# Asynchronous Exclusive Perpetual Grid Exploration without Sense of Direction

François Bonnet, Alessia Milani, Maria Potop-Butucaru, Sébastien Tixeuil

# Asynchronous Exclusive Perpetual Grid Exploration without Sense of Direction

François Bonnet[1], Alessia Milani[2], Maria Potop-Butucaru[3], and Sébastien Tixeuil[3]

[1] School of Information Science, JAIST
[2] Université de Bordeaux 1, LaBRI
[3] UPMC Sorbone Universités, LIP6

**Abstract.** In this paper, we investigate the exclusive perpetual exploration of grid shaped networks using anonymous, oblivious and fully asynchronous robots. Our results hold for robots without sense of direction (*i.e.* they do not agree on a common North, nor do they agree on a common left and right ; furthermore, the "North" and "left" of each robot is decided by an adversary that schedules robots for execution, and may change between invocations of particular robots). We focus on the minimal number of robots that are necessary and sufficient to solve the problem in general grids.

In more details, we prove that three deterministic robots are necessary and sufficient, provided that the size of the grid is $n \times m$ with $3 \leq n \leq m$ or $n = 2$ and $m \geq 4$. Perhaps surprisingly, and unlike results for the exploration with stop problem (where grids are "easier" to explore and stop than rings with respect to the number of robots), exclusive perpetual exploration requires as many robots in the ring as in the grid.

Furthermore, we propose a classification of configurations such that the space of configurations to be checked is drastically reduced. This pre-processing lays the bases for the automated verification of our algorithm for general grids as it permits to avoid combinatorial explosion.

## 1 Introduction

We consider a set of autonomous robots that have to collaborate to perpetually explore an area modeled as a grid graph. Each robot has to visit each vertex of the grid infinitely many times, with the additional constraint that no two robots may be present at the same vertex at the same time or may switch their positions by crossing the same edge. Introduced in [1], this problem is called the *exclusive perpetual exploration.*

Robots are endowed with visibility sensors and motion actuators and operate in *cycles* that comprise *look*, *compute*, and *move* phases. The look phase consists in taking a snapshot of the other robots positions using its visibility sensors. In the compute phase a robot computes a target destination based on the previous observation. The move phase consists in moving toward the computed destination using motion actuators. We consider an asynchronous scheduling model, *i.e.* a finite yet unbounded amount of time may elapse between any two phases of a robot's cycle.

The difficulty to solve the *exclusive perpetual exploration* on the grid, depends on the restricted robot capabilities and on their asynchronous behavior. In particular, robots are *anonymous* (they execute the same protocol and have no mean to distinguish themselves from other robots), *oblivious* (their memory is not persistent between cycles), and they have no sense of direction (*i.e.* they may not agree on a common direction or orientation in the grid as each local direction and orientation are chosen by an adversary and may change between cycles). Asynchrony makes the problem harder because robots have to coordinate their movements despite the fact that a robot can decide to move according to an old snapshot of the system and that different robots may be in different phases of their cycles at the same time.

The robots' positions in the grid is the only information that can be use to decide moving. In this context, the number of robots on the grid directly impacts on the ability to break initial symmetry and to provide a grid reference "milestone" to help robots in their exploration process. Observe that due to the mutual exclusion constraints on the vertices occupancy, we cannot exploit the usual approach [6] of creating "towers" (having more than one robot on the same node) to construct a reference milestone in the graph to be explored.

*Related work* While the vast majority of literature on coordinated distributed robots considers that those robots are evolving in a *continuous* two-dimensional Euclidean space and use visual sensors with perfect accuracy that permit to locate other robots with infinite precision, a recent trend was to shift from the classical continuous model to the *discrete* model. In the discrete model, space is partitioned into a *finite* number of locations. This setting is conveniently represented by a graph, where nodes represent locations that can be sensed, and where edges represent the possibility for a robot to move from one location to the other. Thus, the discrete model restricts both sensing and actuating capabilities of every robot. For each location, a robot is able to sense if the location is empty or if robots are positioned on it (instead of sensing the exact position of a robot). Also, a robot is not able to move from a position to another unless there is explicit indication to do so (*i.e.*, the two locations are connected by an edge in the representing graph). The discrete model permits to simplify many robot protocols by reasoning on finite structures (*i.e.*, graphs) rather than on infinite ones. However, as noted in most related papers [5–7, 9–12], this simplicity comes with the cost of extra symmetry possibilities, especially when the authorized paths are also symmetric.

Assuming visibility capabilities, anonymous and oblivious robots, the three main problems that have been studied in the discrete robot model are *gathering* [9–11] (all robots are requested to reach a single node, not known beforehand), exploration with stop [4–7, 12] (all nodes must be visited by at least one robot, and eventually all robots must stop moving forever), and exclusive perpetual exploration [1, 2] (all nodes must be visited by all robots infinitely often, and no node or edge should be occupied by more than one robot at any time).

For the exploration with stop problem, the fact that robots need to stop after exploring all locations requires robots to "remember" how much of the graph was explored, *i.e.*, be able to distinguish between various stages of the exploration process since robots have no persistent memory. As configurations can be distinguished only by robot positions, the main complexity measure is then the number of robots that are needed to explore a given graph. The vast number of symmetric situations induces a large number of required robots. For tree networks, [7] shows that $\Omega(n)$ robots are necessary for most $n$-sized tree, and that sublinear robot complexity is possible only if the maximum degree of the tree is 3. In uniform rings, [6] proves that the necessary and sufficient number of robots is $\Theta(\log n)$, although it is required that the number $k$ of robots and the size $n$ of the ring are coprime. Note that both approaches are *deterministic*, *i.e.*, if a robot is presented twice the same situation, its behavior is the same in both cases. In [3, 5], the authors propose to adopt a *probabilistic* approach to lift constraints and to obtain tighter bounds. They show that *four* identical probabilistic robots are necessary and sufficient to solve the exploration problem in any anonymous unoriented ring, also removing the coprime constraint between the number of robots and the ring size. The impossibility presented in [5] for three robots to explore a ring is for the semi-synchronous model and naturally extends to the asynchronous model. This impossibility result extends to four robots in the deterministic setting, while five robots are sufficient to explore and stop a ring deterministically [12]. By contrast, it was recently pointed out that in the general case three robots are necessary and sufficient to explore an $n \times m$ grid-shaped network with $m > 3$ [4].

So, with respect to the required number of robots to explore and stop a graph, grid exploration is easier that ring exploration.

Most related to our task is the exclusive perpetual exploration of anonymous graphs [1, 2]. Baldoni *et al.* [1] prove that the mutual exclusion constraint and the underlying graph structure drive an upper bound on the number of robots that can perpetually and exclusively explore a graph. They also provide a method to compute this upper bound for a given graph. In the same paper, they provide an algorithm to perpetually and exclusively explore any partial grid provided that the number of robots enables a solution. However, the considered scheduling model is synchronous and contrary to the vast majority of existing solutions in the literature [4–7, 9–12], robots have a strong sense of direction as they agree on the four basic directions: North, South, East, and West. This strong settings permits to break all cases of initial symmetry since a global total order can be inferred on the vertices of the graph. Finally, Blin *et al.* [2] investigate exclusive perpetual exploration of an anonymous ring. Considering asynchronous scheduling and no sense of direction, they prove that three robots are necessary and sufficient to perpetually and exclusively explore a ring.

*Our contribution* First, we prove that three deterministic robots are necessary and sufficient, provided that the size of the grid is $n \times m$ with $3 \leq n \leq m$ or $n = 2$ and $m \geq 4$. Perhaps surprisingly, and unlike results for the exploration with stop problem (where grids are "easier" to explore and stop than rings with respect to the number of robots), exclusive perpetual exploration requires as many robots in the ring and in the grid.

Second, we propose a general classification of all possible system configurations. Observing that in each symmetric configuration, deterministic robots act in the same manner, we consider a single representative configuration for each set of such symmetric configurations. Moreover, we classify representative configurations with respect to the positions of robots and their mobility constraints. Some configurations are more complex to manage than others, as several robots may be scheduled to move concurrently. Then, due to asynchrony one robot can move before the other, leading to a new unstable configuration where the second robot may move according to an old snapshot. We expect that our classification benefits protocol designers to ease the proving of exploration process and lays the basis of their automatic verification.

## 2 Model and Problem Specification

We consider a distributed system of $k$ mobile robots that are scattered on an $n \times m$ grid graph where $2 \leq n \leq m$. The grid is *anonymous*, *i.e.*, there exists no labeling to distinguish nodes or edges. The robots are *identical*, *i.e.*, they cannot be distinguished using their appearance and they execute the same protocol. Moreover, the robots are *oblivious*, *i.e.*, they have no memory of their past actions, and they neither have a common North direction, nor a common chirality (handedness). Robots cannot explicitly communicate, but have the ability to sense their environment and see the position of the other robots.

Robots operate in three phase cycles: Look, Compute and Move. During the Look phase robots take a snapshot of the grid together with the robots. The collected information (position of the other robots in the egocentered view) are used in the compute phase in which robots decide to move or to stay idle. In the move phase, robots may move to one of their adjacent nodes computed in the previous phase.

The computational model we consider is the asynchronous CORDA model [8, 13] in a discrete setting. Thus, when a robot takes a snapshot of the grid, it sees the other robots on nodes only. On the other hand, the time between Look, Compute, and Move operations is finite but unbounded, and it is decided by the adversary for each action of each robot. Thus, because of the asynchrony, different robots can execute concurrently different phases (*e.g.*, a robot can perform a look operation while another robot is moving), and a robot can use an outdated snapshot of the grid to compute where to move and whether to move. A configuration at a given time is defined by the positions of all robots at that time. We assume that initially every vertex of the grid contains at most one robot.

**Exclusive Perpetual Exploration Problem**

We study the *Exclusive Perpetual Exploration* problem introduced by Baldoni *et al.* [1] for the synchronous setting. We restate the problem for the asynchronous setting.

**Definition 1 (Perpetual Exclusive Grid Exploration).** *For any grid $G$ of size $n \times m$ and for any initial configuration where all robots are located on different vertices, an algorithm solves the perpetual exclusive grid exploration problem if it guarantees the following two properties:*

- ***liveness*** *Each robot visits each vertex in $G$ infinitely often.*
- ***no collision*** *No two robots visit the same vertex or traverse the same edge at the same time.*

## 3 Classification of configurations

The lack of a global orientation implies that many configurations are alike from a robot's point of view. Two configurations are *indistinguishable* if they are symmetric in the robots egocentered view (see Figure 1). However, when the grid is not square (that is, when $n \neq m$) robots can distinguish landscape and portrait orientation. In this paper, we suppose *w.l.o.g.* that all configurations are in a landscape orientation.

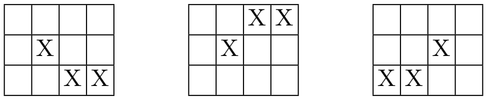


**Fig. 1.** Three indistinguishable configurations on a $3 \times 4$ grid.

*Representative Configuration* For notation purpose, we label each cell of a grid $n \times m$ with an integer in the set $\{0, \ldots, nm-1\}$ according to the left-to-right-top-to-bottom reading direction. This labeling is not available to the robots.

Given an $n \times m$ grid, any configuration involving three robots on different vertices is conveniently represented by a unique sorted sequence $s = (s_1, s_2, s_3)$ where each $s_i \in \{0, \ldots, nm-1\}$ is different and represents one of the cells occupied by the robots. The three configurations on Figure 1, for example, are respectively denoted by the sorted sequences $(5, 10, 11)$, $(2, 3, 5)$, and $(6, 8, 9)$.

Since robots are identical, deterministic, and execute the same code, in every indistinguishable configuration, robots perform the same actions. In each set of indistinguishable configurations, we choose a *representative.*

**Definition 2 (representative configuration).** *Let $G$ be a grid with left-to-right-top-to-bottom labeling, and any non-empty set $\mathcal{I}$ of indistinguishable configurations of $G$, we say that a configuration $s \in \mathcal{I}$ is the* representative configuration *of every configuration in $\mathcal{I}$ iff $s$ is the smallest in $\mathcal{I}$* w.r.t. *the lexicographic order, $\prec$. That is, there is no $s' \in \mathcal{I}$ such that $s' \prec s$.*

Considering Figure 1, the natural labeling is and is the representative configuration, whose corresponding sorted sequence is $(0, 1, 6)$, which is the smallest amongst all indistinguishable configurations. In the sequel, we consider only representative configurations that are thus simply called configurations.

| | | | |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 |

| | | | |
|---|---|---|---|
| X | X | | |
| | | X | |
| | | | |

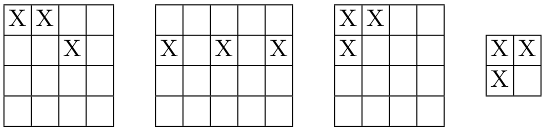
## Classes of configurations (for grids with 3 robots)

Given a particular configuration, we say that two robots are *indistinguishable* iff there is no deterministic way of distinguishing them. For example, in the second grid on Figure 2, the robots respectively located at cell 5 and cell 9 are indistinguishable.

Formally, in a configuration $C$ there are two *indistinguishable* robots iff $C$ is axially symmetric and two robots are symmetric in this axial symmetry[4]. For the second grid on Figure 2, the axis of symmetry is the vertical line that goes through the middle of the third column. For the third grid on Figure 2, the axis of symmetry is the diagonal line that goes from top-left to bottom-right.

Breaking and avoiding symmetric configurations is crucial to guarantee both liveness and the absence of collisions. Thus, we classify the set of configurations into four classes, depending on the amount of symmetry (with respect to the three robots) and the move possibilities of robots:

1. *Asymmetrical configuration*: every robot is uniquely distinguishable. In this case, even an adversarial scheduling cannot lead to ambiguous situations.
2. *Semi-symmetrical configuration*: two robots have symmetrical views but the third. This third robot can break the symmetry, possibly in more than one step.
3. *Semi-symmetrical blocked configuration*: two robots are indistinguishable, the other robot cannot move, but the two indistinguishable robots can move, without violating the vertex mutual exclusion (the no collision property). These configurations are difficult to deal with, since there is no immediate way of breaking the symmetry. One can nevertheless observe that any algorithm solving the perpetual exclusive exploration problem has to enable the movement of the two indistinguishable robots (the third robot is blocked).
4. *Symmetrical blocked configuration*: no possible move. For any possible move, an adversarial scheduling can decide an activation order that would lead to a collision.



| | | | |
|---|---|---|---|
| X | X | | |
| | | X | |
| | | | |
| | | | |

| | | | | |
|---|---|---|---|---|
| | | | | |
| X | | X | | X |
| | | | | |
| | | | | |

| | | | |
|---|---|---|---|
| X | X | | |
| X | | | |
| | | | |
| | | | |

| | |
|---|---|
| X | X |
| X | |

**Fig. 2.** The four types of configurations.

[4] If such an axis exists, it is necessarily the perpendicular bisector of the segment whose extremities are the two indistinguishable robots. The third robot lies on that axis.

From an algorithmic point of view, configurations of type 1 and 2 are ideal, while the configuration of type 4 is a dead-end. Thus, when designing an algorithm one should focus on configurations of type 3 which are the most complex to deal with. The complexity arises from the fact that the two indistinguishable robots may be scheduled to execute concurrently, but due to the asynchrony one robot may move first, bringing the system to a new configuration, while the second robot will move later and according to an old snapshot. This means that the system may transiently be in configurations of type 1 or 2 but having a robot move later because of a snapshot it took when in a configuration of type 3. The algorithm should be resilient to this unexpected movement.

## 4 Algorithms for the Perpetual Grid Exploration

### 4.1 Impossibility results

**Theorem 1.** *There is no algorithm that solves the perpetual exploration problem*

- *with one robot for any grid but the* $1 \times 1$ *and* $1 \times 2$ *grid,*
- *with two robots for any grid.*

*Proof.* With one robot, since there are less configurations than the number of vertices, it is not possible to design a deterministic algorithm[5].

With two robots, there exist symmetric configurations, *e*.g. robots on two corners. If the scheduler always activate simultaneously both robots, it is not possible to break the symmetry and thus robots cannot explore the grid; each robot will stay in its half-side of the grid.

**Theorem 2.** *There is no algorithm that solves the perpetual exploration problem with three robots in the* $2 \times 2$ *grid or* $2 \times 3$ *grid.*

*Proof.* In the grid of size $2 \times 2$ there is only one possible configuration with three robots and it is of type 4. Therefore, no move can be scheduled. Otherwise, robots would collide.

Consider then a grid size $2 \times 3$ and suppose there is an algorithm $\mathcal{A}$ that solves the problem from the initial configuration $(0, 1, 2)$. From this configuration, $\mathcal{A}$ can only decide to move the central robot, or to move the two other robots (we can suppose that the scheduler will always activate both of them simultaneously). In both situations, the system arrives in the configuration $(0, 2, 4)$ from which $\mathcal{A}$ needs to move the robot located on position 4, otherwise the system comes back to the initial configuration. It is important to notice that the scheduler decides if the moving robot moves to the left or to the right; in both cases the system arrives in the configuration $(0, 2, 3)$ but the robot does not visit the same cell.

From the configuration $(0, 2, 3)$, the algorithm $\mathcal{A}$ can only move (i) the robot located on position 2 to the left, or (ii) the robot on position 0 to the right; any other move will bring the system back to a previous configuration, that will create a perpetual loop which does not achieve the perpetual exploration. Let us consider only the case (i), the other case can be analyzed similarly. After the move, the system arrives in configuration $(0, 1, 3)$ and the algorithm has only the choice to move the robot on position 3 to position 4 and then the same robot to position 5 and finally to position 3 to arrive in the previous configuration $(0, 1, 3)$ from which the algorithm will repeat forever the same loop. During this loop, all the cells have been explored, but not by all robots; in particular the robot located on position 0 never moved.

It is important to notice that this impossibility comes from the ability of the scheduler to decide if the robot located on position 4 of configuration $(0, 2, 4)$ moves to the right or left.

[5] One probabilistic robot can explore any grid using a simple random walk.

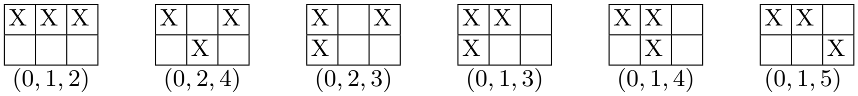


**Fig. 3.** Some configurations for the 2 × 3 grid.

### 4.2 Decomposition in two sub-problems

The exclusive perpetual exploration problem as defined in Section 2 can be decomposed into two independent sub-problems. To complete the exclusive perpetual exploration, one needs to solve both problems. The first sub-problem is directly related to the perpetual exploration, while the second sub-problem deals with the transient period starting from the (arbitrary) initial configuration.

1. Find a sequence of asymmetrical configurations (*i*.e. configurations of type 1) such that when robots "execute" this sequence, each robot visits each cell. This sequence must start and end with the same configuration, and in each step of the sequence exactly one robot moves.
2. From any initial configuration, find an algorithm that allows robots to reach, in a finite number of steps, a configuration that belongs to the previous sequence. (It does not need to be the first one, since the sequence is a cycle.)

One can see the first sub-problem as a weaker version of the perpetual exploration problem when the initial configuration of robots is chosen by the algorithm.

### 4.3 Grid 3 × 3

**Theorem 3.** *There exists an algorithm that solves the exclusive perpetual exploration with three robots starting from any initial configuration in the* 3 × 3 *grid.*

*Proof. Sub-problem* 1 The Figure 4 describes how the perpetual exploration is executed. For each configuration, the blue robot represents the robot that is allowed to move in that configuration. After 4 moves, the initial configuration is reached again with a rotation of the grid by 90° and a circular permutation of the positions of robots. In this process, one robot (the one initially on position 5) has visited the four colored cells. The union of this set of cells and the same sets of cells after rotations of 90°, 180°, and 270° covers the entire grid. Thus after 12 iterations[6] of this sequence (*i.e.* 48 moves) every robot has visited every cell of the grid.

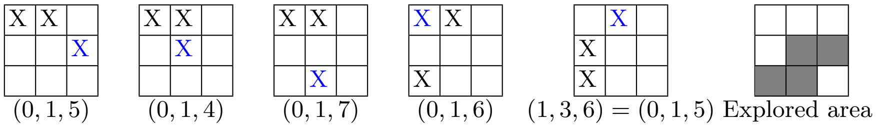


**Fig. 4.** Sequence that achieves the perpetual exploration for the 3 × 3 grid.

*Sub-problem* 2 There is no configuration of type 4 and exactly one of type 3. For all other configurations (*i.e.* configurations of type 2, there is no problem since an algorithm can deterministically

[6] $12 = 3\text{positions}\times 4\text{orientations}$: It guarantees that each robot starts in position 5 for each of the four orientations of the grid.

select one robot and make it move to reach a configuration of type 1; for example from configuration $(0, 1, 8)$, the system evolves to configuration $(0, 1, 7)$. Figure 5 explains how the algorithm deals with the single configuration of type 3. The main problem lies in the fact that only the two indistinguishable robots can move but due to asynchrony, it is possible that *(a)* only one robot sees this configuration and moves (the other one never sees this configuration); the system arrives in $(0, 1, 6)$, or *(b)* both robots see the configuration, compute their moves and then one robot moves; the system arrives in $(0, 1, 6)$. The color red for the robot in position 1 means that the system is in the configuration where this robot has computed its move (phase look/compute) according to a previous configuration but has not yet accomplished its move (phase move).

This is problematic because the system arrives in a configuration $(0, 1, 6)$ where the robot located at position 0 is supposed to be the (only) robot to move (from our sequence of moves described in Figure 4). Thus from configuration $(0, 1, 6)$, three cases are possible depending on the first moving robot:

- The robot in 1 moves before the robot in 0 sees the current configuration. The system reaches $(0, 2, 6)$.
- The robot in 1 moves after the robot in 0 sees the current configuration. The system reaches $(0, 2, 6)$ and then $(0, 1, 8)$.
- The robot in 0 sees the current configuration and moves before the robot in 1 moves. The system arrives in $(0, 1, 5)$ and then $(0, 1, 8)$.

In all cases, the system leaves the "bad" configuration $(0, 1, 3)$ of type 3 and arrives, in at most three steps, in a configuration of type 1 or 2 without any "red" robot. The complete list of moves appears in Appendix A.

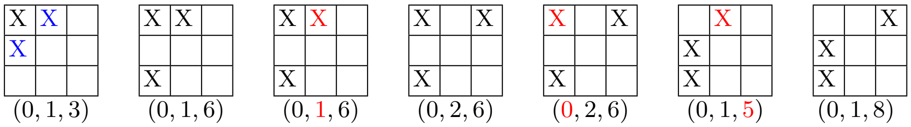


**Fig. 5.** From the configuration $(0, 1, 3)$ of type 3 to a configuration of type 1 or 2.

### 4.4 Grids $2 \times m$ with $m \geq 4$

**Theorem 4.** *There exists an algorithm that solves the exclusive perpetual exploration with three robots starting from any initial configuration in* $2 \times m$ *grids, with* $m \geq 4$.

*Proof. Sub-problem* 1 One valid sequence is described informally on Figure 6 and the formal definition appears in Appendix B. After $2m$ moves, the system returns to the initial configuration with a circular permutation of the robot positions. In this process, one robot (the one initially in position 1) has visited all positions but 0 (Position 0 will be visited by this robot within the next $2m$ moves). Thus, after 3 iterations of this sequence (*i.e.* $3 \times 2m$ moves), every robot has visited every cell of the grid.

*Sub-problem* 2 There is no configuration of type 3 or 4. There exist configurations of type 2 only when $m$ is odd; from these configurations it is easy to break the symmetry by moving the robot which is uniquely distinguishable. From any configuration of type 1, it is sufficient to move

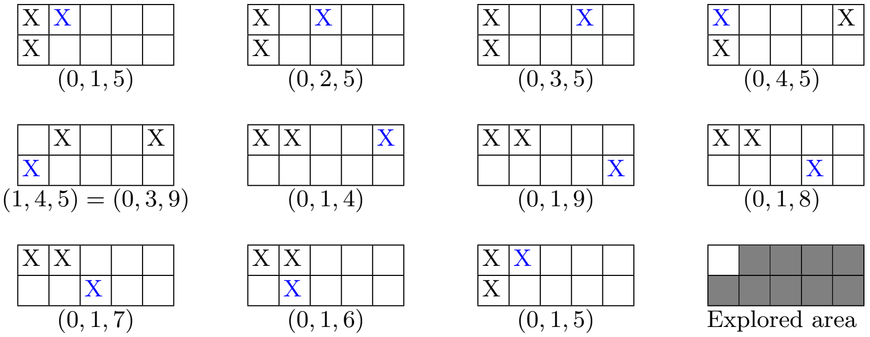


**Fig. 6.** Sequence for the perpetual exploration for 2 × $m$ grids ($m \geq 4$). Here $m = 5$.

successively one robot to position 0, then one robot to position 1, then the last one in a position between $m-1$ and $2m-1$. The total number of moves in this process is bounded by $m+m+1$. The complete list of moves appears in Appendix B.

### 4.5 Grids $n \times m$ with $3 \leq n < m$

**Theorem 5.** *There exists an algorithm that solves the exclusive perpetual exploration with three robots starting from any initial configuration in $n \times m$ grids, with $3 \leq n < m$.*

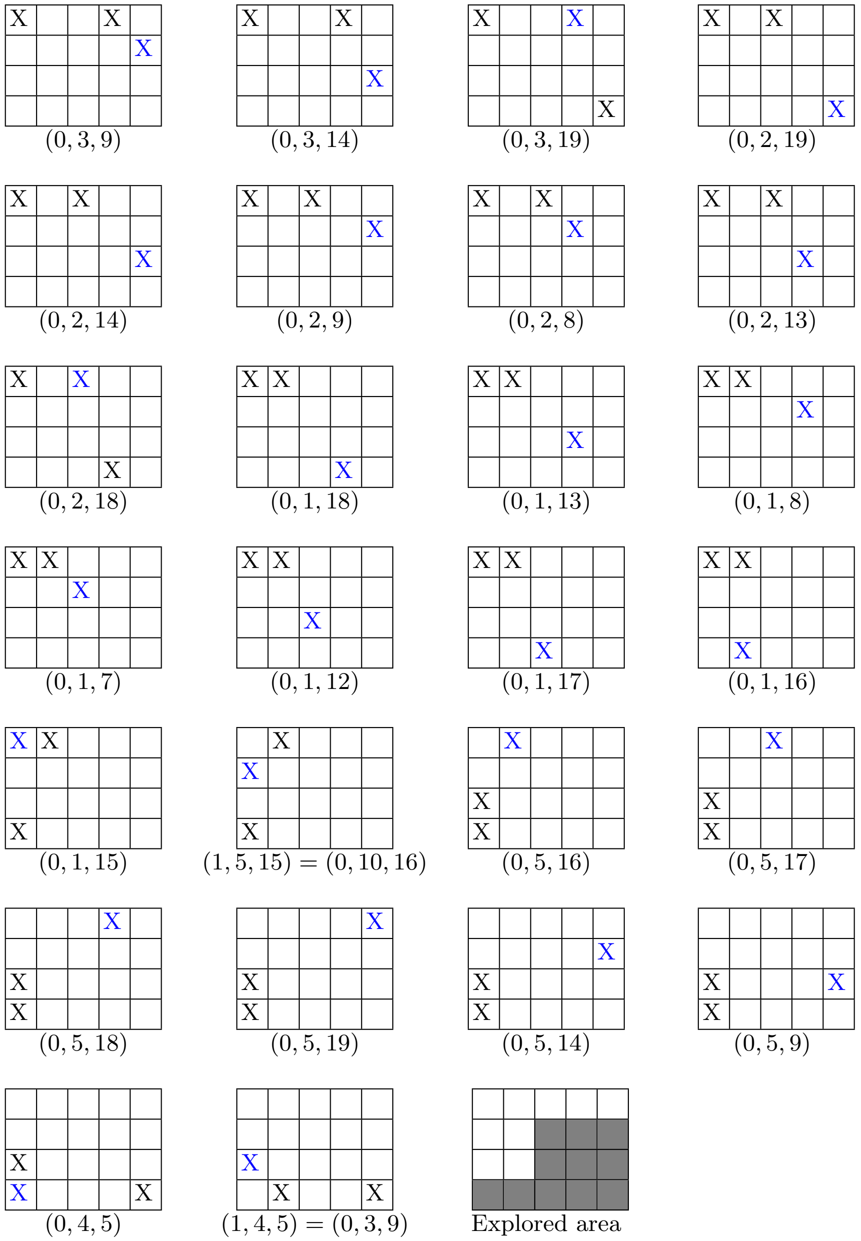
*Proof.* *Sub-problem* 1 One valid sequence is described informally on Figure 7 and formally in Appendix C. After $(2n-2)\times(m-3)+n+1+(n-3)+(m-2)+(n-1)+1$ moves, the configuration of the system returns to the initial configuration with a circular permutation of the three robots and a rotation of the grid by 180°. Doing zigzag moves, one robot (the one initially located on position $2m-1$) explores all the grid but the first line and the two first columns. Thus, after 6 iterations[7] every robot has visited every vertex of the grid.

*Sub-problem* 2 There exists no configuration of type 3 or 4. There exist configurations of type 2 only when $m$ or $n$ is odd; from these configurations it is easy to break the symmetry by moving the robot which is uniquely distinguishable. As for the previous case, we show that, for any configuration of type 1, it is possible to reach one configuration used during the perpetual exploration. It is sufficient to move successively one robot to the position 0, then one robot to the position 1 and finally one robot to the position $(n-1)m$ to reach configuration $C = (0, 1, (n-1)m)$. However, depending on the initial configuration, it may happen that, while executing these moves, the system reaches a configuration $C'$ used in the perpetual exploration phase; in that case, the sub-problem 2 is solved and robots start the perpetual exploration from this configuration $C'$, without reaching $C$. Indeed, a move is already defined from $C'$ and it is not possible to define a different one for this phase since robots are oblivious. The total number of moves is grossly bounded by $nm+nm+nm$. The complete list of moves appears in Appendix C.

### 4.6 Grids $n \times n$ with $n \geq 4$

**Theorem 6.** *There exists an algorithm that solves the exclusive perpetual exploration with three robots starting from any initial configuration in $n \times n$ grids, with $n \geq 4$.*

[7] 6 = 3 positions × 2 orientations: It guarantees that each robot starts in position $2m-1$ for each of the two orientations of the grid.



$(0,3,9)$ $(0,3,14)$ $(0,3,19)$ $(0,2,19)$

$(0,2,14)$ $(0,2,9)$ $(0,2,8)$ $(0,2,13)$

$(0,2,18)$ $(0,1,18)$ $(0,1,13)$ $(0,1,8)$

$(0,1,7)$ $(0,1,12)$ $(0,1,17)$ $(0,1,16)$

$(0,1,15)$ $(1,5,15) = (0,10,16)$ $(0,5,16)$ $(0,5,17)$

$(0,5,18)$ $(0,5,19)$ $(0,5,14)$ $(0,5,9)$

$(0,4,5)$ $(1,4,5) = (0,3,9)$ Explored area

**Fig. 7.** Sequence for the perpetual exploration of $n \times m$ grids. Here $(n, m) = (4, 5)$.

*Proof. Sub-problem* 1 One valid sequence is described informally on Figure 8 and formally in Appendix D. After $(2n-2) \times (n-3) + n + 1$ moves, the configuration of the system returns to the initial configuration with a rotation of the grid by 90° and a circular permutation of the positions of robots. In this process, one robot (the one initially in $2n-1$) has explored all the grid but the first line and the two first columns. Since $n \geq 4$, after 12 iterations every robot has visited every vertex of the grid[8].

*Sub-problem* 2 In the transient phase, robots try to reach configuration $C = (0, 1, (n-1)n)$ by successively moving one robot to position 0, then one to position 1, and finally the last one to position $(n-1)n$. There is however one main difference with the previous case: there exist symmetric configurations where the three robots are not on the same line/column, namely configurations $(0, x, xn)$, with $1 \leq x \leq n-1$ and in particular configuration $(0, 1, n)$, which is of type 3. To deal with these "bad" initial configurations we need a special behavior that is described on Figure 9. Informally, we first move the robot located in 0 to break the symmetry[9] and then we move another robot away from the side of the grid, so that a bad configuration cannot be reached again when the system evolves towards the configuration $(0, 1, (n-1)n)$. The total number of moves is grossly bounded by $4 + n^2 + n^2 + n^2$. The complete list of moves appears in Appendix D.

## 5 Conclusions and open problems

We proved that three robots are necessary and sufficient to solve the problem of the exclusive perpetual exploration of grids. Similarly to [1], it would be interesting to generalize our result to any partial grid. Of course, only partial grids that preserve symmetry patterns are to be considered, as asymmetric partial grids give a global sense of direction for free.

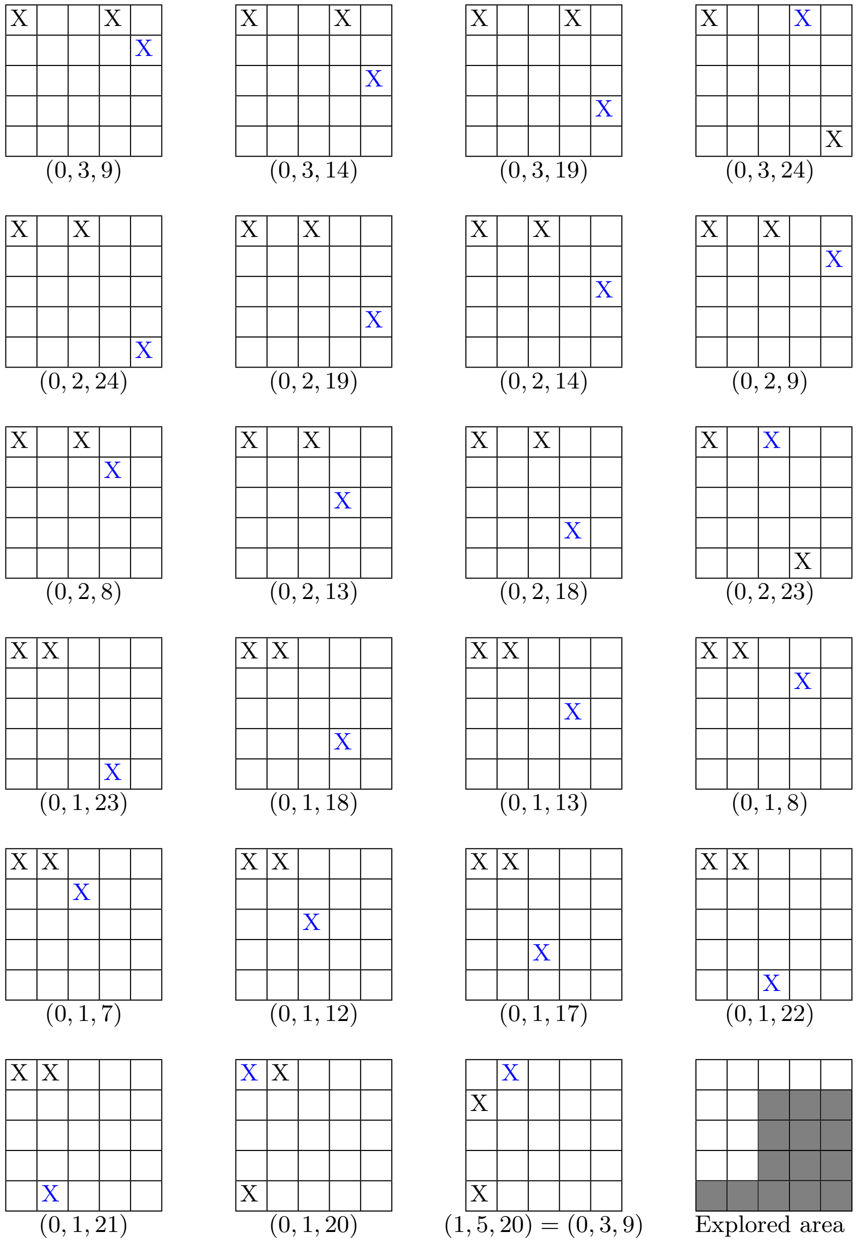
Studying the coordination of robots to solve basic tasks, such as exploration, in a distributed manner requires considering all possible reachable configurations. Our proposal for defining and classifying configurations considerably simplifies the design and verification of our algorithms. We believe it can be extended to address an arbitrary number of robots and be a first step in providing a complete framework to study coordination problems in mobile robots networks.

## References

1. Baldoni, R., Bonnet, F., Milani, A., Raynal, M.: On the solvability of anonymous partial grids exploration by mobile robots. In: 12th International Conference on Principles of Distributed Systems (OPODIS). pp. 428–445 (2008)
2. Blin, L., Milani, A., Potop-Butucaru, M., Tixeuil, S.: Exclusive perpetual ring exploration without chirality. In: Proceedings of DISC 2010. Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Boston, Massachusetts, USA (2010)
3. Devismes, S.: Optimal exploration of small rings. In: Proceedings of the Third International Workshop on Reliability, Availability, and Security. pp. 9:1–9:6. WRAS '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1953563.1953571
4. Devismes, S., Lamani, A., Petit, F., Raymond, P., Tixeuil, S.: Optimal grid exploration by asynchronous oblivious robots. Tech. rep., CoRR (2011), http://arxiv.org/abs/1105.2461

[8] $n \geq 4$ is necessary since for $n = 3$, the central vertex is never visited with this algorithm as "avoiding" the first line and the two first columns prevents robots to visit the central vertex.

[9] Contrary to the $3 \times 3$ grid, configuration $(0, 1, n)$ does not raise issues since configuration $(0, 1, 2n)$ (reached after one move) does not belong to the set of configurations used during the perpetual exploration. From this configuration, we can decide to move the robot in position 2 to position 3, thus this robot executes the same move if it sees the previous configuration or the current configuration.



(0, 3, 9) (0, 3, 14) (0, 3, 19) (0, 3, 24)

(0, 2, 24) (0, 2, 19) (0, 2, 14) (0, 2, 9)

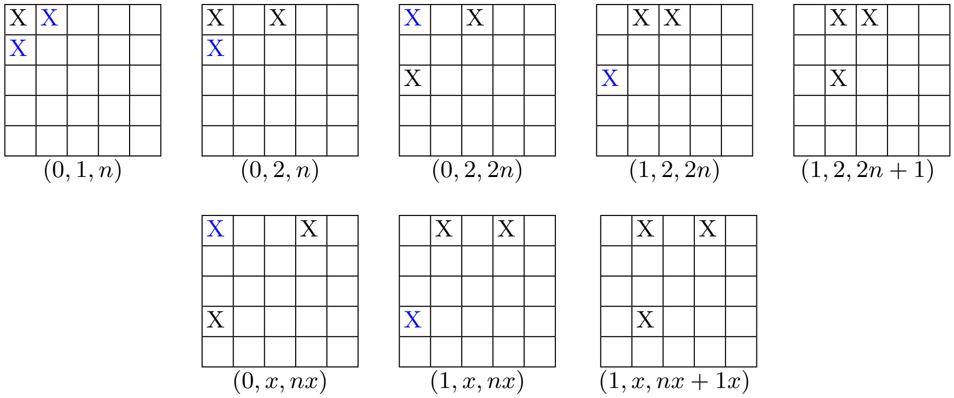
(0, 2, 8) (0, 2, 13) (0, 2, 18) (0, 2, 23)

(0, 1, 23) (0, 1, 18) (0, 1, 13) (0, 1, 8)

(0, 1, 7) (0, 1, 12) (0, 1, 17) (0, 1, 22)

(0, 1, 21) (0, 1, 20) (1, 5, 20) = (0, 3, 9) Explored area

**Fig. 8.** Sequence for the perpetual exploration for $n \times n$ grids ($n \geq 4$). Here $n = 5$.



$(0, 1, n)$ $(0, 2, n)$ $(0, 2, 2n)$ $(1, 2, 2n)$ $(1, 2, 2n + 1)$

$(0, x, nx)$ $(1, x, nx)$ $(1, x, nx + 1x)$

**Fig. 9.** From the configurations $(0, 1, n)$ and $(0, x, xn)$ with $2 \leq x \leq n - 1$ .

5. Devismes, S., Petit, F., Tixeuil, S.: Optimal probabilistic ring exploration by asynchronous oblivious robots. In: Proceedings of Sirocco 2009. Lecture Notes in Computer Science, Springer-Verlag Berlin Heidelberg, Piran, Slovenia (May 2009), http://hal.inria.fr/inria-00360305/fr/
6. Flocchini, P., Ilcinkas, D., Pelc, A., Santoro, N.: Computing without communicating: Ring exploration by asynchronous oblivious robots. In: Proceedings of the International Conference on Principles of Distributed Systems (OPODIS). pp. 105–118. Lecture Notes in Computer Science (LNCS), Springer Berlin / Heidelberg (December 2007)
7. Flocchini, P., Ilcinkas, D., Pelc, A., Santoro, N.: Remembering without memory: Tree exploration by asynchronous oblivious robots. In: 15th International Colloquium on Structural Information and Communication Complexity (SIROCCO). pp. 33–47 (2008)
8. 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)
9. Kamei, S., Lamani, A., Ooshita, F., Tixeuil, S.: Asynchronous mobile robot gathering from symmetric configurations without global multiplicity detection. In: Kosowski, A., Yamashita, M. (eds.) Proceedings of Sirocco 2011. Lecture Notes in Computer Science (LNCS), Springer Berlin / Heidelberg, Gdansk, Poland (June 2011), http://arxiv.org/abs/1104.5660
10. Klasing, R., Kosowski, A., Navarra, A.: Taking advantage of symmetries: Gathering of asynchronous oblivious robots on a ring. In: 12th International Conference on Principles of Distributed Systems (OPODIS). pp. 446–462 (2008)
11. Klasing, R., Markou, E., Pelc, A.: Gathering asynchronous oblivious mobile robots in a ring. Theor. Comput. Sci. 390(1), 27–39 (2008)
12. Lamani, A., Potop-Butucaru, M., Tixeuil, S.: Optimal deterministic ring exploration with oblivious asynchronous robots. In: Patt-Shamir, B. (ed.) Proceedings of Sirocco 2010. Lecture Notes in Computer Science, Springer Berlin / Heidelberg (June 2010)
13. Prencipe, G.: Instantaneous actions vs. full asynchronicity : Controlling and coordinating a set of autonomous mobile robots. In: Restivo, A., Rocca, S.R.D., Roversi, L. (eds.) ICTCS. Lecture Notes in Computer Science, vol. 2202, pp. 154–171. Springer (2001)

## A Grid $3 \times 3$

*Perpetual exploration (sub-problem* 1*):*

- $(0,1,4) \rightarrow (0,1,7)$
- $(0,1,5) \rightarrow (0,1,4)$
- $(0,1,6) \rightarrow (1,3,6) = (0,1,5)$
- $(0,1,7) \rightarrow (0,1,6)$

*Transient phase (sub-problem* 2*):* A “red” robot means that the robots has computed its next move (phase look/compute) in a previous configuration, but has executed it (phase move). Since there is no ambiguity, we do not need to pr

- $(0,1,8) \rightarrow (0,1,7)$
- $(0,1,2) \rightarrow (0,2,4)$
- $(0,2,4) \rightarrow (0,2,3) = (0,1,6)$ or $(0,2,5) = (0,1,6)$[10]
- $(0,2,6) \rightarrow (1,2,6) = (0,1,8)$ or $(2,3,6) = (0,1,8)$[11]
- $(0,2,7) \rightarrow (0,2,6)$
- $(0,4,8) \rightarrow (0,1,8)$[12]
- $(0,5,7) \rightarrow (1,5,7) = (1,3,5)$ or $(3,5,7) = (1,3,5)$[13]
- $(1,3,4) \rightarrow (1,3,5)$
- $(1,3,5) \rightarrow (0,3,5) = (0,1,7)$
- $(1,4,7) \rightarrow (1,3,7) = (1,3,5)$ or $(1,5,7) = (1,3,5)$[14]
- $(0,1,3) \rightarrow (0,1,6)$ or $(0,1,6)$
- $(0,1,6) \rightarrow (0,2,6)$ or $(1,3,6) = (0,1,5)$ or $(0,2,6)$
- $(0,1,5) \rightarrow (0,1,8)$
- $(0,2,6) \rightarrow (2,3,6) = (0,1,8)$

[10] The robot on position 4 cannot distinguish the move to the position 3 or 5, but in both cases, the system arrives in the same configuration $(0,1,6)$.

[11] The robot on position 0 cannot distinguish the move to the position 1 or 3, but in both cases, the system arrives in the same configuration $(0,1,8)$.

[12] The robot on position 4 cannot distinguish the move to the position 1, 3, 5, or 7, but in all cases, the system arrives in the same configuration $(0,1,8)$.

[13] The robot on position 0 cannot distinguish the move to the position 1 or 3, but in both cases, the system arrives in the same configuration $(1,3,5)$.

[14] The robot on position 4 cannot distinguish the move to the position 3 or 5, but in both cases, the system arrives in the same configuration $(1,3,5)$.

## B Grids $2 \times m$ (with $m \geq 4$)

*Perpetual exploration (sub-problem 1):*

- Going "right"; $\forall x \in \{1, \ldots, m-2\}$:
  - $(0, x, m) \rightarrow (0, x+1, m)$
- Transition:
  - $(0, m-1, m) \rightarrow (1, m-1, m) = (0, m-2, 2m-1)$
  - $(0, m-2, 2m-1) \rightarrow (0, m-2, m-1) = (0, 1, m-1)$
  - $(0, 1, m-1) \rightarrow (0, 1, 2m-1)$
- Going "left"; $\forall x \in \{m+1, \ldots, 2m-1\}$:
  - $(0, 1, x) \rightarrow (0, 1, x-1)$

*Transient phase (sub-problem 2):*

- Breaking the symmetry of configurations of type 2 that exist only if $m$ is odd. Let define the middle position $M = \lfloor \frac{m}{2} \rfloor$:
  - $(M-1, M, M+1) \rightarrow (M-1, M+m, M+1)$
  - $(x, M, m-1-x) \rightarrow (x, M-1, m-1-x)$[15] $\quad \forall x \in \{0, \ldots, M-2\}$
  - $(x, M+m, m-1-x) \rightarrow (x, M+m-1, m-1-x)$[16] $\quad \forall x \in \{0, \ldots, M-1\}$
- Reaching one configuration used in the perpetual exploration:
  - $(0, 1, x) \rightarrow (0, 1, x+m) \quad \forall x \in \{2, \ldots, m-2\}$
  - $(0, x, y) \rightarrow (0, x-1, y) \quad \forall x \in \{2, \ldots, m-1\}$ and $\forall y \in \{x+1, 2m-1\}$
  - $(0, x, y) \rightarrow (0, x-m, y) \quad \forall x \in \{m+1, \ldots, 2m-3\}$[17] and $\forall y \in \{x+1, 2m-2\}$
  - $(x, y, z) \rightarrow (x-1, y, z) \quad \forall x, y, z$ such that it does not correspond to any other configuration (i.e. non-symmetric configurations with $x \geq 1$).

  Note that the last rule cannot create a symmetric configuration.

[15] The robot on position $M$ cannot distinguish the move to the position $M-1$ or $M+1$, but in both cases, the system arrives in the same configuration $(x, M-1, m-1-x)$.

[16] The robot on position $M+m$ cannot distinguish the move to the position $M+m-1$ or $M+m+1$, but in both cases, the system arrives in the same configuration $(x, M+m-1, m-1-x)$.

[17] The case $x = m$ is not considered because all configurations $(0, m, y)$ are part of the perpetual exploration.

## C Grids $n \times m$ (with $3 \leq n < m$)

*Perpetual exploration (sub-problem* 1*):* The first, third, and fourth rules correspond to the zigzag moves executed by the robot which explores all the grid but the first line and the two first columns.

- Going "down"; $\forall x \in \{1, \ldots, m-2\}$ and $\forall y \in \{1, \ldots, n-2\}$:
  - $(0, x, y \times m + x + 1) \rightarrow (0, x, (y+1) \times m + x + 1)$
- Going "left"; $\forall x \in \{2, \ldots, m-2\}$:
  - $(0, x, (n-1) \times m + x + 1) \rightarrow (0, x-1, (n-1) \times m + x + 1)$
- Going "up'; $\forall x \in \{2, \ldots, m-2\}$ and $\forall y \in \{1, \ldots, n-2\}$:
  - $(0, x-1, (y+1) \times m + x + 1) \rightarrow (0, x-1, y \times m + x + 1)$
- Going "left"; $\forall x \in \{2, \ldots, m-2\}$:
  - $(0, x-1, m + x + 1) \rightarrow (0, x-1, m + x)$
- End of the sequence:
  - $(0, 1, (n-1) \times m + 2) \rightarrow (0, 1, (n-1) \times m + 1)$
  - $(0, 1, (n-1) \times m + 1) \rightarrow (0, 1, (n-1) \times m)$
  - $(0, 1, (n-1) \times m) \rightarrow (1, m, (n-1) \times m) = (0, (n-2) \times m, (n-1) \times m + 1)$
  - $(0, x \times m, (n-1) \times m + 1) \rightarrow (0, (x-1) \times m, (n-1) \times m + 1) \quad \forall x \in \{2, \ldots, n-2\}$
  - $(0, m, (n-1) \times m + x) \rightarrow (0, m, (n-1) \times m + x + 1) \quad \forall x \in \{1, \ldots, m-2\}$
  - $(0, m, x \times m - 1) \rightarrow (0, m, (x-1) \times m - 1) \quad \forall x \in \{3, \ldots, n\}$
  - $(0, m, 2m-1) \rightarrow (0, m-1, m)$
  - $(0, m-1, m) \rightarrow (1, m-1, m) = (0, m-2, 2m-1)$

*Transient phase (sub-problem* 2*):* These moves are defined only for configurations that do not appear above. Indeed when the system arrives in one of the previous configurations, the transient phase is over and the perpetual exploration starts.

- Breaking the symmetry of configurations of type 3 that exist iif $m$ or $n$ is odd[18].
  - If the three robots are in adjacent position, move the one which is in the middle to the top/bottom/right/left to break the alignment.
  - If the three robots are not in adjacent position, move the one which is distinguishable to the top/bottom/right/left to break the symmetry.
- Moving one robot to position 0:
  - $(x, y, z) \rightarrow (x - m, y, z)$ if $x \geq m$
  - $(x, y, z) \rightarrow (x - 1, y, z)$ if $1 \leq x \leq m - 1$
- Moving one robot to position 1:
  - $(0, y, z) \rightarrow (0, y - m, z)$ if $y \geq m + 1$
  - $(0, m, z) \rightarrow (0, m + 1, z)$ if $y = m$
  - $(0, y, z) \rightarrow (0, y - 1, z)$ if $2 \leq y \leq m - 1$
- Moving one robot to position $(n-1)m$:
  - $(0, 1, z) \rightarrow (0, 1, z + m)$ if $z \leq (n-1)m$
  - $(0, 1, z) \rightarrow (0, 1, z - 1)$ if $z \geq (n-1)m + 1$

[18] We do not give the rules in a formal notation, because there are many special cases that would require special notations. We think that it is more easily understandable with this informal notation. Since no other rule considers configuration of type 3 there is no multiple rules for the same configuration.

## D Grids $n \times n$ (with $n \geq 4$)

*Perpetual exploration (sub-problem* 1*):* The first, third, and fourth rules correspond to the zigzag moves executed by the robot which explores all the grid but the first line and the two first columns.

- Going "down"; $\forall x \in \{1, \ldots, n-2\}$ and $\forall y \in \{1, \ldots, n-2\}$:
  - $(0, x, y \times n + x + 1) \rightarrow (0, x, (y+1) \times n + x + 1)$
- Going "left"; $\forall x \in \{2, \ldots, n-2\}$:
  - $(0, x, (n-1) \times n + x + 1) \rightarrow (0, x-1, (n-1) \times n + x + 1)$
- Going "up'; $\forall x \in \{2, \ldots, n-2\}$ and $\forall y \in \{1, \ldots, n-2\}$:
  - $(0, x-1, (y+1) \times n + x + 1) \rightarrow (0, x-1, y \times n + x + 1)$
- Going "left"; $\forall x \in \{2, \ldots, n-2\}$:
  - $(0, x-1, n + x + 1) \rightarrow (0, x-1, n + x)$
- End of the sequence ("left" twice, then "bottom" once):
  - $(0, 1, (n-1) \times n + 2) \rightarrow (0, 1, (n-1) \times n + 1)$
  - $(0, 1, (n-1) \times n + 1) \rightarrow (0, 1, (n-1) \times n)$
  - $(0, 1, (n-1) \times n) \rightarrow (1, n, (n-1) \times n) = (0, n-2, 2n-1)$

*Transient phase (sub-problem* 2*):* These moves are defined only for configurations that does not appear above. Indeed when the system arrives in one of the previous configurations, the transient phase is over and the perpetual exploration starts.

- Breaking the symmetry of symmetric configurations where robots are not aligned:
  - $(0, 1, n) \rightarrow (0, 1, 2n)$ or $(0, 1, 2n)$
  - $(0, 1, 2n) \rightarrow (0, 2, 2n)$
  - $(0, 1, 2n) \rightarrow (0, 2, 2n)$
  - $(0, x, x \times n) \rightarrow (1, x, x \times n) \quad \forall x \in \{2, \ldots, n-1\}$
  - $(1, x, x \times n) \rightarrow (1, x, x \times n + 1) \quad \forall x \in \{2, \ldots, n-1\}$
- Breaking the symmetry of symmetric configurations where robots are aligned:
  - If the three robots are in adjacent position, move the one which is in the middle to the top/bottom/right/left to break the alignment.
  - If the three robots are not in adjacent position, move the one which is distinguishable to the top/bottom/right/left to break the symmetry.
- Moving one robot to position 0:
  - $(x, y, z) \rightarrow (x-n, y, z)$ if $x \geq n$
  - $(x, y, z) \rightarrow (x-1, y, z)$ if $1 \leq x \leq n-1$[19]
- Moving one robot to position 1:
  - $(0, y, z) \rightarrow (0, y-n, z)$ if $y \geq n+1$ [20]
  - $(0, y, z) \rightarrow (0, y-1, z)$ if $2 \leq y \leq n-1$
- Moving one robot to position $(n-1)n$:
  - $(0, 1, z) \rightarrow (0, 1, z+n)$ if $z \leq (n-1)n$
  - $(0, 1, z) \rightarrow (0, 1, z-1)$ if $z \geq (n-1)n+1$

[19] This rule does not apply for configurations $(1, k, k \times n)$ with $2 \leq k \leq n-1$ since we already defined a specific rule before.

[20] Note that $y = n$ is not possible since it will be equivalent to a configuration $(0, 1, -)$.