Parking Problem by Oblivious Mobile Robots in Infinite Grids

Abhinav Chakraborty[1(✉)] and Krishnendu Mukhopadhyaya[2]

[1] Department of Computer Science and Engineering, Institute of Technical Education and Research, Siksha O Anusandhan University, Bhubaneswar, Odisha, India
abhinav.chakraborty06@gmail.com

[2] Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata, India

Abstract. In this paper, the parking problem of a swarm of mobile robots has been studied. The robots are deployed at the nodes of an infinite grid, which has a subset of prefixed nodes marked as *parking nodes*. Each parking node p_i has a capacity of k_i which is given as input and equals the maximum number of robots a parking node can accommodate. As a solution to the parking problem, robots need to partition themselves into groups so that each parking node contains a number of robots that are equal to the capacity of the node in the final configuration. It is assumed that the number of robots in the initial configuration represents the sum of the capacities of the parking nodes. The robots are assumed to be autonomous, anonymous, homogeneous, identical and oblivious. They operate under an asynchronous scheduler. They neither have any agreement on the coordinate axes nor do they agree on a common chirality. All the initial configurations for which the problem is unsolvable have been identified. A deterministic distributed algorithm has been proposed for the remaining configurations, ensuring the solvability of the problem.

Keywords: Distributed Computing · Mobile Robots · Look-Compute-Move Cycle · Asynchronous · Infinite Grid · Parking Nodes

1 Introduction

Robot swarms are groups of generic mobile robots that can collaboratively execute complex tasks. Such systems of mobile robots are assumed to be simple and inexpensive and offer several advantages over traditional single-robot systems, such as scalability, robustness and versatility. A series of research on the algorithmic aspects of distributed coordination of robot swarms has been reported in the field of distributed computing (see [12] for a comprehensive survey). In the traditional framework of swarm robotics, the robots are assumed to be anonymous (no unique identifiers), autonomous (there is no centralized control), identical (no unique identifiers), homogeneous (each robot executes the same deterministic

S. Devismes et al. (Eds.): ICDCIT 2024, LNCS 14501, pp. 68–84, 2024.
https://doi.org/10.1007/978-3-031-50583-6_5

distributed algorithm) and oblivious (no memory of past information) computational entities. The robots are represented as points in the Euclidean plane. They do not have access to any global coordinate system. However, each robot has its own local coordinate system, with the origin representing the current position of the robot. The robots do not have an explicit means of communication, i.e., they are assumed to be silent. They are disoriented, i.e., they neither agree on a common coordinate axes nor do they have any agreement on chirality. Each robot is equipped with visibility sensors, by which they can perceive the deployment region.

In this paper, the deployment region of the robots is assumed to be an infinite square grid, which represents a natural discretization of the plane. The robots are deployed at the nodes of the input grid graph. The graph also consists of some prefixed grid nodes, designated as parking nodes. When a robot becomes active, it operates according to the *Look-Compute-Move* cycle. A robot takes a snapshot of the entire graph, including the positions of the other robots and parking nodes in the *Look* phase. Based on the snapshot, it computes a destination node in the *Compute* phase according to a deterministic algorithm, where the destination node might be its current position as well. Finally, it moves towards the destination in the *Move* phase. In this paper, we have considered the most general model, which is the asynchronous model ($\mathcal{ASYNC}$). In this setting, there is no common notion of time, and all the robots are activated independently. Each of the Look, Compute and Move phases has a finite but unpredictable duration. In the initial configuration, it has been assumed that the robots are placed at the distinct nodes of the grid graph. During the look phase, the robots can perceive the parking nodes using their visibility sensors. Each parking node has a capacity, which is subjected to a constraint that it can accommodate a maximum number of robots equal to its capacity. The capacity of a parking node is given as an input to each robot. For simplicity, we have assumed that the number of robots in the initial configuration is equal to the sum of the capacities of the parking nodes. In this paper, we have assumed that the robots have *global-strong multiplicity detection capability*. This means the robots are able to determine the exact number of robots that make up the multiplicity in each node. It has been proved later that the parking problem is unsolvable if the robots do not have such capabilities.

1.1 Motivation

The fundamental motivation behind studying the parking problem is twofold. Firstly, the parking problem can be viewed as a special case of the partitioning problem [11], which requires the robots to divide themselves into m groups, each consisting of k robots while converging into a small area. Unlike the partitioning problem, the parking problem requires that each parking node must contain robots exactly equal to its given capacity in the final configuration. However, the capacities of the parking nodes may be different. Moreover, if the capacities of each of the parking nodes are assumed to be k, i.e., they are equal in the initial configuration; the problem is reduced to the k-epf problem [3], which is a generalized version of the embedded pattern formation problem, where each fixed point contains exactly k robots in the final configuration. Secondly, in the traditional models, the robots are assumed to be points that can move freely on the plane. The robots are assumed to move with high accuracy and by infinitesimal distance in the continuous domain. Even if the area of robot deployment is small, a dimensionless robot can move without causing any collision. In practice, it may not always be possible to perform such infinitesimal movements with infinite precision. However, in our paper, the robots are deployed at the nodes of an infinite grid. The movements of the robots are restricted along the grid lines, and a robot can move toward one of its neighbors at any instant of time. The restrictions imposed by the grid model on the movements of the robots make it challenging to design collision-less algorithms, as opposed to the movement of the robots in a continuous environment. In addition to the theoretical benefits, the parking nodes can also be seen as base stations or charging stations with some allowable capacities.

1.2 Related Works

Most of the theoretical studies on swarm robotics have been concentrated on *arbitrary formation problem* and *gathering* under different settings. The Arbitrary Pattern Formation or $\mathcal{APF}$ is a fundamental coordination problem in Swarm Robotics, where the robots are required to form any specific but arbitrary geometric pattern given as input. The study of $\mathcal{APF}$ was initiated in [16]. The authors characterized the class of formable patterns by using the notion of symmetricity, which is essentially the order of the cyclic group that acts on the initial configuration. The $\mathcal{APF}$ was first studied in the $\mathcal{ASYNC}$ by Flocchini et al. [13], where the robots are assumed to be oblivious. While all the previous studies considered the problem with unlimited visibility, Yamauchi et al. [17] studied the problem where the robots have limited visibility. Cicerone et al. [7] studied the $\mathcal{APF}$ problem without assuming common chirality among the robots. Bose et al. [4] were the first to study the problem in a grid-based terrain. D'Angelo et al. [9], studied the gathering problem on finite grids. Stefano et al. studied the optimal gathering problem in infinite grids [15]. In this paper, they proposed an optimal deterministic algorithm that minimizes the total distance traveled by all the robots. The concept of fixed points was first introduced by Fujinaga et al. [14] on the Euclidean plane. In this paper, the *landmark* covering problem was studied. The problem requires that each robot must attain a configuration where all the robots must occupy a single fixed point or landmark. They propose an algorithm based on the assumption that the robots agree on a common chirality. The proposed algorithm minimizes the total distance traveled by all the robots. In [8], Cicerone et al. studied the *embedded pattern formation* problem without assuming any common chirality among the robots. The problem necessitates a distributed algorithm in which each robot must occupy a unique fixed point within a finite amount of time. The *k-circle formation problem* [3,10] has been studied in the setting where the robots agree on the directions and orientations of the Y- axis and on the disoriented setting. Given a positive integer

k, the *k-circle formation problem* asks a swarm of mobile robots to form disjoint circles. Each of these circles must be centered at one of the pre-fixed points on the plane. Each circle must contain a total of k robots at distinct locations on the circumference of the circles. Bhagat et al. [3] also studied the *k- epf problem* in the continuous domain, which is a generalized version of the embedded pattern formation problem. This problem necessitates the arrival and retention of exactly k robots at each fixed point. Cicerone et al. [6] studied a variant of the gathering problem, where each robot must gather at one of the prefixed meeting points. The problem was defined as *gathering on meeting points* problem. The authors proposed a deterministic algorithm that minimizes the total distance traveled by all the robots and minimizes the maximum distance traveled by a single robot. *Gathering over meeting nodes* problem was studied by Bhagat et al. [1,2]. In this problem, the robots are deployed on the nodes of an infinite square grid, which has a subset of nodes marked as meeting nodes. Each robot must gather at one of the prefixed meeting nodes within a finite amount of time.

1.3 Our Contribution

This paper considers the parking problem over an infinite grid. The robots are deployed at the nodes of an infinite grid, which also consists of some prefixed parking nodes. Each parking node p_i has a capacity k_i, which is the maximum number of robots it can accommodate at any moment of time. We assume that the number of robots n is equal to $\sum_{i=1}^{m} k_i$, where m is the total number of parking nodes. We have characterized all the initial configurations and the values of k_i for which the problem is unsolvable. For the remaining configurations, a deterministic algorithm has been proposed that ensures the solvability of the problem.

2 Models and Definitions

2.1 Models

The robots are assumed to be dimensionless, anonymous, autonomous, identical, homogeneous and oblivious. The robots are assumed to be disoriented, i.e., they neither have any agreement on the coordinate axes nor have any agreement on a common chirality. They do not have an explicit means of communication, i.e., they are assumed to be silent. Let $P = (\mathbb{Z},\, E')$ denote the infinite path graph with the vertex set V corresponding to the set of integers $\mathbb{Z}$ and the edge set is denoted by the ordered pair $E' = \{(i,\, i+1) | i \in \mathbb{Z}\}$. Let $\mathcal{R} = \{r_1, r_2 \ldots r_n\}$ denote the set of robots that are deployed at the nodes of G, where G is the input infinite grid graph defined as the usual *Cartesian Product* of the graph $P \times P$. Let $r_i(t)$ denote the node occupied by the robot $r_i \in \mathcal{R}$ at time t. Assume that $\mathcal{R}(t)$ denotes the set of all such distinct nodes occupied by the robots in $\mathcal{R}$ at time t. Since the robots are deployed at the nodes of an infinite square grid, they have an agreement on a common measure of unit distance.

The input grid graph also comprises some prefixed nodes designated as *parking nodes*. Let $\mathcal{P} = \{p_1, p_2, ..., p_m\}$ denote the set of parking positions. In the initial configuration, the parking nodes are located at the distinct nodes of the grid. A robot may be deployed at one of the parking nodes in the initial configuration. The movements of the robots are restricted along the grid lines. At any instant of time, a robot can move only to one of its four neighboring nodes. The movement of the robot is assumed to be instantaneous, i.e., the robot can be observed only at the nodes of the graph and not on the edges. In other words, no robot can be seen while moving. A robot's vision is assumed to be global, meaning that each robot is equipped with visibility sensors that allow it to observe the whole grid graph.

2.2 Terminologies and Definitions

- **Distance between two nodes:** Let $d(u, v)$ denote the distance between two nodes u and v.
- **Capacity of a parking node:** The *capacity* of a parking node given as an input is defined as the maximum number of robots the parking node can accommodate. A parking node is said to be *saturated* if it contains exactly the number of robots equal to its capacity. A parking node is said to be *unsaturated* if it is not saturated. Let $\mu : V \rightarrow \mathbb{N} \cup \{0\}$ be defined as a function, where:

$$\mu(v) = \begin{cases} 0 & \text{if } v \text{ is not a parking node} \\ capacity\ of\ the\ parking\ node & \text{otherwise} \end{cases}$$

  In the initial configuration, let k_i be the capacity of a parking node p_i, $\forall i = 1, 2, \ldots, m$.
- **Symmetry of a configuration $\boldsymbol{C(t)}$:** Two graphs $G_1 = (V_{G_1}, E_{G_1})$ and $G_2 = (V_{G_2}, E_{G_2})$ are said to be isomorphic if there exists a bijection $\phi : V_{G_1} \rightarrow V_{G_2}$ such that any two nodes $u, v \in V_{G_1}$ are adjacent in G_1 if and only if $\phi(u)$, $\phi(v) \in V_{G_2}$ are adjacent in G_2. An automorphism on a graph G is a permutation of its nodes mapping edges to edges and non-edges to non-edges. Let λ_t be defined as a function that denotes the number of robots residing on v at time t. Without any ambiguity, we denote the function λ_t by λ. $C(t) = (\mathcal{R}(t), \mathcal{P}, \lambda, \mu)$ denotes the *system configuration* at any time t. An automorphism of a graph can be extended to the automorphism of a configuration. Two configurations are said to be isomorphic if there exists an *automorphism* ϕ of the input grid graph such that $\lambda(v) = \lambda(\phi(v))$ and $\mu(v) = \mu(\phi(v))$, for all $v \in V$. The set of all automorphisms of a configuration forms a group which is denoted by $Aut(C(t),\ \lambda, \mu)$. If $|Aut(C(t),\ \lambda, \mu)| = 1$, then the configuration is asymmetric. Otherwise, the configuration is said to be symmetric. We assume that the infinite grid is embedded in the *Cartesian plane*. As a result, a grid can admit only three types of automorphism and combinations of them, *translation*: defined by the shifting of the nodes to the same extent, *reflection*: defined by the line of reflection axes and *rotation*:

defined by the angle of rotation and the center of rotation. The reflection axis can be horizontal, vertical or diagonal. It can either pass through the nodes or edges of the grid. If a configuration admits rotational symmetry, then the center of rotation can be either a node, the center of an edge or the center of the area surrounded by four nodes. The angle of rotation can be either 90° or 180°. Since the number of occupied nodes is finite, a translation symmetry is not admissible. Let MER be the *minimum enclosing grid* containing all the occupied nodes of $C(t)$. Assume that the dimension of MER is $a \times b$. The number of grid edges on a side of MER is used to define its length.

– **View:** Starting from a corner of MER, scan the entire grid in a direction parallel to the width of the rectangle. While scanning the grid, we associate the pair $(\lambda(v), \mu(v))$ to each node v that the string encounters. Similarly, we can define the string associated with the same corner and encounter the nodes of the grid in the direction parallel to the length of the grid. Consider the eight senary strings of length ab that are associated with the corners of MER, with two senary strings defined for each corner of MER. Let the two strings defined for a corner i be denoted by s_{ij} and s_{ik}.

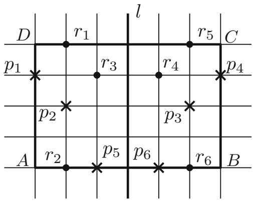

Fig. 1. The configuration is symmetric with respect to l. The crosses represent parking nodes and the black circles represent robot positions

If MER is a non-square rectangle, we can distinguish between the two strings associated with a given corner by looking at the string that runs parallel to the side with the shortest length. Consider any particular corner i of MER. Assume that $|ij| < |ik|$. We consider the direction parallel to ij as the *string direction* associated to i. We define $s_i = s_{ij}$ as the *string representation* associated to the corner i. The direction parallel to the larger side is defined as the *non-string direction* associated to the corner i. In the case of a square grid, between the two strings associated to a corner, the *string representation* is defined as the larger lexicographic string, i.e., $s_i = max(s_{ij}, s_{ik})$, where the maximum is defined according to the lexicographic ordering of the strings. If the configuration is asymmetric, we will always get a unique largest lexicographic string. Without loss of generality, let s_i be the largest lexicographic string among all the strings associated to the corners. Then we refer to i as the *key corner*. If the configuration is asymmetric, the robots can be ordered according to the key corner and the string direction. A *non-key corner* is defined as one that is not a key corner. In Fig. 1, assume that the capacity of each parking node is 1. The lexicographic string associated with the corners C and D are $s_{CB} = s_{DA} =$ ((0,0), (0,1), (0,0), (0,0), (0,0), (1,0), (0,0), (0,1), (0,0), (1,0), (0,0), (1,0), (0,0), (0,0), (0,1), (0,0), (0,0), (0,0), (0,0), (0,0), (0,0), (1,0), (0,0), (0,0), (0,1), (1,0), (0,0), (0,1), (0,0), (1,0), (0,0), (0,1), (0,0), (0,0), (0,0)). The strings $s_{CB} = s_{DA}$ are the maximum lexicographic strings associated and hence C and D are the key corners. The *configuration view* of a node is defined as the tuple (d', x), where d' denotes the distance of a node from the key corner in the string direction and x denotes the type of the node, i.e., x is either an empty node, parking node or a robot position.

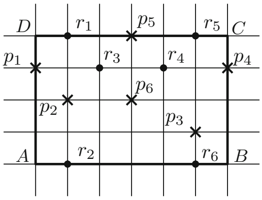

Fig. 2. Figure highlighting the definition of leading corner.

– **Symmetricity of the set $\mathcal{P}$**: We may define the symmetry of the set $\mathcal{P}$ in the same way as we define the symmetry of a configuration. The smallest grid-aligned rectangle that includes all the parking nodes is denoted as $M_{\mathcal{P}}$. We can define a string α_i similar to s_i. The only difference is that each node v is associated with $\mu(v)$ instead of the pair $(\lambda(v), \mu(v))$. If the parking nodes are asymmetric, a unique lexicographic largest string α_i always exists. If the parking nodes are not asymmetric, then the parking nodes are said to be symmetric. The corner with which the lexicographic largest string α_i is associated is defined as the *leading corner*. In Fig. 2, assume that the capacity of $p_1 = p_2 = p_3 = 3$ and $p_4 = p_5 = p_6 = 2$, $\alpha_{DA} = 03000003000000020200000000003002000$ is the largest lexicographic string among the $\alpha_i' s$ and hence we have D as the leading corner. According to this definition of symmetricity of the set $\mathcal{P}$, the parking nodes that are located in the symmetric positions must have equal capacities.

Definition 1. *Let $C(0)$ be any given initial configuration. A parking node p_i is said to have a higher order than the parking node p_j if it appears after p_j in the string representation α_k, associated to some leading corner k of MER. Similarly, a robot r_i has a higher order or has a higher configuration view than r_j if it appears after r_j in the string representation s_k, associated to some key corner k of MER.*

3 Problem Definition and Impossibility Results

3.1 Problem Definition

Let $C(t) = (\mathcal{R}(t), \mathcal{P}, \lambda, \mu)$ denote the system configuration at any time t. Each parking node p_i has a capacity k_i. For each parking node p_i, the capacity k_i is given as an input. The number of robots is assumed to be equal to $\sum_{i=1}^{m} k_i$, where m is the total number of parking nodes. In an initial configuration, all the robots occupy distinct nodes of the grid. The goal of the parking problem is to transform any initial configuration at some time $t > 0$ into a configuration such that each parking node p_i is saturated, i.e., p_i contains exactly k_i robots on it and any robot taking a snapshot in the look phase at time t will decide not to move.

3.2 Partitioning of the Initial Configurations

All the initial configurations can be partitioned into the following disjoint classes.

1. $\mathcal{I}_1$: The parking nodes are asymmetric (Fig. 2).
2. $\mathcal{I}_2$: The parking nodes are symmetric with respect to a unique line of symmetry l. This class of configurations can be further partitioned into:
   $\mathcal{I}_{21}$: $C(t)$ is asymmetric (In Fig. 3(a), if the capacity of each parking node is the same, i.e., 1, the configuration is asymmetric with the parking nodes symmetric with respect to l).

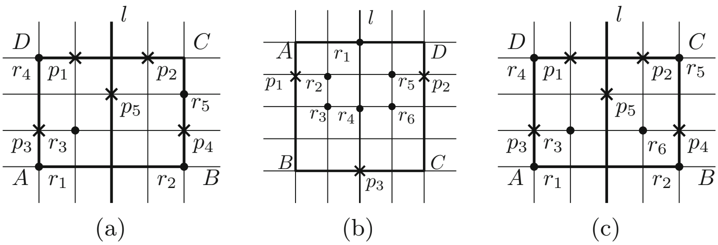

Fig. 3. Examples of $\mathcal{I}_{21}$, $\mathcal{I}_{221}$ and $\mathcal{I}_{223}$ configuration

   $\mathcal{I}_{22}$: $C(t)$ is symmetric with respect to l. This can be further partitioned into the following disjoint classes: (1) $\mathcal{I}_{221}$: There exists at least one robot position on l (In Fig. 3(b), with the assumption that the capacity of each parking node is 2, $C(t)$ is symmetric with respect to l with robots r_1 and r_4 on l). (2) $\mathcal{I}_{222}$: There does not exist any robot position on l. Also, there are no parking nodes on l. (3) $\mathcal{I}_{223}$: There does not exist any robot position on l, but there exists at least one parking node on l (In Fig. 3(c), if the capacity of each parking node not on l is 1, $C(t)$ is symmetric with respect to l and there exists parking node p_5 at l with capacity 2).
3. $\mathcal{I}_3$: The parking nodes are symmetric with respect to rotational symmetry, with c as the center of rotational symmetry. This class of configurations can be further partitioned into:
   $\mathcal{I}_{31}$: $C(t)$ is asymmetric (In Fig. 4(a) if the capacity of each parking node is 2, $C(t)$ is asymmetric with the parking nodes being symmetric with respect to rotational symmetry).

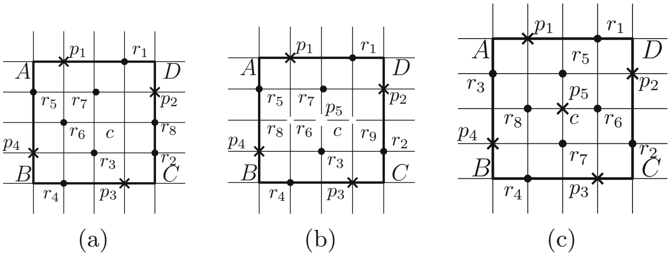

Fig. 4. Examples of $\mathcal{I}_{31}$ configuration, $\mathcal{I}_{321}$ configuration and $\mathcal{I}_{323}$ configuration.

$\mathcal{I}_{32}$: $C(t)$ is symmetric with respect to c. This can be further partitioned into the following disjoint classes: (1) $\mathcal{I}_{321}$: There exists a robot position on c (In Fig. 4(b), if the capacities of parking nodes p_1, p_2, p_3 and p_4 equal 1 and the capacity of the parking node p_5 equals 5, $C(t)$ is symmetric with respect to rotational symmetry. The robot r_6 is at the parking node p_5). (2) $\mathcal{I}_{322}$: There does not exist a robot position or parking node on c. (3) $\mathcal{I}_{323}$: There exists a parking node on c, but no robot on c (In Fig. 4(c) if the capacities of the parking nodes p_1, p_2, p_3 and p_4 equal 1 and the capacity of the parking node p_5 equals 4, $C(t)$ is symmetric with respect to rotational symmetry, with a parking node p_5 on c).

In the remainder of the paper, we assume that l is the line of symmetry if the parking nodes admit a single line of symmetry. If the parking nodes admit rotational symmetry, then c is the center of rotational symmetry. We also assume that if the parking nodes admit rotational symmetry, then l and l' are perpendicular lines passing through c, which divide the grid into four quadrants.

3.3 Impossibility Results

Lemma 1. *Let $\mathcal{A}$ be any algorithm for the parking problem in infinite grids. If there exists an execution of $\mathcal{A}$ such that the configuration $C(t)$ contains a robot multiplicity at a node that is not a parking node, then $\mathcal{A}$ cannot solve the parking problem.*

This lemma ensures that during the execution of any algorithm that solves the parking problem, the robots must perform a collision-less movement at all stages of the algorithm. Suppose the robots are oblivious and not endowed with global-strong multiplicity detection capability. In that case, they cannot detect whether exactly the k_i number of robots reaches the parking node p_i. We formalize the result in the following lemma:

Lemma 2. *Without the global-strong multiplicity detection capability of the robots, the parking problem is unsolvable.*

Lemma 3. *If the initial configuration $C(0) \in \mathcal{I}_{223}$ is such that the capacity of a parking node on l is an odd integer. Then the parking problem is unsolvable.*

It follows from Lemma 3 that if $C(t)$ admits multiple lines of symmetry and if there exists a parking node on l with odd capacity, then also the problem is unsolvable.[1]

Corollary 1. *If the initial configuration* $C(0) \in \mathcal{I}_{323}$*, then the parking problem is unsolvable if the capacity of the parking node at c is neither a multiple of 4 nor 2, depending on whether the angle of rotation is either* $90°$ *or* $180°$.

Let $\mathcal{U}$ be the set of all configurations that are unsolvable according to Lemma 3 and Corollary 1.

4 Algorithm

In this section, the parking problem is solved using a deterministic distributed algorithm *Parking ()* for all initial configurations except those belonging to $\mathcal{U}$. The fundamental strategy of the proposed algorithm is to identify a specific *target parking node* and permit a number of robots to move towards it, where the number of robots is equal to the parking node's capacity. The target parking node is selected in a sequential manner and the procedure executes unless each parking node becomes saturated. The proposed algorithm mainly consists of the following phases: *Guard Selection and Placement (GS)* phase, *Target Parking Node Selection (TPS)* phase, *Candidate Robot Selection (CR)* phase and *Guard Movement (GM)* phase.

Note that according to the definition of the symmetry of the set $\mathcal{P}$, there exists a unique lexicographic string α_i, when the parking nodes are asymmetric. From this, we can observe that if the parking nodes are asymmetric, the parking nodes can be ordered (say $\mathcal{O}_1$). Similarly, if the parking nodes are symmetric with respect to l, with at least one parking node on l, then the parking nodes on l are orderable (say $\mathcal{O}_2$). These orderings are necessary to identify a unique parking node, which will be selected by the robots in order to initialize the parking formation.

4.1 Guard Selection and Placement (*GS*)

Consider the case when the parking nodes are symmetric, but the configuration is asymmetric. In this phase, a unique robot is selected as a guard and placed in such a way that the configuration remains asymmetric during the execution of the algorithm. The following notations are used in describing this phase:

- Condition C_1: There exists at least one robot position outside the rectangle $M_{\mathcal{P}}$.
- Condition C_2: Each robot is inside the rectangle $M_{\mathcal{P}}$.
- Condition C_3: There exists a unique farthest robot from $l \cup \{c\}$.

[1] The proofs of the Lemmas 1 and 3 are in the arxiv version of the paper [5].

Depending on the class of configurations to which $C(t)$ belongs, the phase is described in Table 1. If there is more than one furthest robot from the key corner, then since the configuration is asymmetric, a unique robot can always be selected according to the view of the robots. Note that while the guard is selected and placed, the guard is the unique farthest robot from $l \cup \{c\}$. As a result, it does not have any symmetric image with respect to $l \cup \{c\}$, which implies that the configuration remains asymmetric during the execution of the algorithm.

Table 1. Guard Selection and Placement

Guard Selection and Placement		
Initial Configuration ($\mathcal{I}_{21} \cup \mathcal{I}_{31}$)	Guard	Position of the guard
$C_1 \wedge C_3$	The unique robot farthest from $l \cup \{c\}$	Current position of the guard
$C_1 \wedge \neg C_3$	The unique robot furthest from $l \cup \{c\}$ and having the maximum configuration view among all the furthest robots	The unique robot moves towards an adjacent node away from $l \cup \{c\}$
$C_2 \wedge C_3$	The unique robot furthest from $l \cup \{c\}$	The guard continues its movement away from $l \cup \{c\}$, unless the condition C_1 becomes true
$C_2 \wedge \neg C_3$	The unique robot furthest from $l \cup \{c\}$ and having the maximum configuration view among all the furthest robots	The guard continues its movement away from $l \cup \{c\}$ until the condition C_1 becomes true

4.2 Half-Planes and Quadrants

First, consider the case when $C(0) \in \mathcal{I}_{21}$. The line of symmetry l divides the entire grid into two half-planes. We consider the open half-planes, i.e., the half-planes excluding the nodes on l. Let H_1 and H_2 denote the two half-planes delimited by l. The following definitions are to be considered.

1. $UP(t)$- Number of parking nodes which are unsaturated at time t.
2. *Deficit Measure of a parking node p_i ($Df_{p_i}(t)$):* The deficit measure $Df_{p_i}(t)$ of a parking node p_i at time t is defined as the deficit in the number of robots needed to have exactly k_i robots on p_i.
3. $K_1 = \sum_{p_i \in H_1} Df_{p_i}(t)$ denotes the total deficit in order to have exactly $\sum_{p_i \in H_1} k_i$ number of robots at the parking nodes belonging to the half-plane H_1.
4. $K_2 = \sum_{p_i \in H_2} Df_{p_i}(t)$ denotes the total deficit in order to have exactly $\sum_{p_i \in H_2} k_i$ number of robots at the parking nodes belonging to the half-plane H_2.

Definition 2. *Let $C(t)$ be any initial configuration belonging to the set $\mathcal{I}_{21}$. $C(t)$ is said to be unbalanced if the two half-planes delimited by l contain an unequal number of robots. Otherwise, the configuration is said to be balanced.*

We next consider the following conditions.

1. Condition C_4- There exists a unique half-plane that contains the minimum number of unsaturated parking nodes.
2. Condition C_5- $K_1 \neq K_2$
3. Condition C_6- The configuration is unbalanced.
4. Condition C_7- The configuration is balanced and $\mathcal{R} \cap l \neq \emptyset$.
5. Condition C_8- The configuration is balanced and $\mathcal{R} \cap l = \emptyset$.

The half-plane $\mathcal{H}_{target}$ or $\mathcal{H}^+$ is defined according to Table 2, where the parking at the parking nodes initializes. The other half-plane is denoted by $\mathcal{H}^-$. In Fig. 5 (a), $ABCD$ is the $M_{\mathcal{P}}$ and $AB'C'D$ is the MER. Assume that the capacities of the parking nodes p_1, p_2, p_3 and p_4 are 2, 2, 1 and 1, respectively. The half-plane with more number of robots is selected as $\mathcal{H}^+$. In Fig. 5 (b), assume that the capacities of the parking nodes p_1, p_2, p_3 and p_4 are 3, 3, 2 and 2, respectively. Each of the half-planes contains the same number of robots. Therefore, the configuration is balanced. The half-plane not containing the guard r_5 is defined as $\mathcal{H}^+$. Due to space constraints, the case when the parking nodes are symmetric with respect to rotational symmetry has been included in the arxiv version of the paper [5]. The quadrant $\mathcal{Q}_{target}$ or $\mathcal{Q}^{++}$, where the parking is initialized, is defined according to Table 3 in the arxiv version of the paper [5].

Table 2. Demarcation of the half-planes

Demarcation of the half-planes for fixing the target	
Initial Configuration ($\boldsymbol{\mathcal{I}_{21}}$)	$\boldsymbol{\mathcal{H}^+}$
C_4	The unique half-plane which contains the minimum number of unsaturated parked nodes
$\neg C_4 \wedge C_5 \wedge K_1 < K_2$	H_1
$\neg C_4 \wedge C_5 \wedge K_2 < K_1$	H_2
$\neg C_4 \wedge \neg C_5 \wedge C_6$	The unique half-plane with the maximum number of robot positions
$\neg C_4 \wedge \neg C_5 \wedge \neg C_6 \wedge C_7$	The northernmost robot on l move towards an adjacent node away from l. The unique half-plane with the maximum number of robot positions is defined as $\mathcal{H}^+$
$\neg C_4 \wedge \neg C_5 \wedge \neg C_6 \wedge \neg C_7 \wedge C_8$	The unique half-plane not containing the guard

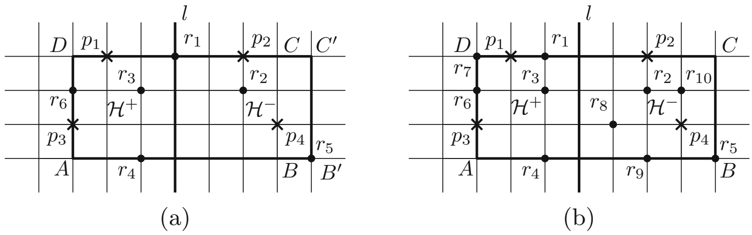

Fig. 5. Example configuration showing demarcations of half-planes.

4.3 Target Parking Node Selection (TPS)

In this phase, the target parking node for the parking problem is selected. Depending on the following classes of configurations, the phase is described in Table 3. Let p_{guard} be the closest parking node from the guard. If multiple such parking nodes exist, the parking node closest to the guard and having maximum order is selected as p_{guard}. We first assume that the target parking nodes are selected in $\mathcal{P} \setminus \{p_{guard}\}$. Due to space constraints, we have discussed the TPS phase in the arxiv version of the paper for the case when the parking nodes admit rotational symmetry [5]. We next consider the following conditions that are relevant in understanding this phase.

1. C_{14}- There exists an unsaturated parking node on l.
2. C_{16}- All the parking nodes belonging to $\mathcal{H}^+$ are saturated.
3. C'_{16}- All the parking nodes belonging to $\mathcal{H}^-$ are saturated.

While all the parking nodes belonging to the set $\mathcal{P} \setminus \{p_{guard}\}$ become saturated, p_{guard} becomes the target parking node. Note that $\neg C_{14}$ implies that the parking

Table 3. Target Parking Node Selection

Target Parking Node Selection	
Initial Configuration $\boldsymbol{C(0)}$	Target Parking Node
$\mathcal{I}_1$	The parking node which is unsaturated and has the highest order with respect to $\mathcal{O}_1$
$\mathcal{I}_2 \wedge C_{14}$	The parking node on l which is unsaturated and has the highest order with respect to $\mathcal{O}_2$
$\mathcal{I}_{21} \wedge \neg\, C_{14} \wedge \neg\, C_{16}$	The parking node, which is unsaturated and has the highest order in $\mathcal{H}^+$ among all the unsaturated nodes in $\mathcal{H}^+$
$\mathcal{I}_{21} \wedge \neg\, C_{14} \wedge C_{16} \wedge \neg C'_{16}$	The parking node, which is unsaturated and has the highest order in $\mathcal{H}^-$ among all the unsaturated nodes in $\mathcal{H}^-$
$\mathcal{I}_{22} \wedge \neg\, C_{14}$	The two parking nodes that have the highest order among all the unsaturated parking nodes and lying on two different half-planes

nodes are symmetric with respect to l and there either does not exist any parking node on l or each parking node on l is saturated. In Fig. 5, A and B are the leading corners. p_1 is the parking node in $\mathcal{H}^+$ which has the highest order. The target parking nodes are selected in the order (p_1, p_3, p_2, p_4).

4.4 Candidate Robot Selection Phase

In view of Lemma 1, while a robot moves towards a parking node, it must ensure collision-free movement. Otherwise, the problem becomes unsolvable. As a result, a robot will move toward its target only when it has a path toward that target that does not contain any other robot positions.

Definition 3. *A path from a robot to a parking node is said to be free if it does not contain any other robot positions.*

A robot would move toward its target only when it has a free path toward it. In this phase, the *candidate robot* is selected and allowed to move toward the target parking node. Let $p \neq p_{guard}$ be the target parking node selected in the TPS phase. Depending on the different classes of configurations, the following cases are to be considered.

1. $C(t)$ is asymmetric. As a result, the robots are orderable. The robot that does not lie on any saturated parking node and has the shortest free path to p is selected as the candidate robot. If multiple such robots exist, the one with the highest order among such robots is selected as the candidate robot.
2. $C(t)$ is symmetric with respect to a single line of symmetry l. First, assume that p is on l. If at least one robot exists on l, then the symmetry can be broken by allowing a robot from l to move towards an adjacent node away from l. As a result, assume that there is no robot position on l. The two closest robots, which do not lie on any saturated parking node and have shortest free paths towards p, are selected as the candidates for p. If there are multiple such robots, the ties are broken by considering the robots that lie on different half-planes and have the highest order among all such robots. Next, assume that p is on the half-planes. The robot that does not lie on any saturated parking node and has a shortest free path toward p is selected as the candidate robot. Note that such candidates are selected in both half-planes.
3. $C(t)$ is symmetric with respect to rotational symmetry. First, assume that p is on c. If there exists a robot on c, the robot on c moves towards an adjacent node, and the configuration becomes asymmetric. Assume the case when there are no robots on c. The robots that are closest to p are selected as candidate robots. In this case, depending on whether the angle of rotational symmetry is 180° or 90°, two or four robots are selected as candidates. Next, assume that p is located either on a quadrant or on of the wedge boundaries. If the target parking node lies on a quadrant, the robot that does not lie on any saturated parking node and has a shortest free path toward p is selected as the candidate robot. It should be noted that such candidates are chosen from each of the four quadrants, for each target parking node. Otherwise, if the

target parking node is on a wedge boundary, the robot(s) not lying on any saturated parking node and having a shortest free path towards the target is (are) selected as candidate robot(s).

Next, assume that p_{guard} is the target parking node. The candidates are selected as the robot which has shortest free path towards p_{guard}. Finally, the guard moves towards p_{guard}. By the choice of p, there always exists a half-line starting from p, which does not contain any robot position. As a result, a free path always exists between the candidate robot and p.

4.5 Guard Movement

Assume the case when the parking nodes are symmetric and the configuration is asymmetric. In the GM phase, the guard moves toward its respective destination and the parking process is terminated. The guard moves only when it finds that, except for one, all the parking nodes have become saturated. It moves towards its destination p in a free path. The guard moves towards its destination and each parking node becomes saturated, transforming the configuration into a final configuration.

5 Correctness

Due to space constraints, the proofs of Lemmas 4–9 and Theorem 1 are mentioned in the arxiv version of the paper [5].

Lemma 4. *In the GS phase, the guard remains invariant while it moves towards its destination.*

Lemma 5. *During the execution of the algorithm Parking(), if the parking nodes admit a single line of symmetry l, then $\mathcal{H}^+$ remains invariant.*

Lemma 6. *During the execution of the algorithm Parking(), if the parking nodes admit rotational symmetry, then $\mathcal{Q}^{++}$ remains invariant.*

Lemma 7. *If the configuration is such that the parking nodes admit a unique line of symmetry l, then during the execution of the algorithm Parking(), the target parking nodes remain invariant.*

Lemma 8. *If the configuration is such that the parking nodes admit rotational symmetry, then during the execution of the algorithm Parking(), the target parking nodes remain invariant.*

Lemma 9. *During the CRS phase, the candidate robot remains invariant.*

Theorem 1. *Algorithm Parking() solves the Parking Problem in Infinite grids for all initial configurations not belonging to the set $\mathcal{U}$.*

6 Conclusion

This chapter proposed a deterministic distributed algorithm for solving the parking problem in infinite grids. We have characterized all the initial configurations and the values of k_i for which the problem is unsolvable, even if the robots are endowed with strong multiplicity detection capability. A deterministic algorithm has been proposed under the assumption that the robots are endowed with global-strong multiplicity detection capability. As a future work, it would be interesting to investigate the problem in case the number of robots is not equal to the sum of the capacities of the parking nodes. In case the number of robots in the initial configuration is less than the sum of the capacities of the parking nodes, one interesting study could be to investigate the problem with the objective of maximizing the number of saturated parking nodes.

References

1. Bhagat, S., Chakraborty, A., Das, B., Mukhopadhyaya, K.: Gathering over meeting nodes in infinite grid*. Fundam. Inform. **187**(1), 1–30 (2022)
2. Bhagat, S., Chakraborty, A., Das, B., Mukhopadhyaya, K.: Optimal gathering over weber meeting nodes in infinite grid. Int. J. Found. Comput. Sci. **34**(1), 25–49 (2023)
3. Bhagat, S., Das, B., Chakraborty, A., Mukhopadhyaya, K.: k-Circle formation and k-epf by asynchronous robots. Algorithms **14**(2) (2021)
4. Bose, K., Adhikary, R., Kundu, M.K., Sau, B.: Arbitrary pattern formation on infinite grid by asynchronous oblivious robots. Theor. Comput. Sci. **815**, 213–227 (2020)
5. Chakraborty, A., Mukhopadhyaya, K.: Parking problem by oblivious mobile robots in infinite grids (2023)
6. Cicerone, S., Stefano, G.D., Navarra, A.: Gathering of robots on meeting-points: feasibility and optimal resolution algorithms. Distrib. Comput. **31**(1), 1–50 (2018)
7. Cicerone, S., Stefano, G.D., Navarra, A.: Asynchronous arbitrary pattern formation: the effects of a rigorous approach. Distrib. Comput. **32**(2), 91–132 (2019)
8. Cicerone, S., Stefano, G.D., Navarra, A.: Embedded pattern formation by asynchronous robots without chirality. Distrib. Comput. **32**(4), 291–315 (2019)
9. D'Angelo, G., Stefano, G.D., Klasing, R., Navarra, A.: Gathering of robots on anonymous grids and trees without multiplicity detection. Theor. Comput. Sci. **610**, 158–168 (2016)
10. Das, B., Chakraborty, A., Bhagat, S., Mukhopadhyaya, K.: k-circle formation by disoriented asynchronous robots. Theor. Comput. Sci. **916**, 40–61 (2022)
11. Efrima, A., Peleg, D.: Distributed algorithms for partitioning a swarm of autonomous mobile robots. Theor. Comput. Sci. **410**(14), 1355–1368 (2009). structural Information and Communication Complexity (SIROCCO 2007)
12. Flocchini, P., Prencipe, G., Santoro, N.: Distributed computing by mobile entities. Curr. Res. Mov. Comput. **11340** (2019)
13. Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Arbitrary pattern formation by asynchronous, anonymous, oblivious robots. Theor. Comput. Sci. **407**(1–3), 412–447 (2008)

14. Fujinaga, N., Ono, H., Kijima, S., Yamashita, M.: Pattern formation through optimum matching by oblivious CORDA robots. In: Lu, C., Masuzawa, T., Mosbah, M. (eds.) OPODIS 2010. LNCS, vol. 6490, pp. 1–15. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17653-1_1
15. Stefano, G.D., Navarra, A.: Gathering of oblivious robots on infinite grids with minimum traveled distance. Inf. Comput. **254**, 377–391 (2017)
16. Sugihara, K., Suzuki, I.: Distributed algorithms for formation of geometric patterns with many mobile robots. J. Field Robot. **13**(3), 127–139 (1996)
17. Yamauchi, Y., Yamashita, M.: Pattern formation by mobile robots with limited visibility. In: Moscibroda, T., Rescigno, A.A. (eds.) SIROCCO 2013. LNCS, vol. 8179, pp. 201–212. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-03578-9_17