Fast Collisionless Pattern Formation by Anonymous, Position-Aware Robots*

Tamás Lukovszki[1] and Friedhelm Meyer auf der Heide[2]

[1]Faculty of Informatics, Eötvös Loránd University, Budapest, Hungary
lukovszki@inf.elte.hu
[2]Heinz Nixdorf Institute and Department of Computer Science, University of Paderborn, Germany
fmadh@uni-paderborn.de

Abstract. We consider a scenario of n identical autonomous robots on a 2D grid. They are memoryless and do not communicate. Their initial configuration does not have to be connected. Each robot r knows its position $p_r \in \mathbf{Z}^2$. In addition, each robot knows the connected pattern F to be formed. F may be given by a set of n points in $\mathbf{Z}^2$, or may be only partially described, e.g., by "form a connected pattern", or "build a connected formation with minimum diameter" (Collisonless Gathering). We employ the Look-Compute-Move (LCM) model, and assume that in a time step each robot is able to move to an unoccupied neighboring grid vertex, thus guaranteeing that two robots will never collide, i.e., occupy the same position. The decision where to move solely depends on the configuration of its 2-hop neighborhood in the grid $\mathbf{Z}^2$.
First we consider a helpful intermediate problem - we call it the *Lemmings problem* - where collision at one single point g, known to all robots, is allowed and the goal is that all robots gather at g. We present an algorithm solving this problem in $2n + D - 1$ time steps, where D denotes the maximum initial distance from any robot to g. This time bound is easily shown to be optimal up to a constant factor.
Based on this strategy, forming a connected pattern can be done within the same time bound. Forming a connected pattern F needs additional considerations. We show how to do so in time $O(n + D^*)$, where D^* denotes the diameter of the point set consisting of the initial configuration and F. For Collisionless gathering we obtain the same time bound, up to constant factors. This significantly improves upon the previous upper bound of $O(nD)$ for this problem presented in [5].

Keywords: Autonomous mobile robots, pattern formation, gathering

1 Introduction

We consider various pattern formation problems by n identical autonomous robots on a 2D grid. They are memoryless (or use only $O(1)$ bits of persistent

* Supported by the German Research Foundation (DFG) within the Collaborative Research Center "On-The-Fly Computing" (SFB 901).

memory) and operate without explicit communication. They have computation and locomotion capabilities and limited visibility range. They are represented by discs of unit diameter. Each robot r knows its position $p_r \in \mathbf{Z}^2$ but not the position of the other robots. In addition, each robot knows the connected pattern F to be formed. F may be given by a set of n points in $\mathbf{Z}^2$, or as a predicate, e.g., "form a horizontal line segment", or may be only partially described, e.g., by "form a connected pattern", or "build a connected formation with minimum diameter" (Collisonless Gathering). All robots have a common coordinate system. Each robot has a visibility range of 2 units, i.e. it can see the robots within its local range of 2 units. With other words, the robots only have information about their 1- and 2-hop grid neighbors. The robots are able to move only on the edges of the grid. They all move synchronously with unit speed, s.t. they travel an edge of the grid in one time unit.

The robots operate corresponding to the *Look-Compute-Move* (LCM) model. In one cycle, a robot takes a snapshot of its current visibility range (Look), makes a decision to stay idle or to move to one of the neighboring vertices (Compute), and in the latter case makes an instantaneous move to this neighbor (Move). We assume that the LCM cycles are synchronous at each robot. Collisions are not allowed during the algoritms, i.e. in each time step each vertex of $\mathbf{Z}^2$ can be occupied by at most one robot. The motion ends when the robots form the connected pattern F. From now on we will use the terms node and robot interchangeably.

1.1 Our Contribution

First we consider a helpful intermediate *gathering* (or *point formation*) problem - we call it the *Lemmings problem* - where collision at one single point g, known to all robots, is allowed. The goal is that all robots gather at g. We consider *oblivious* robots, i.e. the robots do not remember results from any of the previous computations. We present an algorithm solving this problem, called x-y-routing, where the robots only need local knowledge about their 2-hop neighborhood in the grid $\mathbf{Z}^2$. We show that the x-y-routing method can be used to guarantee the gathering of all robots at g in $2n + D - 1$ time steps, where D is the maximum initial hop distance of a robot from g. We prove that this time bound is optimal up to a constant factor.

After this we investigate the gathering problem of n oblivious robots, where no collision is allowed at g and the robots have to form a connected configuration containing g. We show that the x-y-routing solves this problem in $n + D - 1$ time steps. This significantly improves the previous upper bound of $O(nD)$ on this problem presented in [5].

After this we consider *finite state* robots, i.e., the robots can use $O(1)$ bits of persistent memory for the computation. We show, how the set of n robots can be arranged to form a connected axis parallel line segment containing a given point g, known to all robots, in $3n + D + 3$ steps.

Finally, for finite state robots, we show how an arbitrary connected pattern F, known to all robots, can be formed in time $O(n + D^*)$, where D^* denotes the

diameter of the point set consisting of the initial configuration and F. In case when all robots know n, this solution can also be applied for solving the focused coverage problem on the 2D grid. This results in $O(n + D)$ covering time. If the number of robots n is not known for the robots, then best known upper bound on this problem is $O(S)$, presented in [2], where S is the sum of initial distances of the mobile sensors from g.

This paper is organized as follows. Section 2 gives an overview of related work. In Section 3 we describe the x-y-routing algorithm, which plays a key role in our solutions. In Section 4 we define the Lemmings problem, where all robots must be gathered at the given point g. We prove the lower bound of $\Omega(D + n)$ time steps on the running time of each discrete, synchronous algorithm solving this problem. After this we prove that the x-y-routing algorithm solves this problem in at most $2n + D - 1$ time steps. After this, in Section 5 we study the gathering problem, where no collision is allowed at g and the robots have to form a connected configuration containing g. We show that the x-y-routing solves this problem in $n+D-1$ time steps. In Section 6 we show how the set of n robots can be arranged to form a connected axis parallel line segment contaning a given point g in $3n+D+3$ time steps. In Section 7 we show, how a connected pattern F, known to all robots, can be formed in time $O(n + D^*)$, where D^* denotes the diameter of the point set consisting of the initial configuration and F. Section 8 summarizes the work.

2 Related Work

Cohen and Peleg [4] presented an asynchronous algorithm to gather oblivios robots at the center of gravity. Their algorithm uses the LCM (Look-Compute-Move) discrete cycle based model to move their robots. They mathematically proved upper and lower bounds on the convergence speed of their solution.

Cord-Landwehr et al. [6] described an easy-to-check property of target functions that guarantee convergence and gives upper time bounds. This property holds for the target function in [4] and improves the upper bound on the speed of the convergence.

Czyczowicz et al. [7] considered the gathering problem for few fat robots, where the robots are modeled by unit disks. The goal was to gather the robots, such that the union of the unit disks is connected at the end. Collisions of the robots are not allowed during the gathering. A main problem which had to be solved here is that the line of sight of a robot may be blocked by the extent of other robots.

Cord-Landwehr et al. [5] studied the problem of gathering mobile robots with an extent at a given position as dense as possible to form a disk of minimum radius around the gathering point. The authors present an algorithm for the continuous case and the discrete case, where the robots are moving on a grid. They prove an $O(nD)$ upper bound for the gathering time, where n is the number of robots and D is the distance of the farthest robot from the gathering

point. They empirically studied the continuous case, where in they report a few deadlock situations in the simulations.

For the gathering problem of mobile robots many different variants exist differing in levels of synchronization, computational power of the robots, memory, range of visibility, agreement on coordinate system. For a survey we refer to [3].

Another related problem in distributed robotics is the Pattern Formation problem, where a group of mobile robots have to form a desired geometric pattern. The pattern can be given as set of points in the plane (by their coordinates) or as a predicate (e.g. "form a circle"). A common requirement is that the robots have distinct initial positions and that the number of points in the pattern and the number of robots are the same. Suzuki and Yamashita [13] [14] investigated the question what kinds of patterns can be formed by a group of autonomous, anonymous and homogenous mobile robots that do not communicate, but they are able to observe each others movements. In [13] [14] the authors have shown that without agreeing a common coordinate system, a pattern can be formed if and only if it is purely symmetrical, i.e., a regular polygon (or a point), or a set of regular concentric polygons. They also have shown that by agreeing on a coordinate system, the robots can form any geometric pattern. Flocchini et al. [9] have shown that if each robot has a compass needle that indicates North (the compass needles are parallel), then any odd number of robots can form an arbitrary pattern, but an even number, in the worst case, cannot. If each robot has two independent compass needles, say North and East, then any set of robots can form any pattern. Pattern Formation by robots with limited visibility has been studied in [15].

A further related problem is the Focused Coverage self-deployment problem in mobile sensor networks, where an area with maximum radius around a Point of Interest (POI) must be covered without sensing holes. This problem was introduced by Li at al. [12], [10], [11]. They solved the problem by driving the mobile sensors along an equilateral triangle tessellation graph centered at the POI. They showed that their algorithms terminate in finite time. The convergence time has also been evaluated by simulations. Blázovics and Lukovszki [2] presented a collision free algorithm solving the focused coverage problem in $O(S)$ time, where S is the sum of initial distances of the mobile sensors from the POI. The theoretical results has been also validated by simulations.

Another related fundamental problem is the Filling problem (see [1]), in which a given region must be covered by robots. In this problem the robots are initially not in the region, they enter the the space one by one, from a point called "door". When a robot enters the door, it must disperse itself in the region. The goal is to cover the entire region. Barrameda et al. [1] have proven that the Filling problem can be solved with limited visibility, for any simple orthogonal space, i.e., a polygonal region without holes with sides either parallel or orthogonal, with a single door, by finite-state robots with a common coordinate system and common unit of distance in finite time.

For an excellent overview on distributed computing by mobile robots we refer to the the book by Flocchini et al. [8].

3 Collisionless routing towards a point g

Let V be a set of n robots placed at the vertices of the rectangular grid $\mathbb{Z}^2$ and $g \in \mathbb{Z}^2$ a gatheting vertex. Each robot knows its own position and the position of g. We assume that the gathering vertex g has coordinates $(0, 0)$. We use the synchronous Look-Compute-Move model, i.e. all robots robots perform the Look, Compute, and Move steps synchronously. In each time step, each robot is able to move to a neighboring vertex of the rectangular grid or it stays in place. We assume that each robot can see its local environment within two hops. Thus, when a robot r chooses a neigboring vertex p to which it wants to move, r sees all robots that are potentially able to move to p in the same time step. The knowedge about the 2-hop neighborhood makes possible to decide locally, which robot can move to a certain vertex, such that no collision occurs.

3.1 The x-y routing algorithm

Now we present the routing algorithm. In each time step each robot wants to decrease its hop distance to the gathering vertex g, such that it moves in x-direction until it has the same x-coordinate than g. After this, it moves towards g in y-direction until it reaches g. A path of a robot emerging in this way is called an *x-y-path*, which terminates in g. For each robot r, at time t let $nexthop(r, t)$ be the neighboring vertex of r on the x-y-path towards g. For a robot r at g, we define $nexthop(r, t) = g$. If in a time step t the vertex $p = nexthop(r, t)$ is occupied by another robot, then the robot r must stay in place. If in a time step t there are two (or more) robots r and r' with $nexthop(r, t) = nexthop(r', t) = p$, then the robot with smaller y-distance from the gathering vertex g has higher *priority*, if r and r' has the same y-distance from g, then the robot with greater x-coordinate has higher priority. The robot with highest priority, say r, is allowed to move to vertex p and the other robot(s) must stay in place. Fomally, let r and r' be two robots with coordinates (r_x, r_y) and (r'_x, r'_y), respectively, s.t. in time t $nexthop(r, t) = nexthop(r', t) = p$. Then $priority(r, p) > priority(r', p)$ $\iff |r_y| < |r'_y|$ or ($p = g$ and $r_y > 0$ and $r'_y < 0$) or ($|r_y| = |r'_y|$ and $r_x > 0$).

Algorithm 1 x-y-routing(r)

while r has not yet reached g **do**
  $p \leftarrow nexthop(r, t)$
  if p is unoccupied **and** $\nexists$ another robot r' with $nexthop(r', t) = p$, s.t. r' has higher priority than r **then**
    r moves to p
  else
    r stays in place
  end if
  $t \leftarrow t + 1$
end while

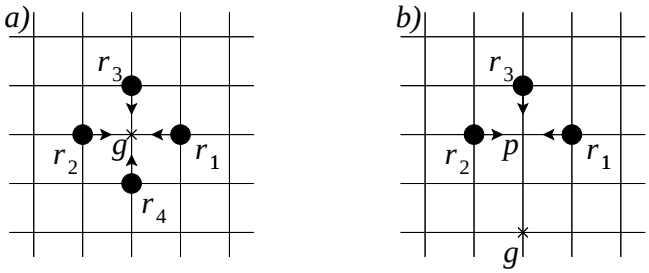

Fig. 1. Priorties of the robots having the same point p as $nexthop(.)$. *a*) If $p = g$, there are at most four robots r_1, r_2, r_3, r_4 having g as $nexthop(.)$. In this case $priority(r_1, g) > priority(r_2, g) > priority(r_3, g) > priority(r_4, g)$. *b*) If $p \neq g$, there are at most three robots r_1, r_2, r_3 having p as $nexthop(.)$. In this case $priority(r_1, g) > priority(r_2, g) > priority(r_3, g)$.

.

The above rule guarantees that, for each unoccupied vertex p, the robot which can occupy it in the next step – if any – is unique, and the robots are able to make this decision based on its local knowledge about their 2-hop neighborhood.

The x-y-routing algorithm is oblivious and the decision of each robot where to move solely depends on the configuration of the 2-hop neighborhood in the 2D grid.

4 The Lemmings problem

We assume that at the beginning the robots are placed on different vertices of the grid $\mathbb{Z}^2$. They try to move on the edges towards the gathering vertex g. The goal is to gather all the robots at g. Collision is only allowed at the gathering vertex g, i.e. we allow that g can be occupied by more than one robots at the same time. The only modification in Algoritm 1 is that Algoritm 1 treats the gathering vertex g as it would be always an unoccupied vertex. After performing the algorithm all robots will reside on vertex g.

4.1 Lower bound

First we show a lower bound of $\Omega(n + D)$ time steps on the Lemmings problem, where n robots must be gathered at g, where D is the maximum initial hop distance of a robot from g. We prove the lower bound for robots with infinite visibility range. Clearly, this bound also holds for robots with limited visibility.

Theorem 1. *Let $V = \{v_1, ..., v_n\}$ be the set of n robots with infinite visibility placed on different vertices of the grid $\mathbb{Z}^2$. Each algorithm, solving the synchronous discrete Lemmings problem needs $\Omega(n + D)$ time steps, where D is the maximum initial hop distance of a robot from g.*

Proof. Since each robot must arrive at g after performing the gathering algorithm and each robot can move to a neighboring grid vertex in each time step, at least D steps are necessary until the furthest robot arrives at g. On the other hand, in each time step only one robot can arrive at g from each direction. Therefore, each algorithm needs at least $n/4$ steps. In the case, when the initial distance of all robots from g is D, then the first robot arrives after D steps. In this step and in each further step at most four robot can arrive at g. Thus, the last robot needs at least $D + n/4 - 1 = \Omega(n + D)$ steps. Consequently, $\Omega(n + D)$ is a lower bound on the running time of each algorithm solving the problem. □

4.2 Upper bound

Now we turn to the analysis of Algorithm 1. The only modification in Algoritm 1 to solve the Lemmings problem is that Algoritm 1 treats the gathering vertex g as it would be always an unoccupied vertex.

First we consider the special but important case, where all robots are initially placed on one x-y-path P terminating in g. We show that after $n+D-1$ steps all robots reach g. We use this result for proving the time bound on the Lemmings problem in the general case.

Lemma 1. *Let $V = \{v_1, ..., v_n\}$ be the set of n oblivious robots placed on the same x-y-path $P \subset \mathbb{Z}^2$ terminating in g, such that all robots are placed on different vertices. Let D be the maximum initial hop distance of a robot from g, i.e. $D = \max_{r \in V}(d(r, g))$. Then by performing Algorithm 1 all robots reach g in $n + D - 1$ steps.*

Proof. We prove the claim by induction on the number of robots n.

For $n = 1$, the claim holds obviously, the robot v_1 gets strictly closer to g in each step, until it reaches g. Therfore, for $n = 1$, the algoritm terminates in $1 + D - 1 = D$ steps.

Now, assume that the claim holds for $n-1$ robots. Let $\{v_1, ..., v_n\}$ be the set of robots ordered by their initial hop distances from g, s.t. $d(v_1, g) < d(v_2, g) < ... < d(v_n, g)$. By the induction hypothesis v_{n-1} reaches g within $d(v_{n-1})+(n-1)-1 = d(v_{n-1})+n-2$ steps. Let $t_1, ..., t_k$ be the time steps in them v_{n-1} moved towards g. In time step t_k it reaches g. Then in time steps $t_1 + 1, ..., t_k + 1$ the robot v_n can also move towards g. In these time steps the distance between v_n and g decreases by $d(v_{n-1}, g)$ units. Therefore, after $t_k + 1$ steps the distance between v_n and g becomes at most $d(v_n, g) - d(v_{n-1}, g) = \delta$. In time steps $t > t_k$ the robot v_n can move towards g in each step. Therefore, v_n reaches g in $t_k + 1 + \delta$ steps. By the induction hypothesis, $t_k \leq d(v_{n-1}, g)+n-2$. Therefore, v_n reaches g within $d(v_{n-1}, g) + n - 2 + 1 + \delta = d(v_n, g) + n - 1$ steps, which proves the claim. □

Now we turn to the general case, where the n robots are arbitrarily placed on different vertices of $\mathbb{Z}^2$. We show that after $2n + D - 1$ steps all robots reach g.

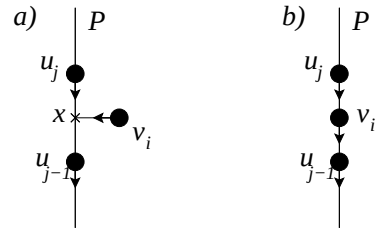

Fig. 2. Joining of robot v_i to the path P

.

Theorem 2. *Let $V = \{v_1, ..., v_n\}$ be the set of n oblivious robots placed on different vertices of $\mathbb{Z}^2$. Let g be the gathering vertex and D be the maximum initial hop distance of a robot from g, i.e. $D = \max_{r \in V}(d(r, g))$. Then by performing Algorithm 1 all robots reach g in $2n + D - 1$ steps.*

Proof. Let $v_1, v_2, ..., v_n$ be the robots ordered regarding the time they arrive at g. The x-y-path of each robot is unique, each such path terminates in g, and the union of the x-y-paths define a tree.

Consider the robot v_n arriving at g as the last one. Let P be the x-y-path from the initial position of v_n to g. Let $U = \{u_1, ..., u_k\}$ with $u_k = v_n$ be the set of robots that are initially on the tree path P. If we remove all robots in $V \setminus U$ from the scene, then by Lemma 1, all robots in U would reach g within $n + D' - 1$ steps, where D' is the initial distance of u_k from g.

Now consider the steps of v_n in the presence of the robots in $V \setminus U$. The key observation is that a robot $v_i \in V \setminus U$ can increase the arrival time of u_k at g by at most two time steps, that are (i) the step t_i, in which v_i reaches the path P, i.e., $nexthop(v_i, t_i) = x \in P$ and v_i moves to x, and (ii) the step immediately after v_i has joined the path P (see Figure 2):

Case (i): Consider the time step t_i, when the hop distance of v_i and P is one and after peforming this step v_i belongs to P (Figure 2.a). Let x be the vertex of P with $x = nexthop(v_i, t_i)$. If there is a robot $u_j \in P$ with $nexthop(u_j, t_i) = x$ with lower priority than v_i then u_j must wait in this step. This will increase the arriving time of v_n to g by one time unit.

Case (ii): The time step t_i+1, imediately after v_i has moved to $x \in P$ (Figure 2.b). In this time step u_j must wait because v_i is immediatly in front of u_j in P, i.e. v_i occupies the vertex $nexthop(u_j, t_i + 1)$. Starting with this time step, the robot v_i behaves exactly the same way as u_j would behave without the existence of v_i. Thus, v_i reaches g at the same time u_j would reach g without the existence of v_i. If no other robot joins the path P on the front of u_j, then u_j reaches g two time units after v_i reached g, i.e. at most two time units later than it would reach without the existence of v_i. The robots $v_{j'}$, $j' = j + 1, ..., k$, arrive at g by at most two time steps later than they would arrive without the existence of v_i.

Conequently, joining of all robots of $V \setminus U$ increases the arrival time of u_k by at most $2|V \setminus U| = 2(n - k)$ time units. Therefore, u_k arrives at g and the

algorithm terminates in at most $D' + k - 1 + 2(n-k) \leq D' + 2n - 1 \leq D + 2n - 1$ steps, where D is the maximum initial distance of a robot from g. □

Remark 1. By Theorem 1 and Theorem 2 the x-y-routing algorithm solves the discrete Lemmings problem, where all robots must be gathered at g in optimal time, up to a constant factor.

5 Forming a connected configuration containing g

Now we turn to the discrete gathering problem, where no collision is allowed at g and the robots have to form a connected configuration containing g. We show that the x-y-routing algorithm (Algoritm 1) solves this problem in $n + D - 1$ steps.

Theorem 3. *Let $V = \{v_1, ..., v_n\}$ be the set of n oblivious robots placed on different vertices of $\mathbb{Z}^2$. Let g be the gathering vertex and D be the maximum initial hop distance of a robot from g, i.e. $D = \max_{r \in V}(d(r, g))$. Then by performing Algorithm 1 the robots form a connected configuration containing g in $n + D - 1$ steps.*

Proof. For each robot $v \in V$, let p_v be the initial position of v and P_v the x-y-path from p_v to g. Let $T = \bigcup_{v \in V} P_v$ be the tree defined as the union of the x-y-paths. For each robot $v \in V$, let p_v^* be the closest position of v to g during the algorithm. Since the hop distance of a robot never increases during the algorithm, p_v^* is the final position of v. Let P_v^* the x-y-path from p_v^* to g and $T^* = \bigcup_{v \in V} P_v^*$.

We show that (i) T^* contains g, (ii) T^* has no unoccupied vertex, i.e. the robots form a connected configuration, and (iii) each vertex of T^* becomes occupied in $n + D - 1$ steps.

(i) Obviously, T^* contains g. A robot v with smallest initial distance form g will occupy g, which will be the final position v.

(ii) Assume for contradiction, that T^* has an unoccupied vertex. Let x be an unoccupied vertex of T^* with maximum hop distance from g. Let $U = \{v \in V : x \in P_v\}$ be the set of robots whose x-y-paths contain x. Then the final position of at least one of the robots in U is further from g than x, otherwise x would not be contained in T^*. Since x is an unoccupied vertex of T^* with maximum hop distance from g, there is at least one robot $u \in U$ with distance one from x, s.t. $d(p_u^*, g) = d(x, g) + 1$. Then one of these robots occupies x in the next step.

(iii) Now we show that each vertex of T^* becomes occupied in $n + D - 1$ steps. By induction, we prove that after $i + D$ steps, $0 \leq i \leq n - 1$, all vertices of T^* with hop distance at most i from g are occupied. Thus, after $i + D$ steps, the robots on those vertices have reached their final positions.

For $i = 0$, the claim holds, since the robot which occupies g will never be stopped by another robot and its initial distance from g is at most D. Therefore, after D steps it occupies g.

Assume that the induction hypothesis holds for i, $0 \leq i < n-1$. Let V_i be the set of robots with final position of hop distance at most i from g. If $V_i = V$ we are done. Otherwise, consider a robot v whose final position will be at a vertex x with distance $i+1$ from g.

We say that a robot u stops v in a certain step t if either $nexthop(v,t)$ is occupied by u or $nexthop(v,t)$ is unoccupied and u occupies it in step t. Observe, that during the algorithm none of the robots in $V \setminus V_i$ can stop v before v reaches its final position. To see this, assume that a robot $w \in V \setminus V_i$ stops v in a certain step t. Then in step $t+1$ the robot w becomes on the front of v in the x-y-path from v to g, i.e. w gets strictly closer to g than v and the distance of w to g remains strictly lower than the distance of v. Therfore, the final distance of w from g will also be strictly lower than the final distance of v, which is by assumption $i+1$.

Therefore, v can be only stopped by robots in V_i before v reaches its final position. Let u be the robot whose final position q is on the x-y-path of v to g and the hop distance $d(q,g) = i$. Let t be the time step in which u reaches q. By the induction hypothesis $t \leq i + D$. If v was stopped by u in some time step $t' < t$, then after this time step v "follows" u, and thus, in time step $t+1 \leq D+i+1$ the robot v also reaches its final position. If v was never stopped by u then v was able to get closer to g in each of the t steps. Since the distance between the initial position of v and its final position x is at most D, the robot v reaches x in at most $D \leq D+i+1$ steps. This completes the proof of the the induction hypothesis for $i+1$.

Since the hop distance between g and any vertex of T^* is at most $n-1$, each robot reaches its final position within $n+D-1$ steps and the claim of the theorem follows. □

6 Forming an axis parallel line segment containing g as an end point

Given a point g, known for each robot. The goal is to arrange the n robots in a connected horizontal line segment containing g as an end point. Now we consider so called finite state robots, i.e. with $O(1)$ persistent bits of memory (see e.g. in [1]). The visibility range of the robots is limited to the 2-hop neighborhood in the 2D grid. We show how the robots can form a connected horizontal line L containing g as its left end point in $3n + D + 3$ steps. A vertical line segment can be formed in a similar way.

6.1 Forming the horizontal line segment L

Forming the connected horizontal line segment L containing g as its left end point consists of 3 phases for each robot. Each of the 3 phases are oblivious, in each step the decision of each robot where to move solely depends on the configuration of the 2-hop neighborhood in the 2D grid. The only persistent memory used by a robot is to store, which phase of the algorithm it currently executes. The robots execute the following phases:

1. Let (g_x, g_y) be the coordinates of g and let g' be the point with coordinates $(g_x - 1, g_y + 1)$. Each robot with initial y-coordinate greater than g_y moves one step upwards and each robot with initial y-coordinate less or equal than g_y moves one step downwards. During this step no collision can arrise. At the end of this step we obtain a horizontal stripe H of height 2 containing the horizontal line coincident with g and the horizontal line coincident with g'.
2. Execute the Lemmings algorithm with sink g'. More precisely, execute Algorithm 1 with sink g', such that g' can be occupied in a step t, if g' is unoccupied at the beginning of step t. When a robot occupies g' in a certain step, it moves one hop to the right from g' to the point $g'' = (g_x, g_y + 1)$ in the next step and starts phase 3.
3. Build L from the source g'' as follows. Let (x, y) be the current coordinates of a robot. Until the the vertex at $(x, y - 1)$ is occupied, move to the right. Otherwise, occupy the vertex at $(x, y-1)$ and terminate the algoritm of that robot.

Theorem 4. *Let $V = \{v_1, ..., v_n\}$ be the set of n finite state robots placed on different vertices of $\mathbb{Z}^2$. Let g be a point, known for all robots. Then by the above algorithm the robots form a connected horizontal line segment L with left end point g in $D + 3n + 3$ steps, where D is the maximum initial hop distance of a robot from g.*

Proof. After phase 1, the horizontal stripe H of height 2 containing the horizontal lines coincident with g and g' is free of robots. Let ℓ be the vertical line coincident to g'. In phase 2, non of the robots visits any vertex in $H \setminus \ell$. In phase 3 each robot only visits vertices in $H \setminus \ell$. Therefore, no collision can occure.

Phase 1 of the algorithm takes 1 time unit. The only difference between phase 2 of the formation of L and the Lemmings algorithm is that g' can be occupied in a step t, if g' is unoccupied at the beginning of step t. The robot, which has occupied g' will move away from g to the left in the next step. and starts phase 3. It is easy to check that all arguments of the proof of Theorem 2 also apply for this case and the time bound $2n + D' - 1$ stated in Theorem holds, where $D' = D + 2$ is the maximum hop distance of a robot from g' at the beginning of phase 2. Therefore, each robot finishes phase 1 and 2 after $2n + D + 3$ steps. It is easy to check that each robot spends at most n time steps in phase 3. Consequently, the robots build the connected line segment L by the algorithm in $3n + D + 3$ steps.

7 Forming an arbitrary connected pattern F

Now we show how the Lemmings algorithm can be used for forming an arbitrary connected pattern of n vertices in the 2D grid. We consider finite state robots. The visibility range of the robots is limited to the 2-hop neighborhood in the 2D grid.

Given a connected pattern F of size n. Let B be the axis-parallel bounding box of F and (x, y) the upper left corner of B. Let g be the point with coordinates

$(x-1, y+1)$ and g' be the point with coordinates $(x, y+1)$ (see Figure 3). Let T be a spanning tree of F. Consider the rooted tree T^* with root g that starts with a shortest path from g through g' to a closest node g^* of F (This path does not contain further nodes of F) and contains a rooted version of T rooted at g^*.

Assume that n robots appear in g, one after another. Empty steps where no new robot appears are allowed, all robots have appeared after some number R of steps. The point g plays a similar role as the "door" in the Filling problem [1], but g is not contained in F. As further difference, F can be any connected pattern in our case. Let $d(T^*)$ be the depth of T^*. It is easy to check that a dept-first filling of T^* (more precisely, the depth-first filling of T) can be executed by the appearing robots, so that, after $O(R + d(T^*))$ many steps, each node of F is occupied by one robot, and no collisions appeared during these steps.

7.1 Forming F

Forming the connected pattern F consists of 3 phases for each robot. The only persistent memory used by a robot is to store, which phase of the algorithm it currently executes. The robots execute the following phases:

1. Let y_{min} and y_{max} be the minimum and maximum y-coordinate of F, respectively. Let $h = y_{max} - y_{min} + 1$ be the height of the connected pattern F. Each robot with initial position above or on the lower horizontal border of F moves $h + 1$ many steps upwards. During this $h + 1$ steps there are no collisions. At the end we obtain a horizontal stripe H of height $h + 1$ containing the axis parallel bounding box B' of $F \cup \{g'\}$ free of robots (see Figure 3).
2. Execute the Lemmings algorithm with sink g. More precisely, execute Algorithm 1, such that g can be occupied in a step t, if it is unoccupied at the beginning of step t. When a robot occupies g in a certain step, it moves one hop to the right from g to the point g' in the next step and starts phase 3.
3. Build F from the source g' by depth-first filling of T, using the arrivals of the robots in g during the Lemmings algorithm as input stream.

Theorem 5. *Let $V = \{v_1, ..., v_n\}$ be the set of n finite state robots placed on different vertices of $\mathbb{Z}^2$. Let F be a connected formation, known for all robots. Then the robots form F in time $O(n + D^*)$, where D^* denotes the diameter of the point set consisting of the initial configuration and F.*

Proof. After phase 1 of the algorithm, the horizontal stripe H of height $h + 1$ containing the axis parallel bounding box B' of $F \cup \{g'\}$ is free of robots. In phase 2, each robot only visits vertices outside B'. In phase 3 each robot only visits vertices in B'. Therfore, no collision can occure.

Phase 1 takes $h + 1 = O(D^*)$ steps. The only difference between phase 2 of the pattern formation and the Lemmings algorithm is that g can be occupied in a step t, if it is unoccupied at the beginning of step t. When a robot occupies g in a certain step, it moves one hop to the right from g to the point g' in

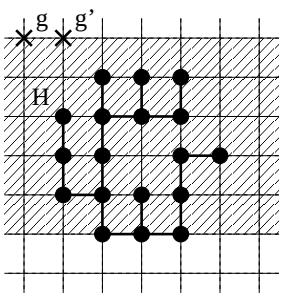

Fig. 3. After phase 1, the horizontal stripe H (shaded region) of height $h+1$ containing the axis parallel bounding box B' of $F \cup \{g'\}$ is free of robots. In phase 2, each robot only visits vertices outside B'. In phase 3 each robot only visits vertices in B'.

the next step and starts phase 3. It is easy to check that all arguments of the proof of Theorem 2 also apply for this case and the time bound $2n + D' - 1$ stated in Theorem holds, where D' is the maximum hop distance of a robot from g at the beginning of phase 2. Therefore, each robot finishes phase 2 after $O(n + D^*)$ steps. It is easy to check that each robot spends $O(n)$ time in phase 3. Consequently, the formation F becomes built by the algorithm in $O(n + D^*)$ steps.

7.2 Focused coverage problem, when n is known for all nodes

A closely related problem to the pattern formation problem is the focused coverage self-deployment problem in mobile sensor networks, where an area with maximum radius around a Point of Interest (POI) must be covered without sensing holes. This problem was introduced in [10], [11],[12]. The authors solved the problem by driving the mobile sensors along an equilateral triangle tessellation graph centered at the POI. They showed that their algorithms terminate in finite time. Subsequently, in [2] a collision free algorithm has been presented solving the focused coverage problem in $O(S)$ time, where S is the sum of initial distances of the mobile sensors from the POI.

Now we consider the focused coverage problem on the 2D grid instead of the an equilateral triangle tessellation graph. Additionally, we assume that the number of sensors n is known for each sensor node. Then each node can compute the connected hole free formation F with maximum radius centered at the POI. Then our algorithm for connected pattern formation can be applied, which solves the problem in $O(n + D^*)$ time, where D^* denotes the diameter of the point set consisting of the initial configuration and F.

Corollary 1. *Let $V = \{v_1, ..., v_n\}$ be the set of n finite state mobile sensor nodes placed on different vertices of $\mathbb{Z}^2$. Assume that all nodes know the POI, n, and the configutation of its 2-hop neighborhood in $\mathbb{Z}^2$. Then by applying the connected pattern formation algorithm, the focused coverage problem can be solved in time*

$O(n + D^*)$, where D^* denotes the diameter of the point set consisting of the initial configuration of the nodes and the POI.

8 Summary

We have investigated the Pattern Formation problem on a 2D grid in the synchronous Look-Compute-Move model. First we have considered the helpful intermediate problem, called the Lemmings problem, where all robots have to be gathered at a given point g, and thus at the single point g collision is allowed, we proved a lower bound of $\Omega(n + D)$ time steps on the running time of each discrete synchronous algorithm solving this problem.

We have introduced the x-y-routing algorithm for solving this problem, where the nodes only need local knowledge about their 2-hop neighborhood in the grid $\mathbf{Z}^2$. Based on this knowledge the nodes are able to move towards the gathering point g without collision, such that after at most $2n + D - 1$ time steps all robots reach g, where D is the maximum hop distance of a robot from g. Thus, the running time of the algorithm is optimal up to a constant factor.

We have shown that the x-y-routing method also solves the gathering problem in $n + D - 1$ time steps, where no collision is allowed at g and the robots have to form a connected configuration containing g. This significantly improves the previous upper bound of $O(nD)$ for this problem presented in [5]

Furthermore, we have shown how a the robots can form a connected axis parallel line segment L containing g as an end point in $3n + D + 3$ steps.

Finally, we have investigated the problem of forming a connected pattern F, which is known to all robots. We have shown how to build F in time $O(n + D^*)$, where D^* denotes the diameter of the point set consisting of the initial configuration and F.

In case when all robots know n, this solution can also be applied for solving the focused coverage problem on the 2D grid. This results in $O(n + D)$ covering time. If the number of robots n is not known for the robots, then best known upper bound on this problem is $O(S)$, presented in [2], where S is the sum of initial distances of the mobile sensors from g.

References

1. Barrameda, E.M., Das, S., Santoro, N.: Deployment of asynchronous robotic sensors in unknown orthogonal environments. In: Algorithmic Aspects of Wireless Sensor Networks, Fourth International Workshop, (ALGOSENSORS), Revised Selected Papers. pp. 125–140. Springer, LNCS Vol. 5389 (2008)
2. Blázovics, L., Lukovszki, T.: Fast localized sensor self-deployment for focused coverage. In: Proc. 9th International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics (ALGOSENSORS 2013). pp. 83–94. Springer, LNCS Vol. 8243 (2014)
3. Cieliebak, M., Flocchini, P., Prencipe, G., Santoro, N.: Distributed computing by mobile robots: Gathering. SIAM J. Comput. 41(4), 829–879 (2012)

4. Cohen, R., Peleg, D.: Convergence properties of the gravitational algorithm in asynchronous robot systems. SIAM J. Comput. 34(6), 1516–1528 (Jun 2005)
5. Cord-Landwehr, A., Degener, B., Fischer, M., Hüllmann, M., Kempkes, B., Klaas, A., Kling, P., Kurras, S., Märtens, M., Meyer auf der Heide, F., Raupach, C., Swierkot, K., Warner, D., Weddemann, C., Wonisch, D.: Collisionless gathering of robots with an extent. In: Proceedings of the 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2011). pp. 178–189. Springer, LNCS Vol. 6543 (2011)
6. Cord-Landwehr, A., Degener, B., Fischer, M., Hüllmann, M., Kempkes, B., Klaas, A., Kling, P., Kurras, S., Märtens, M., Meyer auf der Heide, F., Raupach, C., Swierkot, K., Warner, D., Weddemann, C., Wonisch, D.: A new approach for analyzing convergence algorithms for mobile robots. In: Proc. 38th International Colloquium on Automata, Languages and Programming (ICALP 2011). pp. 650–661. Springer, LNCS Vol. 6756 (2011)
7. Czyzowicz, J., Gasieniec, L., Pelc, A.: Gathering few fat mobile robots in the plane. Theor. Comput. Sci. 410(6-7), 481–499 (2009)
8. Flocchini, P., Prencipe, G., Santoro, N.: Distributed Computing by Oblivious Mobile Robots. Synthesis Lectures on Distributed Computing Theory, Morgan & Claypool Publishers (2012)
9. 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)
10. Li, X., Frey, H., Santoro, N., Stojmenovic, I.: Localized sensor self-deployment for guaranteed coverage radius maximization. In: Communications, 2009. ICC '09. IEEE International Conference on. pp. 1 –5 (2009)
11. Li, X., Frey, H., Santoro, N., Stojmenovic, I.: Focused-coverage by mobile sensor networks. In: 6th IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS). pp. 466–475 (2009)
12. Li, X., Frey, H., Santoro, N., Stojmenovic, I.: Strictly localized sensor self-deployment for optimal focused coverage. IEEE Trans. Mob. Comput. 10(11), 1520–1533 (2011)
13. Suzuki, I., Yamashita, M.: Distributed anonymous mobile robots. In: Proc. 3rd International Colloquium on Structural Information and Communication Complexity (SIROCCO). pp. 313–330 (1996)
14. Suzuki, I., Yamashita, M.: Distributed anonymous mobile robots: Formation of geometric patterns. SIAM J. Comput. 28(4), 1347–1363 (1999)
15. Yamauchi, Y., Yamashita, M.: Pattern formation by mobile robots with limited visibility. In: 20th International Colloquium on Structural Information and Communication Complexity (SIROCCO), Revised Selected Papers. pp. 201–212. Springer, LNCS Vol. 8179 (2013)