Journal of Combinatorial Optimization (2020) 39:988–1016
https://doi.org/10.1007/s10878-020-00533-z

On the robustness of a synchronized multi-robot system

Sergey Bereg[1] · Andrew Brunner[2] · Luis-Evaristo Caraballo[3] · José-Miguel Díaz-Báñez[3] · Mario A. Lopez[2]

Published online: 7 February 2020

Abstract

Area coverage and communication are fundamental concerns in networks of cooperating robots. The goal is to address the issue of how well a group of collaborating robots having a limited communication range is able to monitor a given geographical space. Typically, an area of interest is partitioned into smaller subareas, with each robot in charge of a given subarea. This gives rise to a communication network that allows robots to exchange information when they are sufficiently close to each other. To be effective, the system must be resilient, i.e., be able to recover from robot failures. In a recent paper Bereg et al. (J Comb Optim 36(2):365–391, 2018), the concept of *k-resilience* of a synchronized system was introduced as the cardinality of a smallest set of robots whose failure suffices to cause that at least k surviving robots operate without communication, thus entering a state of *starvation*. It was proven that the problem of computing the k-resilience is NP-hard in general. In this paper, we study several problems related to the resilience of a synchronized system with respect to coverage and communication on realistic topologies including grid and cycle configurations. The *broadcasting resilience* is the minimum number of robots whose removal may disconnect the network. The *coverage resilience* is the minimum number of robots whose removal may result in a non-covered subarea. We prove that the three resilience measures can be efficiently computed for these configurations.

Keywords Synchronization · Resilience · Multi-robot system · Coverage · Isolation · Broadcast · Connectivity

L. E. Caraballo: Funded by Spanish Government under Grant Agreement FPU14/04705.

This research has received funding from the projects GALGO (Spanish Ministry of Economy and Competitiveness, MTM2016-76272-R AEI/FEDER, UE) and CONNECT (European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 734922).

Extended author information available on the last page of the article

1 Introduction

Interest in systems of autonomous robots that perform tasks cooperatively has been on the rise in recent years (Abdulla et al. 2014; Alena et al. 2018; Mazayev et al. 2016). In many cases the cooperation between robots is a crucial requirement for the success of the system. A common scenario involves the completion of a task with each robot being assigned a certain subtask. In this case, if one or more robots fail, the remaining robots are expected to assume the pending subtasks. Even if the risk of failure among the members of the team is low, cooperation could be fundamental, for example, in monitoring missions.

There is a vast literature on optimization problems motivated by problems in robotics and mobile sensors. Algorithmic and combinatorial problems have been inspired by infrastructure security Kranakis and Krizanc (2015), area coverage Choset (2001), scheduling Hwang and Cheng (2001) or robotic assembly Hamacher (1992), to name a few.

In previous work, a combinatorial scheduling problem has been addressed in Díaz-Báñez et al. (2017) and Díaz-Báñez et al. (2015) for monitoring a terrain using a system of synchronized aerial robots (UAVs) with communication constraints. Later, motivated by the protocol proposed in those papers, an optimization problem related to the robustness of the system with respect to failures was studied Bereg et al. (2018). The *k-resilience* of the system, was defined as the cardinality of a smallest set of robots whose failure results in at least k surviving robots operating without communication as they continue to follow the proposed protocol. In Bereg et al. (2018), it was shown that the problem of computing the *k-resilience* is NP-hard when k is part of the input, but can be computed efficiently when $k = 1$. In this paper, we give additional results for this measure and other robustness metrics when the partition of the terrain allows one to model the problem using grids or cycles as the underlying graphs, a common configuration in many real life scenarios while patrolling big areas or boundaries such as coastlines.

1.1 A synchronized system

Consider a team of n mobile robots performing a task while each of them periodically travels along a predetermined closed trajectory (the trajectories are pairwise disjoint). The range of the communication interfaces of the robots is very small compared to the workspace area. Thus, two robots can establish a communication link and exchange information only when they are close enough to each other. Díaz-Báñez et al. (2015) and Díaz-Báñez et al. (2017) address the problem of maintaining periodic communication between robots while each performs its assigned task by moving along its assigned trajectory. The authors study an abstraction of the problem on a simpler scenario where the trajectories are pairwise disjoint circles of unit radius, a tour of a robot in a circle takes one time unit and the communication range of each robot is a value $\epsilon < 0.5$. We say that two circles C_i and C_j are *close* if the distance between their centers is less than $2 + \epsilon$ and we denote the segment connecting their centers by $\{i, j\}$. When two robots u and u' traversing C_i and C_j, respectively, lie on $\{i, j\}$ at

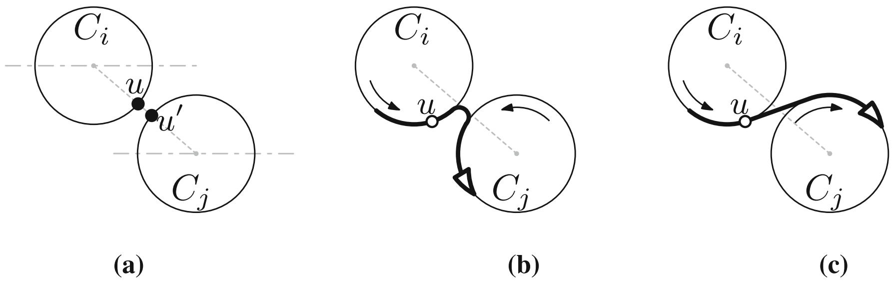

Fig. 1 **a** Two robots u and u' at the link position between close trajectories. **b** Shifting operation between trajectories with the same travel direction. **c** Shifting operation between trajectories with opposite travel directions

the same time, they can exchange information (they are within communication range of each other). In this case, we say that u and u' are in *link position* (see Fig. 1a). Moreover, since the robots travel along their trajectories at the same constant speed of 2π length per time unit, they are *synchronized* and can exchange information once per time unit.

Suppose now that we have set up a system with one robot per circular trajectory such that every pair of robots in close circles are synchronized. The performance of this system is compromised when a robot abandons the mission to refuel or due to some technical failure. Díaz-Báñez et al. (2015) and Díaz-Báñez et al. (2017) also address this problem and propose the following strategy: let ℓ be a link between close trajectories C_i and C_j with synchronized robots u and u', respectively. Suppose that u' in C_j abandons the system. When u in C_i arrives at ℓ, its communication interface detects the absence of u'. To compensate, u assumes the subtask assigned to u' in C_j (which is interrupted at the moment), leaving C_i and shifting to C_j (see Fig. 1b, c). This approach is referred as the *shifting strategy* in Díaz-Báñez et al. (2017). Notice that, due to kinematic constraints while shifting, it is convenient that synchronized robots fly in opposite directions, one in clockwise and the other one in counterclockwise direction (see Fig. 1).

Let $T = \{C_1, \ldots, C_n\}$ be set of unit circles (trajectories) such that C_i and C_j are disjoint for all $i \neq j$. Let $\epsilon < 0.5$ be the communication range of the robots. Let $G_\epsilon(T) = (V, E_\epsilon)$ be a graph whose nodes are the centers of the circles in T and whose edges are the pairs of circle centers at distance $2 + \epsilon$ or less. From now on assume that $G_\epsilon(T)$ is connected, as otherwise there cannot be full communication in the system. As proposed in Díaz-Báñez et al. (2015) and Díaz-Báñez et al. (2017), a *synchronized communication system (SCS)* is a bipartite connected graph $G = (V, E)$, a spanning subgraph of $G_\epsilon(T)$, such that $\{i, j\} \in E$ if and only if the robots in C_i and C_j are synchronized and fly in opposite directions. Every robot in a SCS applies the shifting strategy if it detects the absence of a neighbor in a link position. The graph G is called the *communication graph* of the SCS and two trajectories C_i and C_j are *neighboring* if $\{i, j\} \in E$ (see Fig. 2). Note that while G must be connected and bipartite in order to enforce the desired direction of travel, $G_\epsilon(T)$ just needs to be connected.

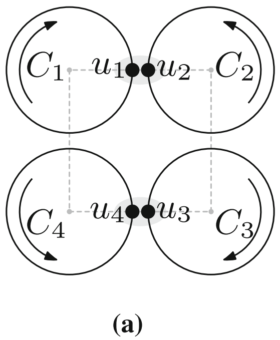

Fig. 2 **a**, **b** are two different states of a SCS formed by four trajectories. The communication graph is drawn using gray dotted lines. The arrow-arcs represent the travel direction of every trajectory. Notice that every pair of neighboring trajectories have opposite travel directions and every pair of robots in neighboring trajectories are synchronized

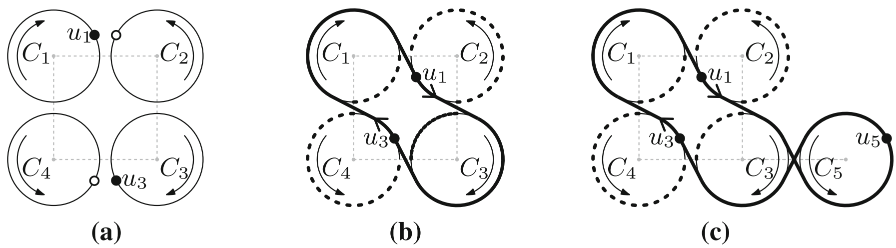

Fig. 3 **a**, **b** When the robots represented by hollow points leave the system then the surviving ones, solid points, follow the closed path drawn with bold solid stroke. The trajectory segments in dotted stroke in (**b**) and (**c**) are non-covered. In **b** u_1 and u_3 are starving, however, in **c**, u_1 and u_3 are not starving, they meet u_5 periodically. The message broadcasting is possible in (**c**) but it is not in (**b**)

1.2 Combinatorial problems and results

A robot is in a state of *starvation* if, while following the travel protocol, it fails to encounter any other robots at the designated link positions, thus altogether losing its ability to communicate. The concept of *k-resilience* of a SCS, introduced in Bereg et al. (2018), asks for the smallest set of robots whose failure suffices to cause the starvation of at least k surviving robots. Observe in Fig. 3a, b, that if the robots represented by hollow points leave the system, then the surviving ones, u_1 and u_3, starve, permanently failing to meet other robots.[1] It was shown in Bereg et al. (2018) that the problem of computing the k-resilience is NP-hard, even if the communication graph is a tree. In this paper we refer to this measure as the *k-isolation resilience* and prove that, for some configurations of the communication graph (cycles and grids), its value can be computed efficiently.

A desirable property of a communication network is the ability for messages broadcasting. We say that there is a *loss of connectivity* in a system if only one robot survives or there is a pair of robots that cannot exchange messages through a sequence of mes-

[1] An illustration of this phenomenon is at https://www.youtube.com/watch?v=64gKnefnXew.

Table 1 Complexities for computing the robustness measures

Measures	Tree	Cycle	$N \times M$ grid	General
Coverage resilience	$O(1)$	$O(n)$	$O(T_{gcd}(N, M))$	$O(n)$
k-isolation resilience	**NP-Hard**	$O(n)$	$O(1)$	**NP-Hard**
Broadcasting resilience	$O(n^{3/2})$	$O(n)$	$O(1)$	$O(n\kappa \cdot \min\{\kappa^3 + n, n\kappa\})$

Results in bold text were proved in Bereg et al. (2018)

sage exchanges between neighboring robots. For example, in Fig. 3b, a broadcast is not possible because the living robots are isolated. It is easy to see that, even without isolated robots, the system may not allow broadcasting, as the surviving robots are partitioned into independent connected components, for communication purposes. In this paper, we introduce a new robustness measure: the *broadcasting resilience* is the minimum number of robots whose removal causes a loss of connectivity in the system. We will show how to efficiently compute this measure for trees, cycles and grids.

Next, we focus on covering. Notice that in a system with one robot per trajectory, every point of every trajectory is visited by some robot periodically. We say that every trajectory point is *covered*. If some robots leave the system and the remaining ones stay in their trajectories (ignoring the shifting strategy altogether) then, obviously all the points of the trajectories of the leaving robots are *non-covered*. Using the shifting strategy one may think that the covering is guaranteed. However, this is not true. Sometimes, the departure of a set robots (independent of whether it causes starvation or not among the active robots) results in some trajectory segments (set of consecutive points of a trajectory), or even entire trajectories, to no longer be visited by an active robot. In this case we say that these are *non-covered* trajectory segments, see Fig. 3b, c. In this paper we introduce the notion of *resilience of a covered* synchronized system (*coverage resilience* for short) as the minimum number of robots whose removal may result in at least one non-covered trajectory segment. Thus, another objective of this work is to study the problem of computing the coverage resilience in a synchronized system. We will prove that this problem can be solved in linear time for any communication graph and, with a closed form solution for trees and grids. Table 1 summarizes our results.

1.3 Related work

The three resilience measures are of interest for both ground and aerial robots. Also, they are related to the most commonly used measures in robotics and ad-hoc mobile networks. The k-isolation resilience and the broadcasting resilience are related to the *meeting-time* (the maximum amount of time for two robots to communicate), *broadcast-time* (maximum time for the dissemination of a message) and *rendezvous* (the arrangement of two robots to meet). Allowing two robots to rendezvous so that they can collaboratively explore an unknown environment has been widely considered in robotics, see for example Flocchini et al. (2016) and Roy and Dudek (2001). Broadcasting and meeting time have been studied in mobile networks, robotics and random walks, see for example Clark et al. (2003), Hsieh et al. (2008), Lovász (1993) and Winfield (2000).

On the other hand, the coverage resilience is related to the *idle-time* of a point p in a terrain, that is, the maximum time that p is unattended by any of the robots. Notice that the maximum number of robots that can fail so that the idle-time of all points is finite is the value of the coverage resilience minus one. The problem of monitoring a region to minimize idleness is studied in the mobile robot literature under the name patrolling. Patrolling has been considered intensively in robotics where it is often viewed as a form of coverage. It is defined as the act of surveillance by walking around an area in order to protect or supervise it. The frequency of visits as a criterion for measuring the efficiency of patrolling is called idleness. For a survey of diverse approaches to patrolling based on idleness criteria we refer the reader to Alena et al. (2018) and Almeida et al. (2004).

As we will show, the values of the resilience stated in this paper depend on the topology of the communication graph. This graph is computed after the *partition strategy*, in which the environment is partitioned into sections patrolled separately by individual robots, has been applied. Partitioning or *area decomposition* for path planning in robotics is a widely researched subject in coverage and tracking tasks and a common topology is the grid (Galceran and Carreras 2013). In grid based methods, the area partitioning is performed by applying a grid overlay on top of the area leading to a discrete configuration space, where if all the cells are visited, then a complete coverage is assumed (see Fig. 4). Additionally, cycles are considered in boundary or fence patrolling (Czyzowicz et al. 2011). A strategy is to fragment the boundary into sections which are patrolled separately by individual robots (Collins et al. 2013) (see Fig. 5). Other studied configurations are trees. In fact, in area coverage, often it is assumed that the underlying graph is a tree (computed, for example, on the dual graph of a triangulation of the terrain). Spanning trees have been frequently used for multi-robot coverage (Hazon and Kaminka 2008) and boundary patrolling (Czyzowicz et al. 2016). Also note that, if we consider the underlying graph G of the area decomposition whose vertices are cells and whose edges are the pairs of close cells then, if G is connected, a spanning tree of G can be used as communication graph because any tree is bipartite and synchronizable (Díaz-Báñez et al. 2015, 2017). In our model, the area to be covered is partitioned and each agent operates in a periodic curve on its assigned subarea.

1.4 Outline

In Sect. 2, we formally state the optimization resilience problems studied here and, in Sect. 3, the necessary technical tools are presented. The results on the coverage resilience are shown in Sect. 4. These include a linear time algorithm for computing the coverage resilience for general communication graphs, as well as analytic expressions, as functions of the input size, for particular configurations including trees and grids. In Sect. 5, we show that the k-isolation resilience introduced in Bereg et al. (2018) can be computed efficiently for cycles and grids, in spite of being NP-hard in general. In Sect. 6, we establish the relationship between the broadcasting resilience and circulant

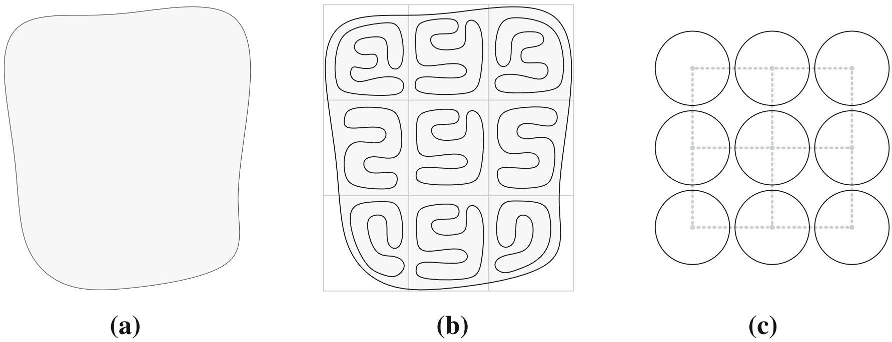

Fig. 4 **a** Region to patrol. **b** Area decomposition of the region to patrol using a grid pattern and the paths to cover every cell of the region. **c** Simplification of the practical model using an abstraction with circular trajectories. The underlying communication graph is represented using gray dotted strokes

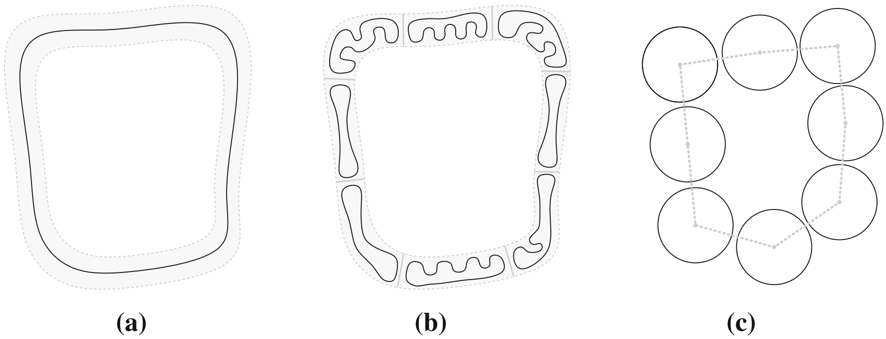

Fig. 5 **a** Solid black stroke represent a border to monitor. The gray strip around the border is the region to patrol. **b** Decomposition of the region into cells, one cell per robot and for every cell a closed trajectory to cover it is computed. **c** Simplification of the practical model using an abstraction with circular trajectories. The underlying communication graph is represented using gray dotted strokes

graphs, and solve the problem for cycles and grids. Finally, in Sect. 7 we conclude with a summary of our results and a description of various open problems.

2 Problems definition

In the rest of this paper we consider the simple circular model introduced by the authors of Díaz-Báñez et al. (2015). However, all the results can be extended to general closed trajectories using the same arguments of Díaz-Báñez et al. (2015) and Díaz-Báñez et al. (2017) under the same set of assumptions. The first two definitions are borrowed from Bereg et al. (2018).

Definition 1 (*Synchronized communication system* (*SCS*)) Let $T = \{C_1, \ldots, C_n\}$ be a set of unit circles (trajectories) which are pairwise disjoint. Let $\epsilon < 0.5$ be the communication range of the robots. Let G be a bipartite connected graph whose vertices are the centers of the circles in T and whose edges are given by a subset of the pairs (not necessarily all) of circle centers at distance $2+\epsilon$ or less. A *synchronized communication system* (SCS) with *communication graph* G consists of a team of n robots, one per trajectory, such that every pair of neighboring trajectories in G have opposite movement directions and every pair of robots in neighboring trajectories are synchronized. An m-partial SCS, $0 < m \leq n$, is a synchronized communication system in which $n - m$ robots have left the team and the m remaining robots apply the shifting strategy. If $n = m$, the SCS is said to be *saturated*.

Note that an SCS is a type of partial SCS where no robots have left. Thus, any claims about partial SCSs holds for saturated SCSs as well.

We assume that the shifting operation is instantaneous; this ensures that, if the fallen robot was synchronized with other neighbor, the active robot will arrive on time at the corresponding link positions after the shifting operation has been performed (in real scenarios the robot accelerates during shifting in order to reach the next link position at the expected time, then it can maintain the planned constant speed).

In an m-partial SCS, a robot *starves* or it is in *starvation* if every time that it arrives at a link position the corresponding neighbor is not there causing a shifting to the neighboring trajectory. The *starvation number* of an SCS is the maximum possible number of starving robots in a partial SCS.

Definition 2 (*k-isolation resilience of a SCS*) The *k-isolation resilience* of a SCS ($k \geq 1$) is the minimum number r_i, such that, there exist r_i robots whose removal causes the starvation of at least k surviving robots. If it is not possible to obtain k starving robots then the k-isolation resilience is set to infinity.

Remark 1 As noted in Corollary 16 and Lemma 17 of Bereg et al. (2018) if the starvation number of an SCS is s then for all $k > s$ the k-isolation resilience of the system is infinity and the s-isolation-resilience is $n - s$.

We say that an m-partial SCS has *loss of connectivity* if there exists a pair of surviving robots that cannot exchange messages (possibly through a sequence of message relays between neighbors) or if there is only one surviving robot in the system.

Definition 3 (*Broadcasting resilience*) The *broadcasting resilience* of a SCS is the minimum number r_b, such that, there exist a set of r_b robots whose removal causes a loss of connectivity in the system.

Definition 4 (*Coverage resilience*) The *coverage resilience* of a SCS is the minimum number r_c, such that, there exist r_c robots whose removal results in at least one non-covered trajectory segment.

A generalization of the concept above is the T-coverage resilience.

Definition 5 (*T-coverage resilience*) Given $T > 0$, the T-coverage resilience is the minimum number r_{c_T}, such that, there exist r_{c_T} robots whose removal causes the idle-time of some trajectory point to be at least T units of time.

Remark 2 Notice that in a saturated SCS the idle-time of any trajectory point is 1. So, the T-coverage resilience is zero for all $T \leq 1$.

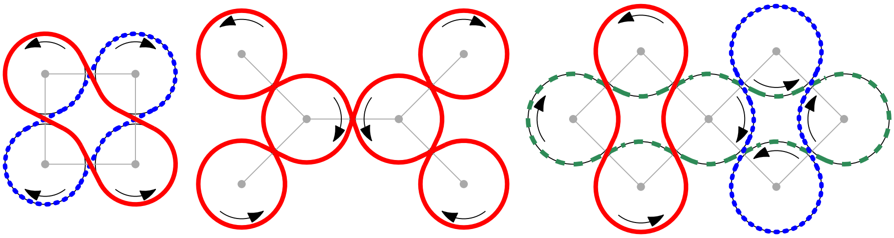

Fig. 6 SCS trajectories are partitioned into rings. **a** two rings; **b** one ring; **c** three rings

In general, higher resilience values correspond to increased fault tolerance. We focus on the following problems.

Problem 1 Given a SCS, determine its coverage resilience.

Problem 2 Given a SCS whose communication graph is a cycle or a grid, determine its k-isolation resilience for a given natural number k.

Problem 3 Given a SCS, compute its broadcasting resilience.

3 Technical tools

In this section we present the two technical tools that we will use to solve the stated problems. The first subsection deals with the notion of *rings*, a concept first introduced in Bereg et al. (2018). The second subsection presents two new concepts, *tokens* and *token graph*.

3.1 Rings

This subsection presents (with no proof) some results and concepts related to *rings* that were introduced in Bereg et al. (2018). We enunciate them here in order to make this paper self-contained.

Definition 6 (*Ring*) A *ring* in an SCS with communication graph G, is the locus of points visited by a robot following the assigned movement direction in each trajectory and always shifting to the neighboring trajectory (in G) at the corresponding link positions.

Figure 6 shows various SCSs with different numbers of rings, each ring shown in a different color and stroke type.

Remark 3 Each point in a trajectory belongs to a single ring, so, the rings in an SCS are pairwise disjoint and partition the trajectory points into equivalence classes. Each ring is a closed path composed of segments of trajectories and has a direction of travel determined by the movement in the participating trajectories.

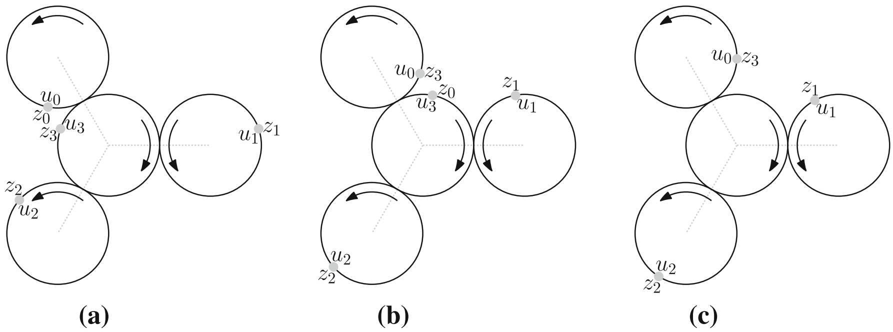

Fig. 7 Tokens and robots movement. **a** Initial state of the SCS. At time 0, robot u_i holds token z_i. **b** State of the system at time $t_1 > 0$. Robots u_0 and u_3 met and exchanged tokens. **c** State of the system at time $t_2 > t_1 > 0$. Robot u_3 and its token z_0 have been removed

Definition 7 (*Path in a ring*) A *path* in a ring r from a point $p \in r$ to a point $q \in r$ is the ordered set of visited points from p to q following the travel direction of r (it may contain tours on r).

As suggested by the examples in Fig. 6, for any given system, different rings may have different. In discussing the length of a ring, it is convenient to ignore the effect on distance arising from shifting between neighboring trajectories and we assume neighboring circular trajectories are tangent to each other.

Definition 8 (*Length of a ring*) The *length of a ring* is defined as the sum of the lengths of trajectory arcs forming the ring. The length of every ring in an SCS is in $2\pi\mathbb{N}$ (Corollary 13 of Bereg et al. (2018)).

Remark 4 We can extend the concept of *length* to a path in a ring as the sum of the lengths of trajectory arcs forming the path. The length of a path between two robots in the same ring is also in $2\pi\mathbb{N}$ (Lemma 12 of Bereg et al. (2018)).

Lemma 1 *In an m-partial SCS the number of robots in a given ring remains invariant; if the length of the ring is $2\ell\pi$ then it has at most ℓ robots. Furthermore, in a saturated SCS, a ring of length $2\ell\pi$ has exactly ℓ robots, each at distance 2π from the next.*

3.2 Tokens and the token graph

In this section we introduce the notion of a *token* as an abstract entity used to describe the behavior of a partial SCS. The key idea is to focus on the tokens instead of the robots. This will be crucial to compute the broadcasting resilience. Let $u_0, \ldots, u_{n-1}$ be the set of robots at the beginning of time (when the SCS is deployed) each one carrying a token. We assume the following protocol: at all times, each active robot holds a token. When two robots meet, they exchange their tokens. When a robot is removed, the token it carries is removed as well, see Fig. 7.

Remark 5 A starving robot never exchanges its token.

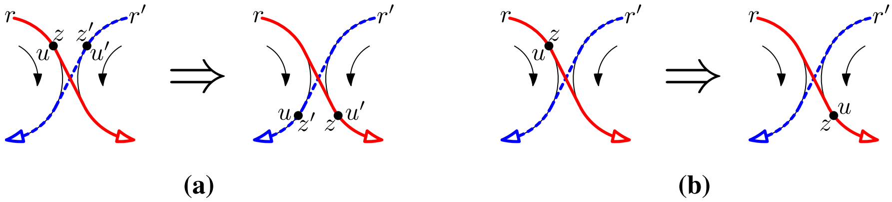

Fig. 8 **a** Robots u and u', bearing tokens z and z' respectively, arrive at the same time at a link position, so, they meet each other. Analogously, we can say that the tokens z and z' meet each other as well. **b** Robot u, bearing token z, performs a shifting operation because it detects no robot at the link position

Lemma 2 *Consider a partial SCS. Independently of the number of removed robots and when they were removed, a live token always remains in its initial ring and moves with constant speed (2π per unit time) along its ring.*

Proof Let z be a token in a partial SCS. Let u be the robot bearing z at an arbitrary time t. Let r be u's ring at time t. Let $\Delta t > 0$ be a real number. Suppose first that in the time interval from t to $t + \Delta t$, robot u remains active and does not reach any link position. Then, in this time interval, u moves along r with constant speed of 2π per unit time (z has the same behavior). Suppose now that there exists t', with $t \leq t' \leq t + \Delta t$, such that u remains active from t to t' and, u arrives at a link position ℓ at t' for the first time since t. Thus, in the time interval from t to t' the token z moves with u along r with constant speed 2π per unit time. If, at time t', u meets another robot u' at ℓ, then both robots remain in their trajectories and exchange their tokens, see Fig. 8a. Otherwise, if at time t', u detects no neighbor at ℓ, then it performs a shifting operation and passes to the neighboring trajectory in the same ring, see Fig. 8b. Either way, token z keeps moving in ring r immediately after t' with constant speed 2π per time unit. From successive applications of the previous arguments the result follows. □

Definition 9 Two tokens *meet* each other if they arrive at the same link position at the same time (see Fig. 8a).

The proof of the following result follows easily from Lemma 2.

Lemma 3 *Let z and z' be two tokens of a partial SCS in rings r and r' (possibly the same ring), respectively. The tokens z and z' will meet each other if and only if the following three conditions are fulfilled:*

1. *there is a link position ℓ where the rings r and r' cross each other,*
2. *there are two paths of the same length L: one from z to ℓ in r (possibly longer than r) and other from z' to ℓ in r' (possibly longer than r'), and,*
3. *z and z' remains in the system after $\frac{L}{2\pi}$ time units.*

Definition 10 (*Correspondence between tokens*) Let z and z' be two tokens in a partial SCS. We say that there is a *correspondence* between z and z' at time t if there exists a $\Delta t > 0$ such that z and z' meet each other at time $t + \Delta t$.

Informally, we can say that there is a correspondence between two tokens if, provided they survive long enough, they meet each other.

Lemma 4 *Let r and r' be two rings of a partial SCS of lengths $2\pi\mu$ and $2\pi\mu'$, respectively. Let z and z' be two tokens in r and r', respectively. If z and z' meet each other at ℓ then they will meet each other every* $\mathrm{mcm}(\mu, \mu')$ *(minimum common multiple of μ and μ') time units while they remain in the system.*

Proof Suppose that z and z' meet each other at ℓ at time t. The location reached after traveling $\mathrm{mcm}(\mu, \mu')$ time units from ℓ in r is the point $\ell + 2\pi \cdot \mathrm{mcm}(\mu, \mu') = \ell$. The same occurs by traveling on r'. If at time $t + \mathrm{mcm}(\mu, \mu')$ the tokens z and z' remain in the system, then they meet each other at ℓ (Lemma 3). The result follows from successive applications of this argument. □

Let x be a point in a ring r and let d be a non-negative real number. The point $x + d$ is the point reached by traveling distance d from x following the travel direction in r. Analogously, the point $x - d$ is the point y such that $y + d$ is x. Notice that d could be greater than the length of r. Also notice that $x - d + d = x + d - d = x$.

Lemma 5 *Let z and z' be two tokens in a partial SCS at time t_1. Suppose that at time $t_2 > t_1$ a set (possibly empty) of robots (with their respective tokens) have been removed from the system but z and z' remain. Then, there is a correspondence between z and z' at time t_1 if and only if there is a correspondence between z and z' at time t_2.*

Proof Let r and r' be the rings of z and z', respectively (r and r' could be the same ring). Let $2\pi\mu$ and $2\pi\mu'$ be the lengths of r and r', respectively.

Suppose there is a correspondence between z and z' at time t_1. Let x_{t_1} and x'_{t_1} be the positions of z and z' in r and r', respectively, at time t_1. Then, by the definition of correspondence and Lemma 3, there are paths p and p' (in r and r' respectively) of length $2\pi\Delta t$ starting at x_{t_1} and x'_{t_1}, respectively, and ending at a link position ℓ. That is:

$$x_{t_1} + 2\pi\Delta t = \ell$$
$$x'_{t_1} + 2\pi\Delta t = \ell.$$

Let k be the smallest non-negative integer such that $k \cdot \mathrm{mcm}(\mu, \mu') + \Delta t \geq t_2 - t_1$. From Lemma 4, we have that:

$$x_{t_1} + 2\pi(k \cdot \mathrm{mcm}(\mu, \mu') + \Delta t) = \ell$$
$$x'_{t_1} + 2\pi(k \cdot \mathrm{mcm}(\mu, \mu') + \Delta t) = \ell.$$

Let $x_{t_1} + 2\pi(t_2 - t_1)$ and $x'_{t_1} + 2\pi(t_2 - t_1)$ be the positions occupied by z and z' after traveling for $t_2 - t_1$ time units, respectively. Let q and q' be the paths obtained by traveling from these positions in r and r', respectively, during $k \cdot \mathrm{mcm}(\mu, \mu') + \Delta t - (t_2 - t_1)$ time units. Notice that q and q' have the same length $2\pi(k \cdot \mathrm{mcm}(\mu, \mu') + \Delta t - (t_2 - t_1))$. Moreover:

$$x_{t_1} + 2\pi(t_2 - t_1) + 2\pi(k \cdot \mathrm{mcm}(\mu, \mu') + \Delta t - (t_2 - t_1)) = \ell$$
$$x'_{t_1} + 2\pi(t_2 - t_1) + 2\pi(k \cdot \mathrm{mcm}(\mu, \mu') + \Delta t - (t_2 - t_1)) = \ell.$$

Therefore, there will be a correspondence between z and z' at time t_2.

Now, suppose there is a correspondence between z and z' at time t_2. Let x_{t_2} and x'_{t_2} be the positions at time t_2 of z and z' in r and r', respectively. Then, by definition of correspondence and Lemma 3, there are two paths p and p' (in r and r' respectively) of length $2\pi\Delta t$ starting at x_{t_2} and x'_{t_2}, respectively, and ending at a link position ℓ. That is:

$$x_{t_2} + 2\pi\Delta t = \ell$$
$$x'_{t_2} + 2\pi\Delta t = \ell.$$

Recall that if x is a position in a ring and d is non-negative real number then the position $x - d$ is the point y such that $y + d$ is x. Then, from Lemma 2 we have that $x_{t_2} - 2\pi(t_2 - t_1)$ and $x'_{t_2} - 2\pi(t_2 - t_1)$ were the positions at time t_1 of z and z', respectively. Let q and q' be the paths obtained by traveling from these positions in r and r', respectively, during $t_2 - t_1 + \Delta t$ time units. Notice that q and q' have the same length $2\pi(t_2 - t_1 + \Delta t)$. Moreover:

$$x_{t_2} - 2\pi(t_2 - t_1) + 2\pi(t_2 - t_1 + \Delta t) = x_{t_2} + 2\pi\Delta t = \ell$$
$$x'_{t_2} - 2\pi(t_2 - t_1) + 2\pi(t_2 - t_1 + \Delta t) = x'_{t_2} + 2\pi\Delta t = \ell.$$

Therefore, there was a correspondence between z and z' at time t_1 and the result follows. □

Remark 6 The previous lemma states that the *relation* of correspondence between two tokens remains invariant unless one of them is removed from the system.

Definition 11 (*Token graph*) Let $\mathcal{F}$ be an m-partial SCS. Let Z be the set of surviving tokens. The *token graph* of $\mathcal{F}$ is the graph $\mathcal{T}_{\mathcal{F}}$ whose vertices are the tokens in Z [that is, $V(\mathcal{T}_{\mathcal{F}}) = Z$] and whose set of edges is:

$$E(\mathcal{T}_{\mathcal{F}}) = \{\{z, z'\} | z \in Z, z' \in Z, \text{there is a correspondence between } z \text{ and } z'\}.$$

From Remark 6, the token graph of a partial SCS remains invariant while no additional robots removed. Figure 9a shows the token graph of the SCS of Fig. 7 before the robot u_3 is removed.

From the definition of token graph and Lemma 5, the next result (illustrated in Fig. 9b) follows.

Lemma 6 *Let $\mathcal{F}$ be an m-partial SCS with set of tokens Z. Let $\mathcal{F}'$ be the m'-partial SCS resulting from the removal of some robots from $\mathcal{F}$, with $Z' \subset Z$, the set of tokens of $\mathcal{F}'$. Let $\mathcal{T}_{\mathcal{F}}$ and $\mathcal{T}_{\mathcal{F}'}$ be the token graphs of $\mathcal{F}$ and $\mathcal{F}'$ respectively. Then, $\mathcal{T}_{\mathcal{F}'}$ is the subgraph induced by Z' in $\mathcal{T}_{\mathcal{F}}$.*

The following lemma relates starvation with token and will be useful for computing the k-isolation resilience. The proof is straightforward by using Remark 5 and Lemma 5.

Lemma 7 *Let z be a token in a partial SCS $\mathcal{F}$. Let u be the robot bearing z. The robot u is starving in $\mathcal{F}$ if and only if no matters how long the system is running with no more removal of robots, z does not meet any other token or, equivalently, z has degree 0 in the token graph of $\mathcal{F}$.*

4 Computing the coverage resilience

In this section we show how to compute the coverage resilience for arbitrary graphs as well as for trees and grids.

Notice that if a token z is at a point x of a ring r then x is being covered by the robot bearing z. This simple observation allows us to study the coverage resilience using tokens.

Upon deployment of a system, a ring of length $2\pi l$ contains l tokens at distance 2π from one to the next (Lemma 1). From Lemma 2, the distribution of the tokens in a ring when some robots have been removed from the system looks like Fig. 10. The following theorems are deduced.

Theorem 1 *Let r be a ring of length $2l\pi$ in a partial SCS. If $a < l$ is the maximum number of absent consecutive tokens in r then the idle-time of any point in r is $a + 1$ (see Fig. 10).*

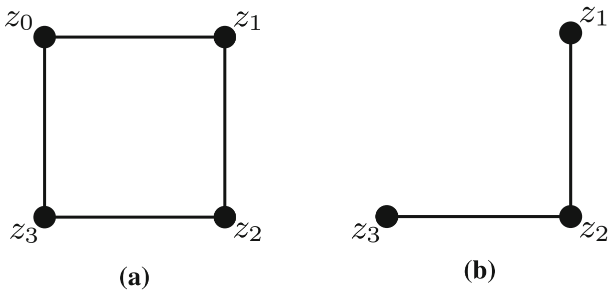

Fig. 9 **a** Token graph of the SCS of Fig. 7 at time 0 (Fig. 7a). Notice that the token graph at time t_1 (Fig. 7b) does not change. **b** The token graph of the resultant partial SCS after removing robot u_3 (Fig. 7c)

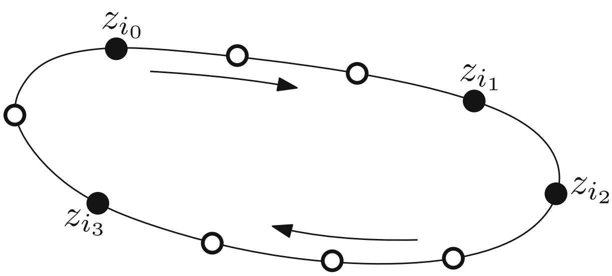

Fig. 10 Tokens in a ring of a partial SCS. The hollow points represent the removed tokens

Theorem 2 *Let r be a ring in a partial SCS. A trajectory segment of r is non-covered if and only if r does not contain surviving robots.*

From Remark 3 and Theorem 2 the following result is deduced.

Corollary 3 *Let c be the number of rings in an SCS. Let* $2l_1\pi, 2l_2\pi, \ldots, 2l_c\pi$ *be the lengths of the c rings. The coverage resilience of the system is the minimum of* $\{l_1, l_2, \ldots, l_c\}$.

Theorem 4 *The coverage resilience of a given SCS can be computed in linear time on the number of trajectories in the system.*

Proof Corollary 3 suggests a simple algorithm to compute the coverage resilience of a SCS by determining the rings in the system and their lengths. To do that, it is sufficient to follow the movement of a token until reaching the starting point. In this way, the length of a detected ring is the sum of the lengths of the traversed arcs. The complexity of detecting a ring in this way is $O(m)$ where m is the number of link positions traversed by the ring. When a ring is detected (by returning to the starting point), then we choose a non visited trajectory arc in order to detect another ring and so on. If there are no more non-visited trajectory arcs then we are done. Let G be the communication graph of the SCS. Recall that every link position of the system is an edge of the communication graph G. Clearly, the complexity of this algorithm is $O(|E|)$ where E is the set of edges in G. Taking into account that the communication graph is planar then this algorithm has running time $O(n)$ where n is the number of trajectories. □

Given a value $T > 0$, the T-coverage resilience can also be computed using the rings of the SCS. Let c be the number of rings in the system and $2l_1\pi, 2l_2\pi, \ldots, 2l_c\pi$ their lengths. Let l^* be the minimum of $\{l_1, l_2, \ldots, l_c\}$. Using Lemma 1 we deduce that if $l^* \geq \lceil T \rceil$, then the T-coverage resilience is $\lceil T \rceil - 1$; otherwise, the T-coverage resilience is l^*. As a consequence, we can state the following.

Theorem 5 *The T-coverage resilience of an SCS is* $\min\{l^*, \lceil T \rceil - 1\}$ *where* $2\pi l^*$ *is the length of the shortest ring in the system.*

4.1 Coverage resilience for trees and grids

If the communication graph is a tree, it contains a single ring (Lemma 7 of Bereg et al. (2018)) and thus its coverage resilience is n, where n is the number of trajectories. Furthermore, the T-coverage resilience is $\min\{n, \lceil T \rceil - 1\}$. Therefore the system is very stable (with respect to covering) because no matter the number of robots is removed, the remaining robots will cover all the trajectories.

In the rest of this section, we study the coverage resilience of a SCS where the communication graph is a grid. Consider a set of $M \cdot N$ trajectories distributed in M rows and N columns. Each trajectory is identified by a pair (i, j) where $1 \leq i \leq M$ and $1 \leq j \leq N$ indicate the row and the column, respectively, where the trajectory is located. In the communication graph, trajectory (i, j) is linked to trajectories $(i-1, j)$

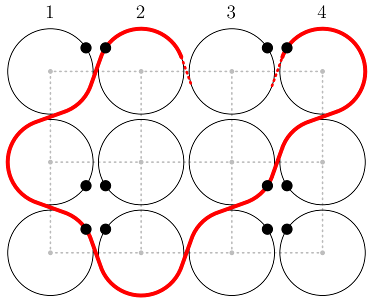

Fig. 11 A 3×4 grid SCS. The drawn portion of a ring hits the top boundary at trajectories $(1, 2)$ and $(1, 4)$, it hits the bottom boundary at $(3, 2)$, the left boundary at $(2, 1)$ and the right one at $(1, 4)$.

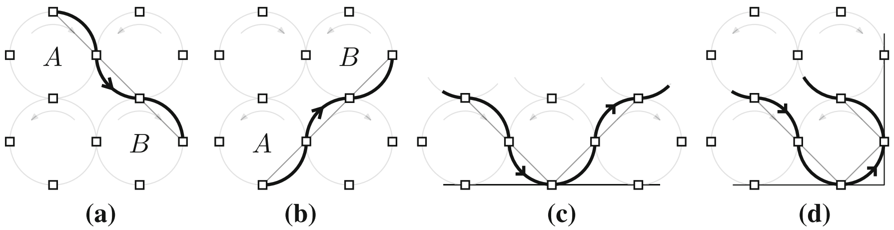

Fig. 12 Local behavior of a ring in a grid SCS

if $i > 1$, $(i, j-1)$ if $j > 1$, $(i+1, j)$ if $i < M$ and $(i, j+1)$ if $j < N$. We refer to this type of SCS as a *grid SCS*. An *m-partial grid SCS* is analogously defined.

In a grid SCS we say that a ring *hits* the top boundary of the grid if the ring passes through the top section of a circle in the first row. Analogously we can define when a ring hits the left, bottom or right boundary of the ring. See Fig. 11.

Figure 12 shows the local behavior of a ring as it visits the trajectories of a grid SCS. In this figure, small white squares denote two kinds of points: the points where the ring hits the boundaries and the contact points between two neighboring circles (for simplicity, tangent trajectories are considered). Note that these points are traversed diagonally (with slopes 1 and -1) by a ring. We consider the *movement lattice* formed by these points (see Fig. 13). If the grid communication graph has M rows and N columns then the movement lattice has $2M+1$ rows indexed from 0 (topmost) to $2M$ (bottommost) and $2N+1$ columns indexed from 0 (leftmost) to $2N$ (rightmost). In this way, every vertex of the lattice can be referenced by its row and column (see Fig. 13). Notice that a starving robot moves diagonally on this lattice (following the ring that houses it) and only changes its direction when reaching a vertex with out-degree equal to one (i.e. when it hits a grid boundary, it *bounces*).

Lemma 8 *An $M \times M$ grid SCS has M rings of length $2M\pi$. Each ring hits each of the four boundaries exactly once.*

Proof Let H be the movement lattice of the system. Let r be the ring that hits the top boundary at the vertex $(0, i)$ of H. It is easy to see that r hits the right boundary at vertex $(2M-i, 2M)$, the bottom boundary at the vertex $(2M, 2M-i)$ and the left boundary at $(i, 0)$. Also, r does not hit the boundaries in any other vertex. Note that a step from one vertex to another in the movement lattice corresponds to a section of a ring of length $\pi/2$. The ring r visits $4M$ vertices on H, thus the length of r is

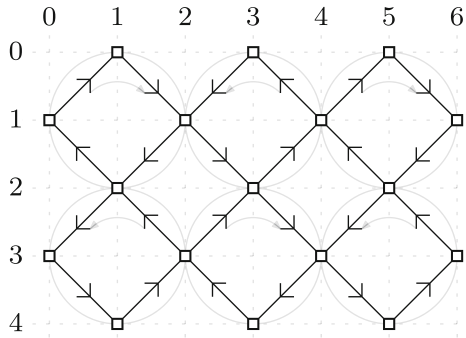

Fig. 13 Movement lattice in a grid

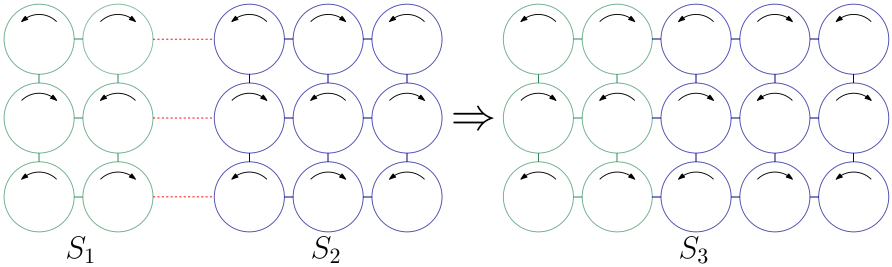

Fig. 14 S_3 is the resultant SCS of the concatenation of the SCSs S_1 and S_2

$4M \cdot \pi/2$, i.e., $2M\pi$. Repeating this argument for each hitting point in the top boundary we obtain M rings of length $2M\pi$. The sum of the lengths of all circles in the system is $2M^2\pi$ and the sum of the lengths of the M rings in the system is $2M^2\pi$ too, so there is no other ring in the system. □

Let S_1 and S_2 be two grid SCSs. We say that S_1 and S_2 are *concatenable* if S_1 and S_2 have the same number of rows M and for all $1 \leq i \leq M$ the movement direction assigned to the last trajectory of the ith row of S_1 is opposite to the movement direction assigned to the first trajectory of the ith row of S_2. The *concatenation* of S_1 and S_2, such that the last trajectory in the ith row of S_1 is linked with the first trajectory in the ith row of S_2 (for all $1 \leq i \leq M$), produces a new $M \times (N + N')$ grid SCS, see Fig. 14.

The following result is a technical lemma that we need in order to complete the proof of Theorem 6.

Lemma 9 *Let S and U be two concatenable grid SCSs of size $M \times N$ and $M \times M$, respectively. Suppose that S has k rings of the same length l, and every ring in S hits each of the left and right boundaries exactly c times, and hits each of the top and bottom boundaries exactly c' times. Let R be the $M \times (N + M)$ grid SCS resulting from the concatenation of S and U. Then, R has exactly k rings of the same length $l + 2cM\pi$ and every ring in R hits each of the left and right boundaries exactly c times, and hits each of the top and bottom boundaries exactly $c' + c$ times.*

Proof By Lemma 8, U has M rings of length $2M\pi$, and all of them hit each boundary once. Thus, every ring in U extends the length of a ring in S by $2M\pi$, see Fig. 15.

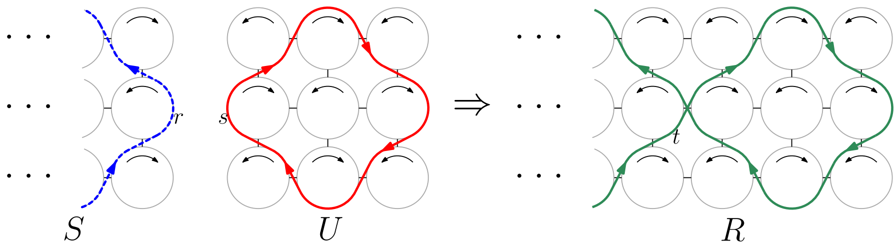

Fig. 15 r is the ring in S that hits the right boundary in the second row. s is the ring in U that hits the left boundary in the second row. t is the ring in R obtained from r and s in the concatenation of S and U

Since every ring in S hits the right boundary c times then, after concatenation, each ring in S is fused with c rings in U. Note that two rings in S can not be fused together in the concatenation. Then, every ring in R is formed by the fusion of a ring in S and c rings in U. Therefore, R has k rings of the same length $l + 2cM\pi$. Every ring so obtained hits the left boundary c times at the c hitting points of the respective ring in S, and hits the right boundary c times at the hitting points of the c respective rings in U. Every ring in R hits each of the top and bottom boundaries $c' + c$ times, at the c' hitting points of the respective ring in S and at the hitting points of the c respective rings in U.

We now proceed to establish a key result of this subsection.

Theorem 6 *An $M \times N$ grid SCS has* $\gcd(M, N)$ (*greatest common divisor of M and N*) *rings of the same length* $\frac{2\pi MN}{\gcd(M,N)}$. *Moreover, every ring r hits each of the left and right boundaries* $\frac{M}{\gcd(M,N)}$ *times and hits each of the top and bottom boundaries* $\frac{N}{\gcd(M,N)}$ *times.*

Proof We prove the result by induction in the number of rows. For $M = 1$, we have that every $1 \times N$ grid SCS has a single ring, $\gcd(1, N) = 1$, one hitting point in left and right boundaries, and N hitting points in the top and bottom boundaries. Thus, the theorem holds in this case. Assume as inductive hypothesis that: for a fixed value P, the theorem holds for every $M \times N$ grid SCS with $M \le P$.

We need to prove the theorem for a $(P + 1) \times N$ grid SCS. If $N \le P$ then, using the fact that a $(P + 1) \times N$ grid SCS is equivalent to a $N \times (P + 1)$ grid SCS, the theorem holds by the inductive hypothesis. If $N = P + 1$ then the theorem holds by Lemma 8. In order to prove the theorem for a $(P + 1) \times N$ grid SCS with $N > P + 1$ we use induction in the number of columns. Assume as second inductive hypothesis that: for a fixed value $Q \ge P + 1$, the theorem holds for every $(P + 1) \times N$ grid SCS with $N \le Q$.

Let S be a $(P+1)\times(Q+1)$ grid SCS. We have that $Q+1 > P+1$, then removing the last $P + 1$ columns of S we obtain a $(P + 1) \times (Q - P)$ grid SCS denoted by S'. The theorem holds for S' by the second inductive hypothesis. The $P + 1$ removed columns conform a $(P + 1) \times (P + 1)$ grid SCS which is concatenable with S'. So, concatenating S' with the $P + 1$ removed columns we obtain S. Then, by using Lemma 9 and properties of the greatest common divisor, the result follows. ◻

By using Theorem 6, Corollary 3 and Theorem 5 we arrive to the main result of this subsection.

Theorem 7 *The coverage resilience of an $M \times N$ grid SCS is:*

$$\frac{M \cdot N}{\gcd(M, N)}.$$

And, the value of the T-coverage resilience is:

$$\min\left\{\frac{M \cdot N}{\gcd(M, N)}, \lceil T \rceil - 1\right\}.$$

5 Computing the *k*-isolation resilience for cycles and grids

In Bereg et al. (2018), the authors show that the problem of computing the k-isolation resilience is NP-hard, even when the communication graph is a tree. In this section we show how to compute efficiently this measure when the communication graph is a cycle or a grid.

5.1 Cycles

Lemma 10 *Let G be the communication graph of a m-partial SCS. If G is a cycle, then the system has exactly two rings, one with CW direction and the other with CCW direction. Furthermore, every edge of G corresponds to a crossing of the two rings.*

Proof We proceed by induction on the number $2M$ of trajectories (nodes) in G (recall that G is bipartite, so every cycle has even length). If $M = 2$, we have 4 trajectories, and the claim holds, as shown in Fig. 6a. Assume as inductive hypothesis that, for a fixed value M, the claim holds for every cycle graph with $2M$ trajectories. Now, let us consider a cycle graph G with $2(M + 1)$ trajectories (Fig. 16a). If we remove two consecutive trajectories (C and D) and "glue" the ends in the broken section we obtain a cycle graph G' with $2M$ trajectories (Fig. 16b). By the inductive hypothesis, the claim holds for G'. Figure 16b shows the clockwise ring using solid stroke in red, and the other one using dashed stroke in blue. The sections of rings in the removed two circles are also shown in Fig. 16b. After that, inserting again the two removed trajectories into the original position we obtain a cycle graph with $2(M+1)$ trajectories, reestablishing the claim, see Fig. 16c. Note that, as depicted in Fig. 16c, when we reinsert the removed trajectories, we match terminals (solid-circle to solid-circle, solid-square to solid-square, etc.). ◻

Lemma 11 *In an m-partial SCS, whose communication graph is a cycle with rings r and r', a robot in r is starving if and only if r' is empty of robots (tokens).*

Proof Suppose that there are tokens z and z' in r and r', respectively. From Lemmas 2 and 10 we deduce that if z and z' remain in the system long enough they will meet

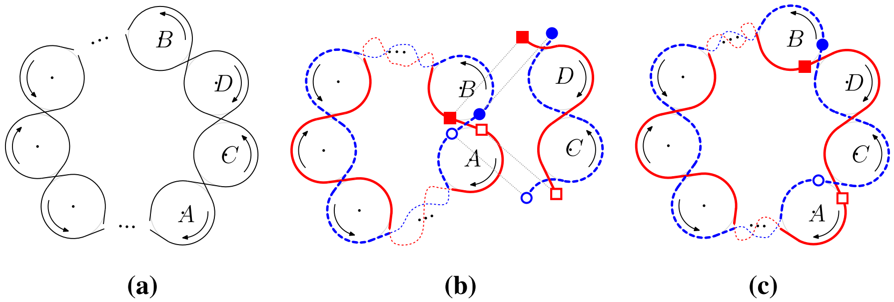

Fig. 16 Induction proof of Lemma 10

each other, so, there is a correspondence between them. Also, note that there is no correspondence between tokens of the same ring. Then, from Lemma 7 it follows that a robot in r bearing a token z is starving if and only if the ring r' is empty of tokens. □

Using Lemmas 11 and 1 we conclude.

Theorem 8 *Consider a system whose communication graph is a cycle and let r and r' be the two rings with lengths $2\pi l$ and $2\pi l'$, respectively. The starvation number of the system is* $\max\{l, l'\}$ *and the k-isolation resilience is* $\min\{l, l'\}$ *if $k \leq$* $\max\{l, l'\}$ *and infinity, otherwise.*

Remark 7 If the communication graph is a cycle with starvation number ζ, then for all $k \leq \zeta$, the value of the k-isolation resilience matches the value of the coverage resilience of the system (Theorem 8 and Corollary 3).

To close this subsection, notice that to compute the k-isolation resilience of an SCS whose communication graph is a cycle we can use the same linear time algorithm described in Theorem 4.

5.2 Grids

In this subsection we revisit the grid SCS introduced in Sect. 4.1. We show a set of results that allow us to calculate the k-isolation resilience with a closed formula. All the results presented in previous sections or in Bereg et al. (2018) are directly based on the properties of rings (their lengths, number, or topology). However, the following results, surprisingly, do not use these tools.

Consider again the movement lattice introduced in Sect. 4.1. Analyzing the movement of a token in the movement lattice of a grid SCS, we arrive at the following remark.

Remark 8 In the lattice of a grid SCS, every token moves with constant speed according to a *mathematical billiard* pattern of motion Tabachnikov (2005). Moreover, the *X-motion* (resp. *Y-motion*) of a token, which is the projection of the token motion onto the X-axis (resp. Y-axis), is a periodic movement of a ball on a line segment that bounces off the ends, see Fig. 17. Notice also that the speed in the X-axis (resp. Y-axis), hence on the lattice as well, of all the tokens is the same.

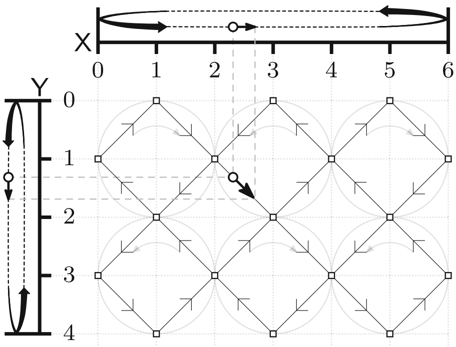

Fig. 17 The small hollow disk represents a token moving in the movement lattice. The horizontal and vertical projections of the token motion are also shown

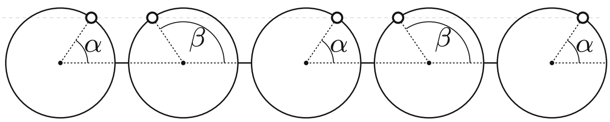

Fig. 18 The white points represent the positions of robots in trajectories of a same row in a grid SCS, $\beta = \pi - \alpha$

Lemma 12 *In a partial grid SCS, if two robots occupy trajectories in the same row (resp. column) at some instant of time, then their respective tokens have always been and will always be in the same row (resp. column) of the movement lattice.*

Proof Let u and v be robots in the same row, say row i, at some time t (the case of robots in the same column is treated similarly). Suppose that robot u is at position α. Since the system is synchronized, the position of robot v is at α or $\pi - \alpha$, see Fig. 18. However, independent of whether the position of v is α or $\pi - \alpha$, they both have the same direction along the Y axis (both going down or both going up) due to synchronization. Then, due to the periodic constant motion of the tokens in Y axis (Remark 8), the tokens of u and v are in the same row at any time (before t and after t). The lemma follows. □

Theorem 9 *A robot in a partial grid SCS is starving if and only if there is no other robot in the same row or column at any time.*

Proof ($\Rightarrow$) Let u be a starving robot. For the sake of contradiction, suppose that there is another robot v in the same row at some time t (the case when they are in the same column is analyzed similarly). Let z and z' be the tokens of u and v at time t, respectively. Since the X-motion of each of these tokens is periodic and has the same constant speed, it is easy to see that after some time, say Δt units of time, z and z' will meet each other in the X-axis projection. Thus, the tokens z and z' have the same x and y coordinates in the movement lattice at time $t + \Delta t$. Since z and z' at time $t + \Delta t$ belong to two different robots (one robot is u and the other robot is not necessarily v), these robots are at the link position between two neighboring trajectories as shown in Fig. 1a. Therefore, robot u is not starving, a contradiction.

($\Leftarrow$) It suffices to show that, if a robot u is not starving, then there is another robot in the same row or column at some time. Robot u will be communicating with another robot, say v, at some time t. There are two possibilities for the link position in the grid. The trajectories of u and v at time t are either in the same row or in the same column. Either way, the theorem follows. □

Finally, by induction on k and the pigeonhole principle, we conclude.

Theorem 10 *The starvation number in an $M \times N$ grid SCS is $\min(M, N)$ and its k-isolation resilience is $k(M + N - 2) - k(k - 1)$ if $k \leq \min(M, N)$ and infinity, otherwise.*

6 The broadcasting resilience

The robots of a synchronized communication system (a saturated SCS) conform a *connected network* because every pair of robots can send/receive messages between each other, possibly by multiple relays between neighboring robots.

We can model the message transmission by means of the following protocol: the messages are carried by the tokens instead of the robots. If a token is carrying a message and meets another token who does not know it, then, after the meeting, both of them know the message. Therefore, when a robot u sends a message at time t, we assume that u associates the message with its token at time t. At time $t' > t$, we say that a robot u' knows the message if and only if it is bearing a token that knows the message.

Lemma 13 *Let $\mathcal{F}$ be an m-partial SCS and assume that no more robots leave the system. If at some time t robot u of $\mathcal{F}$ is bearing a token z then u will bear z periodically. More formally, if u in $\mathcal{F}$ is bearing token z at time t, then there exist a value $\sigma \in \mathbb{N}$ ($\sigma > 0$) such that u is bearing z again at time at time $t + \sigma$.*

Proof Let $U = \{u_{i_1}, \ldots, u_{i_m}\}$ be the set of live robots in $\mathcal{F}$ and let Z be the set of tokens in $\mathcal{F}$. Fix a time t_0. For every trajectory C_j $(1 \leq j \leq n)$ in the system, let p_j be the position of C_j where there is or there should be (due to the synchronization of the system) a robot at time t_0, see Fig. 19. Let $P = \{p_1, \ldots, p_n\}$. Notice that if a robot u is at a point $p \in P$ then after one time unit u' is at point p' that is in P as well. For every $l \in \mathbb{N}$ and $1 \leq j \leq m$, let $(\rho_j^{(l)}, \tau_j^{(l)})$ be an ordered pair where $\rho_j^{(l)}$ and $\tau_j^{(l)}$ are the position and token of robot u_{i_j} at time $t_0 + l$, respectively. Notice that $\rho_j^{(l)} \in P$ and $\tau_j^{(l)} \in Z$ for all j, l.

Let $X(l) = ((\rho_1^{(l)}, \tau_1^{(l)}), \ldots, (\rho_m^{(l)}, \tau_m^{(l)}))$. Note that $X(l)$ describes the state of the system at time $t_0 + l$, a kind of snapshot of the system at time $t_0 + l$. Let us analyze the infinite sequence $\mathcal{X} = X(0), X(1), X(2), \ldots$. The values of this sequence are the columns of Table 2. Now, recall that $(\rho_j^{(l)}, \tau_j^{(l)})$ is in $P \times Z$ for all l, j, then $X(l)$ is in $(P \times Z)^m$ for all l. Notice that $|P| = n$ and $|Z| = m$, thus $|(P \times Z)^m| = (nm)^m$. It follows, using the pigeonhole principle, that the sequence $\mathcal{X}$ repeats values. Let σ be a natural number such that $X(\sigma)$ is the first repeated value in $\mathcal{X}$. That is, there is a value $l' \in \mathbb{N}$, $0 \leq l' < \sigma$ such that $X(l') = X(\sigma)$, see Table 2.

We claim that $l' = 0$. Suppose that $l' > 0$. As a consequence of the behavior of a partial SCS it is easy to show that $X(l' + 1) = X(\sigma + 1)$ and $X(l' - 1) = X(\sigma - 1)$.

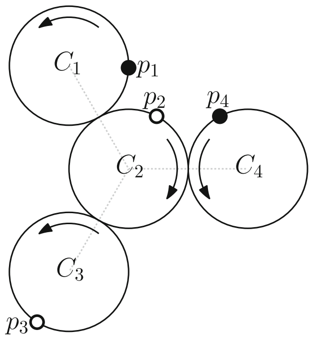

Fig. 19 A 2-partial SCS at time t_0, the solid points are the positions occupied by the surviving robots and the hollow points are the positions where a removed robot should be

Table 2 The jth row represents the sequence of transitions of the robot u_{i_j} at times $t_0, t_0 + 1, t_0 + 2, \ldots$

	0	1	$\cdots$	σ	$\cdots$
u_{i_1}	$(\rho_1^{(0)}, \tau_1^{(0)})$	$(\rho_1^{(1)}, \tau_1^{(1)})$	$\cdots$	$(\rho_1^{(\sigma)}, \tau_1^{(\sigma)})$	$\cdots$
u_{i_2}	$(\rho_2^{(0)}, \tau_2^{(0)})$	$(\rho_2^{(1)}, \tau_2^{(1)})$	$\cdots$	$(\rho_2^{(\sigma)}, \tau_2^{(\sigma)})$	$\cdots$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	
u_{i_m}	$(\rho_m^{(0)}, \tau_m^{(0)})$	$(\rho_m^{(1)}, \tau_m^{(1)})$	$\cdots$	$(\rho_m^{(\sigma)}, \tau_m^{(\sigma)})$	$\cdots$

Then the value $X(\sigma - 1)$ is the first repeated value in $\mathcal{X}$, a contradiction! Therefore $l' = 0$ and σ is the period of the system, i.e., $X(l) = X(l + \sigma)$ for all $l \in \mathbb{N}$. □

Lemma 14 *Let $\mathcal{F}$ be an m-partial SCS and assume that no more robots are removed from $\mathcal{F}$. Let $\mathcal{T}_{\mathcal{F}}$ be the token graph of $\mathcal{F}$. If at time t a robot u is bearing a token z then at time $t' > t$ robot u is bearing a token z' which is in the same connected component of z in $\mathcal{T}_{\mathcal{F}}$.*

Proof From the nature of the token graph, we know that if a robot u with token τ exchanges its token with a robot u' with token τ', then τ and τ' are adjacent in the token graph. Then, the sequence of token's exchanges made by u from time t to time t' defines a path in $\mathcal{T}_{\mathcal{F}}$. □

Lemma 15 *Let $\mathcal{F}$ be an m-partial SCS and assume that no more robots are removed from the system. Let $\mathcal{T}_{\mathcal{F}}$ be the token graph of $\mathcal{F}$. Let u and u' be robots in $\mathcal{F}$ bearing tokens z and z' at time t, respectively. A message sent by u at time t is delivered to u' if and only if z and z' are in the same connected component of $\mathcal{T}_{\mathcal{F}}$.*

Proof ($\Rightarrow$) Suppose that a message sent by u at time t (hence, associated with z) is delivered to u' at time t'. Let τ be the token held by u' at time t'. It is easy to see that if τ contains the message then there is a path in $\mathcal{T}_{\mathcal{F}}$ between z and τ. Thus, from Lemma 14, we deduce that z and z' are in the same connected component of $\mathcal{T}_{\mathcal{F}}$.

($\Leftarrow$) Suppose that z and z' are in the same connected component of $\mathcal{T}_{\mathcal{F}}$. Thus, the message in z will be delivered to z' at some time. From Lemma 13, we deduce that the message will reach u'. □

Remark 9 From the previous lemma we deduce that if it is possible to send a message at time t from robot u to robot u', then it is also possible to send a message at time t from u' to u.

From the previous remark and Lemma 15 we arrive to the following corollary.

Corollary 1 *Let $\mathcal{F}$ be an m-partial SCS and assume that no more robots are removed from the system. Let $\mathcal{T}_{\mathcal{F}}$ be the token graph of $\mathcal{F}$. There is a loss of connectivity in $\mathcal{F}$ if and only if $\mathcal{T}_{\mathcal{F}}$ is disconnected or if it consists of a single vertex.*

Let $G = (V, E)$ be a connected graph. A *separator set* is a subset $S \subset V$ such that the subgraph induced by $V \backslash S$ is disconnected. The *connectivity* of G, denoted by $\kappa(G)$ is $|V| - 1$ if $G = K_n$ (the complete graph); otherwise, $\kappa(G)$ is the cardinality of a smallest separator set.

From Lemma 6, Corollary 1 and Henzinger et al. (2000), the next result follows.

Lemma 16 *Let $\mathcal{F}$ be a SCS and let $\mathcal{T}_{\mathcal{F}}$ be the token graph of $\mathcal{F}$. The broadcasting resilience of $\mathcal{F}$ is $\kappa(\mathcal{T}_{\mathcal{F}})$ and it can be computed in $O(n\kappa(\mathcal{T}_{\mathcal{F}}) \cdot \min\{\kappa^3(\mathcal{T}_{\mathcal{F}}) + n, n\kappa(\mathcal{T}_{\mathcal{F}})\})$ time.*

By using the definitions of the resilience measures we have.

Corollary 2 *The broadcasting resilience of a SCS $\mathcal{F}$ is less than or equal to its 1-isolation resilience.*

In Bereg et al. (2018), it is shown that the 1-isolation resilience of a SCS can be computed in $\tilde{O}(n)$, therefore, we can compute a bound for the broadcasting resilience of a SCS in almost linear time. The next subsections focuses on some configurations that appear in practical applications: trees, cycles and grids.

6.1 Computing the broadcasting resilience on SCS's with one ring

We start this subsection with the definition of a specific and widely studied type of graph: the *circulant graph*.

Definition 12 Let $V = \{0, 1, \ldots, n-1\}$. A graph $G = (V, E)$ is *circulant* if for all $d \in \mathbb{Z} : \exists i \in V\ \{i, (i+d) \bmod n\} \in E \Rightarrow \forall j \in V\ \{j, (j+d) \bmod n\} \in E$.

Lemma 17 *Let $\mathcal{F}$ be a SCS with a single ring. The token graph of $\mathcal{F}$ is circulant.*

Proof Let r be the ring of the system. Enumerate the tokens in $\mathcal{F}$ from 0 to $n - 1$ following the travel direction in r. Let $d \in \mathbb{Z}$ be an arbitrary value. Suppose that there is a correspondence between two tokens i and $(i + d) \bmod n$. Let $j \neq i$ be any other token. We prove that there is a correspondence between tokens j and $(j+d) \bmod n$. Let $0 < l < n$ be a natural number such that $i = (j+l) \bmod n$. Notice that $(i+d) \bmod n = (j + d + l) \bmod n$. Fix an arbitrary time t. Due to the way in which we enumerate the tokens, there exist two paths p and p' of the same length $2l\pi$, from j to i and from $(j+d) \bmod n$ to $(i+d) \bmod n$, respectively (Remark 4). From the definition of correspondence between tokens, it follows that at time t there are two paths, q and q', of the same length from i and $(i + d) \bmod n$ respectively, to a common link position ℓ. Then, there is a correspondence between j and $(j + d) \bmod n$ because of the paths $p + q$ and $p' + q'$ (where + indicates concatenation). □

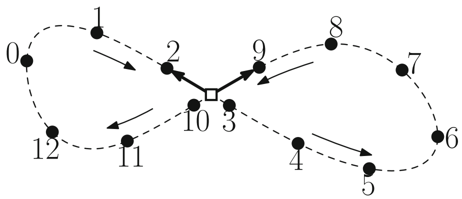

Fig. 20 The dashed stroke represent a ring r. The solid points are the tokens numbered following the travel direction on r. The hollow square represents a link position ℓ. The bold solid arrows from ℓ indicates the travel directions to find the first pair of tokens that will meet at ℓ

Remark 10 Notice that a circulant graph can be encoded as $(n, \{d_1, \ldots, d_m\})$ where n is the number of vertices and the set of d_i values, $0 < d_i \leq n/2$, are the 'jumps' between adjacent nodes, that is, the nodes j and $(j + d_i) \bmod n$ are adjacent for all $0 \leq j < n$.

Lemma 18 *Let $\mathcal{F}$ be a SCS with a single ring. The broadcasting resilience of $\mathcal{F}$ can be computed in $O(n^{3/2})$ time.*

Proof Remark 10 implies that in order to compute $\mathcal{T}_{\mathcal{F}}$ it suffices to compute the set of jumps in the circulant. First, enumerate the tokens from 0 to $n - 1$ following the travel direction in the ring. Then, for every link position ℓ do the following: from ℓ travel in the two opposite directions of the ring until we find the first pair of token i and j that will meet at ℓ, see Fig. 20 for an illustration. In this example, $i = 2$ and $j = 9$. Now take $d = \min(|j - i|, n - |j - i|)$ and add it to the set of jumps. This takes $O(1)$ time. In Fig. 20, $d = 6$, inducing the edges $\{0, 6\}, \{1, 7\}, \ldots, \{5, 11\}, \{6, 12\}, \{7, 0\}, \ldots, \{12, 5\}$. Since the number of link positions in the system is $O(n)$, computing the jumps of the token graph $\mathcal{T}_{\mathcal{F}}$ takes $O(n)$ time. Thus, we encode $\mathcal{T}_{\mathcal{F}}$ by (n, J) where J is the set of jumps. Then, $\kappa(\mathcal{T}_{\mathcal{F}})$ can be computed in $O(n^{3/2})$ time using the algorithm from Boesch and Tindell (1984) and Meijer (1991). By Lemma 16, the result follows. □

Now, since a SCS whose communication graph is a tree has one ring (Lemma 7 of Bereg et al. (2018)), we have that.

Corollary 3 *The broadcasting resilience of a SCS whose communication graph is a tree can be computed in $O(n^{3/2})$ time.*

6.2 The broadcasting resilience on SCS's whose communication graph is a cycle

Lemma 19 *Let $\mathcal{F}$ be a SCS whose communication graph is a cycle. The broadcasting resilience of $\mathcal{F}$ is $\min\{l, l'\}$ where $2l\pi$ and $2l'\pi$ are the lengths of the two rings in $\mathcal{F}$. Moreover, the broadcasting resilience can be computed in linear time.*

Proof From Lemmas 2 and 10 it follows that the token graph of $\mathcal{F}$ is the complete bipartite graph $K_{l,l'}$ and therefore the minimum separator vertex set of $K_{l,l'}$ has cardinality $\min\{l, l'\}$. The linear time algorithm described in Theorem 4 allows us to compute the rings of the system and their lengths. □

Corollary 4 *Let $\mathcal{F}$ be a SCS whose communication graph is a cycle. Let $2l\pi$ and $2l'\pi$ be the lengths of the two rings in $\mathcal{F}$. The three measures, broadcasting resilience,* *coverage resilience and k-isolation resilience (for all* $k \leq \max\{l, l'\}$*), have the same value, which is* $\min\{l, l'\}$.

6.3 The broadcasting resilience on SCS's whose communication graph is a grid

Lemma 20 *Let* $\mathcal{F}$ *be an* $N \times M$ *SCS. The broadcasting resilience of* $\mathcal{F}$ *is* $N + M - 2$.

Proof From Theorem 9 we deduce that the token graph of $\mathcal{F}$ is a complete graph if and only if $N = 1$ or $M = 1$. In any of these cases the lemma holds.

If $N \geq 2$ and $M \geq 2$, the token graph $\mathcal{T}_{\mathcal{F}}$ has a separator set $S \subset V(\mathcal{T}_{\mathcal{F}})$ such that $|S| = \kappa(\mathcal{T}_{\mathcal{F}})$. Now, from Theorem 10 we know that the 1-isolation resilience of $\mathcal{F}$ is $N + M - 2$, then, by using Corollary 2, we have that $\kappa(\mathcal{T}_{\mathcal{F}}) \leq N + M - 2$. Then:

$$|V(\mathcal{T}_{\mathcal{F}})\backslash S| = N \cdot M - \kappa(\mathcal{T}_{\mathcal{F}}) \geq N \cdot M - N - M + 2. \tag{1}$$

In the following we prove that:

$$|V(\mathcal{T}_{\mathcal{F}})\backslash S| \leq N \cdot M - N - M + 2. \tag{2}$$

Let $A \neq \emptyset$ and $B \neq \emptyset$ form a bipartition of $V(\mathcal{T}_{\mathcal{F}})\backslash S$ such that $\{a, b\} \notin E(\mathcal{T}_{\mathcal{F}})$ for all $a \in A$ and $b \in B$. Let $\texttt{rows}(A) \subset \{1 \ldots N\}$ and $\texttt{rows}(B) \subset \{1 \ldots N\}$ denote the sets of rows occupied by the tokens in A and B, respectively. Analogously, let $\texttt{cols}(A) \subset \{1 \ldots M\}$ and $\texttt{cols}(B) \subset \{1 \ldots M\}$ denote the sets of columns occupied by the tokens in A and B, respectively. Notice that $\texttt{rows}(A) \cap \texttt{rows}(B) = \texttt{cols}(A) \cap \texttt{cols}(B) = \emptyset$.

Take $r \in \texttt{rows}(A)$ and $c \in \texttt{cols}(A)$, let z be the token of $\mathcal{F}$ on the circle of row r and column c. If z is not in A then z is in S. Let $A' = A \cup \{z\}$ and $S' = S\backslash\{z\}$. Notice that A' and B form a bipartition of $V(\mathcal{T}_{\mathcal{F}})\backslash S'$ and there is no edge from A' to B. Therefore, S' is a separator set and $|S'| < |S|$ which is a contradiction. Thus, for every $r \in \texttt{rows}(A)$ and $c \in \texttt{cols}(A)$ the token on the circle (r, c) is in A and $|A| = |\texttt{rows}(A)| \cdot |\texttt{cols}(A)|$. Analogously, we can prove that $|B| = |\texttt{rows}(B)| \cdot |\texttt{cols}(B)|$. To simplify the notation, let $|\texttt{rows}(A)| = r_A$, $|\texttt{cols}(A)| = c_A$, $|\texttt{rows}(B)| = r_B$ and $|\texttt{cols}(B)| = c_B$. Then: $|V(\mathcal{T}_{\mathcal{F}})\backslash S| = |A| + |B| = r_A \cdot c_A + r_B \cdot c_B$.

Since $r_B \leq N - r_A$ and $c_B \leq M - c_A$, then:

$$\begin{aligned} &r_A \cdot c_A + r_B \cdot c_B - N \cdot M + N + M - 2 \\ &\quad \leq r_A \cdot c_A + (N - r_A) \cdot (M - c_A) - N \cdot M + N + M - 2. \end{aligned}$$

The right part of the above inequality can be rewritten as:

$$(1 - c_A)(N - r_A - 1) + (1 - r_A)(M - c_A - 1).$$

Notice that $(1 - c_A) \leq 0$ and $(1 - r_A) \leq 0$. On the other hand, $(N - r_A - 1) \geq 0$ and $(M - c_A - 1) \geq 0$. This proves inequality (2) and the stated lemma follows from inequalities (1) and (2). $\square$

7 Conclusion and open problems

Area coverage in cooperative robot networks is a fundamental component of many applications. For instance, a group of UAVs can form a network and accomplish complicated missions such as rescue, searching, patrolling and mapping. There are two main issues which must be considered when developing a solution with a cooperative robot network: coverage and communication.

In this paper we propose efficient algorithms for computing various parameters of a synchronized system of robots related to the robustness of the network in the presence of failures. With respect to communication between robots, we considered two quality measures: the k-isolation resilience, introduced in Bereg et al. (2018) and a new measure, the broadcasting resilience. The k-isolation resilience is the minimum number of robots whose removal may cause the starvation (isolation) of at least k surviving robots. Although the problem of computing this measure is NP-hard in general, even for trees, we showed how to solve the problem when the communication graph is a cycle (in linear time) or a grid (with an analytic expression). The broadcasting resilience is related to the capacity of the system for sending messages through the network. We showed how to efficiently compute this measure on trees, cycles and grids. For the study of the robustness with respect to area coverage, we introduced the concept of coverage resilience as the minimum number of robots whose removal may result in at least one non-covered trajectory segment. Notice that if we denote this measure as r_c, the removal of at most $r_c - 1$ robots from the system guarantees that the total area is covered. In order to bound the idle-time of the system, that is, the maximum time a point of the area is unattended by the robots, we also define the T-coverage resilience as the minimum number of robots we can remove so that the idle-time is at least T. We gave a linear time algorithm to compute these values of resilience for a general communication graph and we found closed form solutions depending on the input size for trees and grid-graphs. Moreover, we showed some relationship between the parameters. For example, the three measures match when the communication graph is a cycle.

The main open problem suggested by this paper is to improve the complexity of the broadcasting resilience problem in general. Another research line is to consider randomization in the shifting process when one or more robots fail. The idea of using random walks of $k < n$ robots shifting to a neighboring trajectory at the communication link with probability p results in some interesting open problems. These include the computation of the expected *meeting time* (Tetali and Winkler 1991) or the *hitting time* (Patel et al. 2016) of random walks in a partial SCS. Both problems are suggested by the concept of k-isolation resilience. Related to the coverage resilience, it would be interesting to study the expected *idle-time* of the system, that is, the expected time for a point in the terrain to be visited. Finally, another interesting task is to calculate the values of the shifting probability p that maximize the resilience measures.

References

Abdulla AEAA, Fadlullah ZMd, Nishiyama H, Kato N, Ono F, Miura R (2014) An optimal data collection technique for improved utility in uas-aided networks. In: 2014 IEEE conference on computer communications, INFOCOM 2014, pp 736–744

Alena O, Niels A, James C, Bruce G, Erwin P (2018) Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: a survey. Networks, to appear

Almeida A, Ramalho G, Santana H, Tedesco P, Menezes T, Corruble V, Chevaleyre Y (2004) Recent advances on multi-agent patrolling. In: Brazilian symposium on artificial intelligence. Springer, pp 474–483

Bereg S, Caraballo L-E, Díaz-Báñez J-M, Lopez MA (2018) Computing the k-resilience of a synchronized multi-robot system. J Comb Optim 36(2):365–391

Boesch F, Tindell R (1984) Circulants and their connectivities. J Graph Theory 8(4):487–499

Choset H (2001) Coverage for robotics-a survey of recent results. Ann Math Artif Intell 31(1–4):113–126

Clark CM, Rock SM, Latombe J-C (2003) Motion planning for multiple mobile robots using dynamic networks. In: Proceedings of the 2003 IEEE international conference on robotics and automation, ICRA'03, vol 3. IEEE, pp 4222–4227

Collins A, Czyzowicz J, Gasieniec L, Kosowski A, Kranakis E, Krizanc D, Martin R, Ponce OM (2013) Optimal patrolling of fragmented boundaries. In: Proceedings of the twenty-fifth annual ACM symposium on Parallelism in algorithms and architectures. ACM, pp 241–250

Czyzowicz J, Gasieniec L, Kosowski A, Kranakis E (2011) Boundary patrolling by mobile agents with distinct maximal speeds. In: European symposium on algorithms. Springer, pp 701–712

Czyzowicz J, Kosowski A, Kranakis E, Taleb N (2016) Patrolling trees with mobile robots. In: International symposium on foundations and practice of security. Springer, pp 331–344

Díaz-Báñez J-M, Caraballo L-E, Lopez M, Bereg S, Maza I, Ollero A (2015) The synchronization problem for information exchange between aerial robots under communication constraints. In: 2015 International conference on robotics and automation (ICRA). IEEE

Díaz-Báñez J-M, Caraballo L-E, Lopez M, Bereg S, Maza I, Ollero A (2017) A general framework for synchronizing a team of robots under communication constraints. IEEE Trans Robotics 33(4):748–755

Flocchini P, Santoro N, Viglietta G, Yamashita M (2016) Rendezvous with constant memory. Theor. Comput. Sci. 621:57–72

Galceran E, Carreras M (2013) A survey on coverage path planning for robotics. Robotics Auton Syst 61(12):1258–1276

Hamacher HW (1992) Combinatorial optimization models motivated by robotic assembly problems. In: Combinatorial optimization. Springer, pp 187–198

Hazon N, Kaminka GA (2008) On redundancy, efficiency, and robustness in coverage for multiple robots. Robotics Auton Syst 56(12):1102–1114

Henzinger MR, Rao S, Gabow HN (2000) Computing vertex connectivity: new bounds from old techniques. J Algorithms 34(2):222–250

Hsieh MA, Cowley A, Kumar V, Taylor CJ (2008) Maintaining network connectivity and performance in robot teams. J Field Robotics 25(1–2):111–131

Hwang S-I, Cheng S-T (2001) Combinatorial optimization in real-time scheduling: theory and algorithms. J Comb Optim 5(3):345–375

Kranakis E, Krizanc D (2015) Optimization problems in infrastructure security. In: International symposium on foundations and practice of security. Springer, pp 3–13

Lovász L (1993) Random walks on graphs. Comb Paul Erdos Eighty 2(1–46):4

Mazayev A, Correia N, Schütz G (2016) Data gathering in wireless sensor networks using unmanned aerial vehicles. Int J Wireless Inf Netw 23(4):297–309

Meijer PT (1991) Connectivities and diameters of circulant graphs. Master's thesis, Simon Fraser University, 12

Patel R, Carron A, Bullo F (2016) The hitting time of multiple random walks. SIAM J Matrix Anal Appl 37(3):933–954

Roy N, Dudek G (2001) Collaborative robot exploration and rendezvous: algorithms, performance bounds and observations. Auton Robots 11(2):117–136

Tabachnikov S (2005) Geometry and billiards. Amer. Math. Soc, Providence

Tetali P, Winkler P (1991) On a random walk problem arising in self-stabilizing token management. In: Proceedings of the tenth annual ACM symposium on principles of distributed computing. ACM, pp 273–280

Winfield AFT (2000) Distributed sensing and data collection via broken ad hoc wireless connected networks of mobile robots. In: Distributed autonomous robotic systems, vol 4. Springer, pp 273–282

Affiliations

Sergey Bereg[1] · Andrew Brunner[2] · Luis-Evaristo Caraballo[3] · José-Miguel Díaz-Báñez[3] · Mario A. Lopez[2]

✉ Luis-Evaristo Caraballo
lcaraballo@us.es

1 Department of Computer Science, University of Texas at Dallas, 800 West Campbell Road, Richardson, TX 75080, USA

2 Department of Computer Science, University of Denver, 2155 E. Wesley Avenue, Denver, CO 80208, USA

3 Higher Technical School of Engineering, University of Seville, Camino de los Descubrimientos, No number, 41092 Seville, Spain