem and proposed a mixed-integer non-linear programming (MINLP) formulation. To enhance solvability, we linearized the model, incorporating time discretization. The proposed model remains highly general, accommodating charging processes from solar panels and radioisotope thermoelectric generators (RTGs) while considering factors such as weather conditions and the time of the sol. Using small-scale examples, we demonstrated that the model behaves as expected and adjusts appropriately to varying parameters.

Although the proposed formulation results in a linear model, it necessitates auxiliary variables and additional constraints, which significantly increase its complexity. As a result, solving the model with MIP solvers becomes challenging for larger instances. While optimal

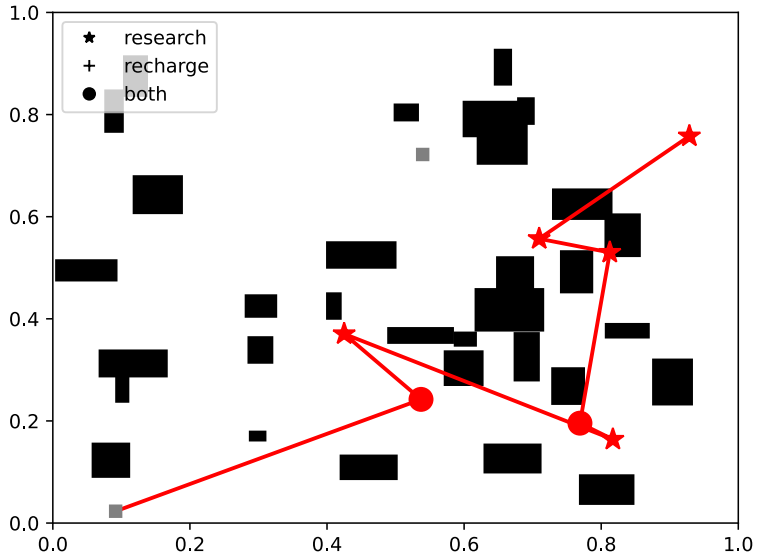

Figure 8. The best-found solution for an instance of 8 PoIs.

solutions were achieved for instances with up to eight Points of Interest (PoIs), for larger instances, the integrality gap after 20 minutes of computation ranged between 60% and 300%. This approach may suffice for Mars missions in the near future, but as the scope of missions expands to include more PoIs and rovers, more efficient heuristic or approximation algorithms will be necessary.

Future work should focus on developing such methods and expanding the problem formulation in several ways, including:

- **Vehicles**: Introducing a heterogeneous fleet, where each rover is equipped with a distinct set of tools, enabling it to service different subsets of PoIs with varying speeds.
- **Points of Interest (PoIs)**: Considering constraints such as time windows for servicing PoIs, as well as non-deterministic benefits, energy requirements, and task durations.
- **Environment**: Accounting for non-deterministic or dynamic travel times. For instance, even the best satellite imaging on Mars (e.g., HiRISE [19]) provides a resolution of only 1 m/pixel and covers only a small fraction of the Martian surface, leaving significant uncertainty in navigation.
- **Objective Function**: Expanding beyond single-objective optimization to incorporate multi-criteria objectives. For example, once a set of equally optimal solutions is obtained, a second optimization pass could minimize factors such as vehicle wear, travel time, risks, cost, or delays.

- **Realism of simulation**: Our work provides insights into the limitations of exact methods and serves as a baseline for the development of future inexact methods. However, both the exact model and any algorithms developed in the future should be validated within a more realistic simulation environment. This may include the use of Mars-specific data collected during past missions, as well as deployment in a physical testing environment. Such an environment—designed to replicate Martian surface conditions and incorporating physical rovers—is currently being developed as part of a parallel project at the Warsaw University of Technology. We therefore anticipate possibility for validation of our approach in such an environment in the near future.

The RRRP and its extensions represent a rich area for further research, with practical implications for the planning and execution of future Mars missions.

References

[1] Blazewicz J. and Kobler D. Review of properties of different precedence graphs for scheduling problems. *European Journal of Operational Research*, 142(3):435–443, 2002.

[2] Burzyński W. and Kaleta M. Task scheduling for autonomous vehicles in the martian environment. In Mańdziuk J., Żychowski A., and Małkiński M., editors, *Progress in Polish Artificial Intelligence Research 5. Proceedings of the 5th Polish Conference on Artificial Intelligence (PP-RAI'2024) 18–20.04.2024*, PP-RAI'2024, page 366–373, Warsaw, Poland, 2024. Politechnika Warszawska.

[3] Carsten J., Rankin A., Ferguson D., and Stentz A. Global path planning on board the mars exploration rovers. In *2007 IEEE Aerospace Conference*, pages 1–11, 2007.

[4] Chien S., Barrett A., Estlin T., and Rabideau G. A comparison of coordinated planning methods for cooperating rovers. In *Proceedings of the Fourth International Conference on Autonomous Agents*, AGENTS '00, page 100–101, New York, NY, USA, 2000. Association for Computing Machinery.

[5] Colby M., Yliniemi L., and Tumer K. Autonomous multiagent space exploration with high-level human feedback. *Journal of Aerospace Information Systems*, 13(8):301–315, 2016.

[6] Dobson A., Solovey K., Shome R., Halperin D., and Bekris K. Scalable asymptotically-optimal multi-robot motion planning. *Autonomous Robots*, 44, 03 2020.

[7] Elshaer R. and Awad H. A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. *Computers & Industrial Engineering*, 140:106242, 2020.

[8] Ghrist R., O'Kane J. M., and LaValle S. M. *Pareto Optimal Coordination on Roadmaps*, pages 171–186. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[9] Gunawan A., Kendall G., McCollum B., Seow H.-V., and and L. S. L. Vehicle routing: Review of benchmark datasets. *Journal of the Operational Research Society*, 72(8):1794–1807, 2021.

[10] Huang Y., Wu S., Mu Z., Long X., Chu S., and Zhao G. A multi-agent reinforcement learning method for swarm robots in space collaborative exploration. In *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*, pages 139–144, 2020.

[11] Huntsberger T., Pirjanian P., Trebi-Ollennu A., Das Nayar H., Aghazarian H., Ganino A., Garrett M., Joshi S., and Schenker P. Campout: a control architecture for tightly coupled coordination of multirobot systems for planetary surface exploration. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 33(5):550–559, 2003.

[12] Karaman S. and Frazzoli E. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30:846–894, 2011.

[13] Konstantakopoulos G. D., Gayialis S., and Kechagias E. Vehicle routing problem and related algorithms for logistics distribution: A literature review and classification. *Operational research*, 22(3):2033–2062, 2022.

[14] Kucukoglu I., Dewil R., and Cattrysse D. The electric vehicle routing problem and its variations: A literature review. *Computers & Industrial Engineering*, 161:107650, 2021.

[15] Le D. and Plaku E. Guiding sampling-based tree search for motion planning with dynamics via probabilistic roadmap abstractions. *IEEE International Conference on Intelligent Robots and Systems*, pages 212–217, 10 2014.

[16] Le D. and Plaku E. Cooperative, dynamics-based and abstraction-guided multi-robot motion planning. *J. Artif. Int. Res.*, 63(1):361–390, sep 2018.

[17] Lee C. An exact algorithm for the electric-vehicle routing problem with nonlinear charging time. *Journal of the Operational Research Society*, 72(7):1461–1485, 2021.

[18] Martins L. d. C., Tordecilla R. D., Castaneda J., Juan A. A., and Faulin J. Electric vehicle routing, arc routing, and team orienteering problems in sustainable transportation. *Energies*, 14(16), 2021.

[19] NASA. High resolution imaging experiment (hirise) website. https://mars.nasa.gov/mro/mission/instruments/hirise/. Acesssed: (2024-02-11).

[20] Nayak S., Paton M., and Otte M. W. A heuristic-guided dynamical multi-rover motion planning framework for planetary surface missions. *IEEE Robotics and Automation Letters*, 8(5):2542–2549, 2023.

[21] Nayak S., Yeotikar S., Carrillo E., Rudnick-Cohen E., Jaffar M. K. M., Patel R., Azarm S., Herrmann J. W., Xu H., and Otte M. Experimental comparison of decentralized task allocation algorithms under imperfect communication. *IEEE Robotics and Automation Letters*, 5(2):572–579, 2020.

[22] Petrovsky A., Kalinov I., Karpyshev P., Tsetserukou D., Ivanov A., and Golkar A. The two-wheeled robotic swarm concept for mars exploration. *Acta Astronautica*, 194:1–8, 2022.

[23] Sherwood R., Mishkin A. H., Chien S. A., Estlin T. A., Backes P. G., Cooper B. K., Rabideau G. R., and Engelhardt B. An integrated planning and scheduling prototype for automated mars rover command generation. 2001.

[24] Tahami H., Rabadi G., and Haouari M. Exact approaches for routing capacitated electric vehicles. *Transportation Research Part E: Logistics and Transportation Review*, 144:102126, 2020.

[25] Tompkins P., Stentz A., and Wettergreen D. Global path planning for mars rover exploration. In *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No.04TH8720)*, volume 2, pages 801–815 Vol.2, 2004.

[26] van den Berg J. and Overmars M. Prioritized motion planning for multiple robots. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 430–435, 2005.

[27] Švestka P. and Overmars M. H. Coordinated path planning for multiple robots. *Robotics and Autonomous Systems*, 23(3):125–152, 1998.

[28] Wagner G., Kang M., and Choset H. Probabilistic path planning for multiple robots with subdimensional expansion. In *2012 IEEE International Conference on Robotics and Automation*, pages 2886–2892, 2012.

[29] Yang Z., Zhao Q., Lv X., Wang L., and Liu W. Multiple robots autonomous task planning based on improved genetic algorithm. In *2024 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 547–552, 2024.

[30] Çağrı Koç, Bektaş T., Jabali O., and Laporte G. A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows. *Computers & Operations Research*, 64:11–27, 2015.

Received 03.12.2024, Accepted 17.06.2025