Placement of charging stations for energy-constrained robots in spider graphs*

Lucas Cardoso Pereira[1], **Santiago Valdés Ravelo**[1]

[1]Institute of Computing – State University of Campinas (UNICAMP)
Campinas, São Paulo, Brazil

l281817@dac.unicamp.br, ravelo@unicamp.br

Abstract. *In the* MIN-STATION *problem, we are given a simple graph $G = (V,E)$, a set of m robots initially positioned on vertices in $S \subseteq V$, and a set of target vertices $T \subseteq V$, with $|S| = |T| = m$, along with a positive integer r. The goal is to determine the minimum number of charging stations to place on the vertices so that each robot can move from a distinct vertex in S to a distinct vertex in T without running out of energy, given that each robot can traverse at most r edges between consecutive recharges. This work reviews existing results in the literature for paths, which admit an $O(|V|)$-time algorithm, and presents a more general linear-time solution for spider graphs, achieving $O(|V|)$ time complexity.*

1. Introduction

The strategic placement of charging stations is a critical challenge for energy-constrained robots [4, 3]. Das [2] formalized this as the MIN-STATION problem: given a simple graph $G = (V,E)$, starting $S \subseteq V$ and target $T \subseteq V$ positions ($|S| = |T|$), and a robot battery capacity r, the goal is to place a minimum set of charging stations $C \subseteq V$ ensuring all robots starting on each $s \in S$ reach a distinct target $t \in T$ without traversing more than r edges without visiting a charging station (energy constraint). MIN-STATION is NP-hard even for graphs with maximum degree 6, but admits exact polynomial algorithms for paths and cycles with $O(|V|)$ and $O(|V|^2)$ complexity, respectively [2]. In this paper, we extend these tractability boundaries by proposing an optimal $O(|V|)$ greedy algorithm for spider graphs (trees with a single vertex of degree ≥ 3).

2. Graph notation

This work considers simple undirected graphs. A graph $G = (V_G, E_G)$ is defined by a set V_G of elements, called vertices, and a set E_G of unordered pairs of vertices, called edges. Whenever the context is clear, we write V and E instead of V_G and E_G, respectively. If G contains an edge $e = uv$, we say that u and v are adjacent and incident to the edge e. The degree of a vertex u in G is the number of vertices adjacent to u, or equivalently, the number of edges incident to u.

A graph $H = (V_H, E_H)$ is a subgraph of G if $V_H \subseteq V_G$ and $E_H \subseteq E_G$. In this case, we write $H \subseteq G$. We denote by $G \setminus e$ the subgraph obtained from G by removing the edge e, and by $G \setminus u$ the subgraph obtained from G by removing the vertex u together with all edges incident to u.

*This work was supported by CNPq – National Council of Technological and Scientific Development of Brazil through the proccess 196667/2025-9.

A graph $G = (V,E)$ is called a *path* if its vertices can be ordered as $(v_1, v_2, \dots, v_n)$ such that $E = \{v_i v_{i+1} \mid i \in 1,2,\dots,n-1\}$. The *distance* between two vertices v_i and v_j on a path is $|i-j|$. An *interval* $[x,y]$ between two vertices x and y in a path is the subpath containing all vertices between them, including x and y.

A graph G is *connected* if, for every pair of vertices $u,w \in V$, there exists a path $(v_1, v_2, \dots, v_n)$ in G such that $v_1 = u$ and $v_n = w$. A *component* of G is a maximal connected subgraph of G.

A *cycle* is a graph with at least three vertices whose vertices can be ordered as $(v_1, v_2, \dots, v_n)$ such that $E = \{v_i v_{i+1} \mid i \in 1,2,\dots,n-1\} \cup \{v_1 v_n\}$. Note that a cycle can be seen as a path with an additional edge between the first and last vertices. A graph G is called *cyclic* if it contains a subgraph that is a cycle, otherwise, G is *acyclic*.

A *tree* is a connected acyclic graph. A *spider* is a tree with a single vertex c of degree at least 3, called the *center*, while all other vertices have degree at most 2. The components obtained from the graph $G \setminus c$ are paths, and are called the *radials* of the spider.

3. Review of the solution for paths

The greedy approach proposed by Das [2] for the MIN-STATION problem on paths relies on three fundamental structural properties. First, for robots traveling in the same direction, stations must be placed exactly at the limit of their battery capacity r to minimize the total amount of stations. Second, in an optimal solution, the paths of robots traveling in opposite directions never cross, as their targets can be swapped without increasing the cost. Finally, a perfect left-to-right matching between the robots' starting positions and their corresponding targets is always optimal. The lemmas proposed by Das are provided in the appendix.

These properties ensure that the MIN-STATION problem on paths can be solved using a single-pass greedy strategy. By scanning the path from left to right and maintaining two variables: a counter for the robots' routing demands and another for the distance traveled, stations are placed strictly when the maximum movement capacity r is reached. This approach yields an optimal solution within $O(|V|)$ time complexity [2].

4. Proposed Algorithm for Spiders

As established in [2], the MIN-STATION problem can be solved optimally on paths in $O(|V|)$ time using a greedy approach. In this section, we extend this result to spider graphs by exploiting their structure and proving the optimality of a radial greedy strategy.

Let $SP = (V,E)$ be a spider graph, defined as a tree with exactly one vertex $c \in V$ of degree $k \geq 3$, called the *center*, and all other vertices having degree at most 2. The graph $SP \setminus c$ consists of k disjoint paths (the radials), denoted by $R_1, R_2, \dots, R_k$. Each radial R_i contains exactly one vertex of degree 1 in SP, called a *leaf*, and exactly one vertex adjacent to c in SP.

We begin by analyzing the substructure of the problem along its radials.

Lemma 1. *Let R_i be a radial of a spider graph SP with center c. In an optimal solution to the* MIN-STATION *problem on SP, the placement of charging stations within R_i can* *be determined by applying the greedy path strategy from the leaf of R_i toward c, placing stations only when the maximum battery capacity r is reached.*

Demonstração. Since c is the only vertex connecting R_i to the rest of SP, any robot whose path starts in R_i and ends in $SP \setminus R_i$, or vice versa, must pass through c. For motion segments entirely contained within R_i, the topology is that of a path.

By Lemmas 1 and 2 in [2], for robots moving in a fixed direction on a path, the number of stations is minimized when stations are placed at the furthest feasible vertices (i.e., at distance r from the previous station or starting position). Thus, by processing R_i sequentially from its leaf toward c, we minimize the number of stations placed within R_i while maximizing the remaining battery capacity r' ($0 < r' \leq r$) of the robots upon reaching c.

Placing a station earlier along R_i would only decrease (or at best preserve) r' at c, potentially forcing an additional station to be placed in $SP \setminus R_i$, which contradicts the global optimality of the solution. □

After processing the radials, two situations may arise: (i) all robots have reached their targets, and nothing further needs to be done; or (ii) some radials contain robots that have not yet reached their targets (and, symmetrically, some radials contain unassigned target vertices). In the latter case, each such robot can reach the center c with a remaining battery capacity r'_i. We can therefore define a new greedy choice to handle the center, as follows.

Lemma 2. *Let L_R be the set of remaining robots and L_T the set of remaining targets. There exists an optimal solution in which the robot with the greatest remaining battery capacity is assigned to a target (or charging station) in the farthest radial among those containing remaining targets.*

Demonstração. Let r_a be a robot with maximum remaining battery capacity, and let t_f be a target located in a radial that is farthest from the center among those containing remaining targets.

Consider an optimal solution. If in the solution r_a is already assigned to a target in the farthest radial (in particular, to t_f), we are done. Otherwise, suppose r_a is assigned to some target t' in a closer radial, and let r_b be a robot assigned (in $\mathscr{S}$) to a target t_f in a farther radial.

Consider swapping the assignments of r_a and r_b: assign r_a to t_f and r_b to t'. Since r_a has maximum remaining battery capacity, it arrives at c with at least as much remaining capacity as r_b. Therefore, from c, r_a can reach t_f and r_b can reach t', where the latter requires no more distance than t_f. Hence, this swap does not increase the number of required charging stations. □

The above lemma shows that matching robots in non-increasing order of remaining battery capacity with targets in non-increasing order of distance from the center does not increase the number of required charging stations. Accordingly, our strategy is to perform such a matching and then determine whether an additional charging station is necessary. Observe that, if a charging station is required, it must be placed at the center c, since all robots in L_R can reach c, and from c every target or charging station associated with L_T is reachable. Therefore,

Theorem 1. *The* MIN-STATION *problem on a spider graph $SP = (V, E)$ can be solved optimally in $O(|V|)$ time.*

Demonstração. Let k be the number of radials of SP. The algorithm processes each radial R_i, for $1 \leq i \leq k$, independently from the leaf toward the center. By Lemma 1, the greedy placement of stations within each R_i is optimal and ensures that the robots reach c with the maximum possible remaining battery capacity. Since each radial R_i is a path, processing R_i takes time proportional to its length, namely $O(|V_{R_i}|)$ [2]. Thus, processing all radials takes $\sum_{i=1}^{k} |O(V_{R_i})| = O(|V|)$.

After processing all k radials and reaching c, the algorithm evaluates the aggregated state at the center. It constructs the sets L_R of remaining robots and L_T of remaining targets, and assigns robots to targets according to Lemma 2. This assignment is computed by sorting the elements of L_R and L_T. Since each value is an integer in the range $[0, |V| - 1]$ (corresponding to distances in the tree), counting sort can be applied to each set in $O(|V|)$ time [1]. Therefore, this step also runs in $O(|V|)$ time.

At this point, there are only two possible conditions under which no charging station is required at the center c. First, after the greedy matching between robots and targets, every robot can directly reach its matched target with its remaining battery capacity. Second, a robot may first move to some charging station already placed in a radial and, from there, continue to its target. This condition holds if each robot can reach the closest charging station to c in some radial and all targets are reachable from such station. Both conditions can be verified through the sorting procedure. Therefore, if neither condition holds, any feasible solution must place a charging station at c.

For each matched pair in $L_R \times L_T$, these feasibility checks take $O(1)$ time. Since there are at most $|L_R| = O(|V|)$ pairs, the total time spent in this phase is $O(|V|)$.

Combining all steps, the overall time complexity of the algorithm is $O(|V|)$. □

5. Conclusion and Future Work

In this paper, we addressed the MIN-STATION problem, which seeks to determine the minimum number of charging stations required to guarantee the successful movement of energy-constrained robots. Building on the structural properties and the linear-time solution for simple paths [2], we proposed an $O(|V|)$ greedy algorithm for spider graphs. By formalizing the optimal substructure along the radials of the spider and establishing the precise condition for station placement at the center, we showed that postponing the placement of charging stations from the leaves toward the center yields a globally optimal solution.

As future work, we plan to extend this combinatorial analysis to more complex tree-like structures, such as caterpillars, lobsters and general trees, as well as to planar and cubic graphs. Additionally, formulating the MIN-STATION problem as an Integer Linear Program (ILP) may provide a robust approach for obtaining exact solutions or high-quality approximations on general graphs, helping to address the inherent NP-hardness of the problem.

Referências

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2nd edition, 2001.

[2] A. K. Das. *Charging Station Placement for Limited Energy Robots*, pages 97–108. Springer Nature Switzerland, 2025.

[3] T. Kundu and I. Saha. Approximation algorithms for charging station placement for mobile robots. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4770–4776. IEEE, 2023.

[4] G. P. Strimel and M. M. Veloso. Coverage planning with finite resources. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2950–2956. IEEE, 2014.