Autonomous Mobile Robots: A Distributed Computing Perspective

Giuseppe Prencipe(✉)

Dipartimento di Informatica, Università di Pisa, Pisa, Italy
prencipe@di.unipi.it

Abstract. The distributed coordination and control of a team of autonomous mobile robots is a problem widely studied in a variety of fields, such as engineering, artificial intelligence, artificial life, robotics. Generally, in these areas, the problem is studied mostly from an empirical point of view.

Recently, the study of what can be computed by such team of robots has become increasingly popular in theoretical computer science and especially in distributed computing, where it is now an integral part of the investigations on computability by mobile entities [28]. In this paper we describe the current investigations on the algorithmic limitations of what autonomous mobile robots can do with respect to different coordination problems, and overview the main research topics that are gaining attention in this area.

1 Introduction

For the last twenty years, the major trend in robotic research, both from engineering and behavioral viewpoints, has been to move away from the design and deployment of few, rather complex, usually expensive, application-specific robots. In fact, the interest has shifted towards the design and use of a large number of "generic" robots which are very simple, with very limited capabilities and, thus, relatively inexpensive, but capable, together, of performing rather complex tasks.

The advantages of such an approach are clear and many, including: reduced costs; ease of system expandability which in turns allows for incremental and on-demand deployment; simple and affordable fault-tolerance capabilities; reusability of the robots in different applications [26,49].

One of the first studies conducted in this direction in the AI community is that of Matarić [44]. The main idea in Matarić's work is that "interactions between individual agents need not to be complex to produce complex global consequences".

Other investigations in the AI community include the study of [4] on stigmergy communication and on the use a set of simple robots that operate completely autonomously and independently to collect pucks spread over a square

This research is supported in part by MIUR of Italy under project ARS TechnoMedia.

P. Flocchini et al. (Eds.): ALGOSENSORS 2013, LNCS 8243, pp. 6–21, 2014.
DOI: 10.1007/978-3-642-45346-5_2,

arena in a single cluster; the ALLIANCE architecture and the studies on selfish behavior of cooperative robots in animal societies by Parker [49]; the formation and navigation problems in multi-robot teams in the context of primitive animal behavior in pattern formation by Balch and Arkin [3]; and the experiments in cooperative cleaning behavior of Jung et al. [38].

Alternative approaches to the problem of studying multi-robot systems, can be found in the CEBOT system of Fucuda, Kawaguchi et al. [32,41], in the planner-based architecture of Noreils [47], in the information requirements theory of Donald et al. [26] (see [7] for a survey), in the Swarm Intelligence of Beni and Hackwood [5], in the Self-Assembly Machine ("fructum") of Murata et al. [46], etc.

The common feature of all these approaches is that they do not deal with formal correctness of the solutions, that are only analyzed empirically. In all these investigations, algorithmic aspects were somehow implicitly an issue, but clearly not a major concern, let alone the focus, of the study. An investigation with an algorithmic flavor has been undertaken within the AI community by Durfee [27], who argues in favor of limiting the knowledge that an intelligent robot must possess in order to be able to coordinate its behavior with others.

More recently, the study of teams of autonomous mobile robots has gained attention also in distributed computing area, keeping pace with the trend originally started in robotics and AI. However, here the problem has been tackled from a different perspective: from a *computational* point of view. In other words, the focus is to understand the relationship between the capabilities of the robots and the solvability of the tasks they are given. In these studies, the impact of the *knowledge* of the environment is analyzed: can the robots form an arbitrary geometric pattern if they have a *compass*? Can they gather in a point? Which information each robot must have about its fellows in order for them to collectively achieve their goal? The goal is to look for the minimum power to give to the robots so that they can solve a given task; hence, to formally analyze the strengths and weaknesses of the distributed coordination and control.

In this paper we describe the current investigations on the interplay between robots capabilities, computability, and algorithmic solutions of coordination problems by autonomous mobile robots.

2 Modeling Autonomous Mobile Robots

The considered computational universe is a 2-dimensional plane populated by a set of n autonomous mobile robots, denoted by $r_1, \ldots, r_n$, that are modeled as devices with computational capabilities which are able to freely move on a two-dimensional plane.

The Robots and Their Behavior. A robot is a computational unit capable of sensing the positions of other robots in its surrounding, performing local computations on the sensed data, and moving towards the computed destination. The local computation is done according to a deterministic algorithm that takes in input the sensed data (i.e., the robots' positions), and returns a destination point towards which the executing robot moves. All the robots execute the same algorithm. The local view of each robot includes a unit of length, an origin, and a Cartesian coordinate system defined by the *directions* of two coordinate axes, identified as the X and Y axis, together with their *orientations*, identified as the positive and negative sides of the axes. Each robot cyclically performs the following operations: **(i)** *Look*: The robot observes the world by activating its sensors which will return a *snapshot* of the positions of all other robots within the visibility range with respect to its local coordinate system. Each robot is viewed as a point, hence its position in the plane is given by its coordinates, and the result of the snapshot (hence, of the observation) is just a set of coordinates in its local coordinate system: this set forms the *view of the world* of r. **(ii)** *Compute*: The robot performs a *local computation* according to a deterministic algorithm $\mathcal{A}$ (we also say that the robot *executes* $\mathcal{A}$). The algorithm is the same for all robots, and the result of the *Compute* state is a *destination point*. **(iii)** *Move*: If the destination point is the current location of r, r performs a *null movement* (i.e., it does not move); otherwise it moves towards the computed destination but it can stop anytime during its movement.[1]

The robots are completely *autonomous*: no central control is needed. Furthermore they are *anonymous*, meaning that they are a priori indistinguishable by their appearance, and they do not (need to) have any kind of identifiers that can be used during the computation.

Moreover, the robots are *silent*: there are no explicit direct means of communication, and any communication occurs in a totally implicit manner. Specifically, it happens by means of observing the robots' positions in the plane, and taking a deterministic decision accordingly. In other words, the only mean for a robot to send information to some other robot is to move and let the others observe (reminiscent of bees in a bee dance).

Activation and Operation Schedule. With respect to the *activation* schedule of the robots and of the timing of the *operations* within their cycles, there are two main models, asynchronous and semi-synchronous.

In the asynchronous model (Async), no assumptions on the cycle time of each robot, and on the time each robot takes to execute each state of a given cycle are made [29]. It is only assumed that each cycle is completed in finite time, and that the distance traveled in a cycle is finite. Moreover, the robots do not need to have a common notion of time, and each robot can execute its actions at unpredictable time instants.

More precisely, there are only two limiting assumptions. The first one refers to space; namely, the distance traveled by a robot during a computational cycle. **(A1)** *The distance traveled by a robot r in a move is not infinite. Furthermore, there exists an arbitrarily small constant $\delta_r > 0$, such that if the destination point*

[1] e.g. because of limits to the robot's motorial capabilities.

is closer than δ_r, *r will reach it; otherwise, r will move towards it of at least* δ_r. As no other assumptions on space exist, the distance traveled by a robot in a cycle is unpredictable.

The second limiting assumption is on the length of a cycle. **(A2)** *The amount of time required by a robot r to complete a computational cycle is not infinite. Furthermore, there exists a constant* $\varepsilon_r > 0$ *such that the cycle will require at least* ε_r *time.* As no other assumption on time exists, the resulting system is *fully asynchronous* and the duration of each activity (or inactivity) is unpredictable; this setting is usually denoted by Async.

There are two important consequences: First, since the time that passes after a robot starts observing the positions of all others and before it starts moving is arbitrary, but finite, the actual move of a robot may be based on a situation that was observed arbitrarily far in the past, and therefore it may be totally different from the current situation. Second, since movements can take a finite but unpredictable amount of time, and different robots might be in different states of their cycles at a given time instant, it is possible that a robot can be seen *while* it is moving by other robots that are observing.[2]

In the *semi-synchronous* (Ssync) model, the activations of the robots is logically divided into global rounds; in each round, one or more robots are activated and obtain the same snapshot; based on that snapshot, they compute and perform their move [57].

In particular, there is a global clock tick reaching all robots simultaneously, and a robot's cycle is an instantaneous event that starts at a clock tick and ends by the next. The only unpredictability is given by the fact that at each clock tick, every robot is either *active* or *inactive*, and only active robots perform their cycle. The unpredictability is restricted by the fact that at least one robot is active at every time instant, and every robot becomes active at infinitely many unpredictable time instants. A very special case is when every robot is active at every clock tick; in this case the robots are *fully synchronized* (this specific setting is usually denoted by Fsync).

In this setting, at any given time, all active robots are executing the same cycle state; thus no robot will look while another is moving. In other words, a robot observes other robots only when they are stationary. This implies that the computation is always performed based on accurate information about the current configuration. Furthermore, since no robot can be seen *while* it is moving, the movement can be considered *instantaneous*. An additional consequence of atomicity and synchronization is that, for them to hold, the maximum distance that a robot can move in one cycle is bounded.

Capabilities. Different settings arise from different assumptions that are made on the robots' capabilities, and on the amount of information that they share and use during the accomplishment of the assigned task. In particular,

[2] Note that this does not mean that the observing robot can distinguish a moving robot from a non moving one.

- **Visibility.** The robots may be able to sense the complete plane or just a portion of it. We will refer to the first case as the *Unlimited Visibility* case. In contrast, if each robot can sense only up to a distance $V > 0$ from it, we are in the *Limited Visibility* case. In the following, we will say also that the robots have unlimited/limited visibility.
- **Geometric Agreement.** Each robot r has its own unit of length, and a *local compass* defining a local Cartesian coordinate system defined by the *directions* of two coordinate axes, identified as the X and Y axis, together with their *orientations*, identified as the positive and negative sides of the axes. This local coordinate system is self-centric, i.e. the origin is the position of the observing robot. Depending on the level of consistency among the robots on the direction and orientation of the axes of their local compasses, different classes of *global* geometric agreement can be identified: *total agreement* (or *consistent compass*), when the robots agree on the direction and orientation of both axes; *partial agreement* (or *one axis*) when all robots agree on the direction and orientation of only one axis; *chirality* when the robots agree on the orientation of the axes (i.e., clockwise); and *no agreement* (or *disorientation*), where no consistency among the local coordinate systems is known to exist.
- **Memory.** The robots can access local memory to store different amount of information regarding the positions in the plane of their fellows. In the *oblivious* model, all the information contained in the workspace is *cleared* at the end of each cycle. In the *non-oblivious* (or *persistent memory*) model, part (or all) of the local memory is *legacy*: unless explicitly erased by the robot, it will persist throughout the robot's cycles. In this model, an important parameter is the *size* of the persistent workspace. One extreme is the *unbounded memory* case, where no information is ever erased; hence robots can remember all past computations and actions. On the opposite side is the case when the size of the persistent workspace is *constant*; in this case, the entities are just Finite-State Machines, and are called *finite-state* robots.

Let us stress that the only means for the robots to coordinate is the observation of the others' positions and their change through time. For oblivious robots, even this form of communication is impossible, since there is no memory of previous positions.

3 Static Problems

Pattern Formation. The Pattern Formation problem is one of the most important coordination problem and has been extensively investigated in the literature (e.g., see [10,56,57,60]). The problem is practically important, because, if the robots can form a given pattern, they can agree on their respective roles in a subsequent, coordinated action.

In its most general definition, the robots are required to form an *arbitrary* pattern. The geometric pattern to be formed is a set of points (given by their Cartesian coordinates) in the plane, and it is initially known by all the robots in the system.

The robots are said to *form the pattern* if, at the end of the computation, the positions of the robots coincide, in everybody's local view, with the points of the pattern. The formed pattern may be *translated*, *rotated*, *scaled*, and *flipped* into its mirror position with respect to the initial pattern. Initially the robots are in arbitrary positions, with the only requirement that no two robots are in the same position, and that, of course, the number of points prescribed in the pattern and the number of robots are the same.

The basic research questions are which patterns can be formed, and how they can be formed. Many proposed procedures do not terminate and never form the desired pattern: the robots just converge towards it; such procedures are said to *converge*.

There exists solution to solve this problem in both Async (e.g., [31]) and Ssync (e.g., [57]), by always considering robots with unlimited visibility. In all the solutions, the kind of patterns that can be formed by the robots depends on the level of agreement the robots have on their local coordinate systems.

Several studies also investigated on the formation of specific patterns, such as lines and circles. In the Line Formation problem, the robots are required to place themselves on a line, whose position is not prescribed in advance (if $n = 2$, then a line is always formed). In [15], this problem has been tackled by studying an apparently totally different problem: the *spreading*. In this problem, the robots, that at the beginning are arbitrarily placed on the plane, are required to evenly spread within the perimeter of a given region. In their work, the authors focus on the one-dimensional case: in this case, the robots have to form a line, and place themselves uniformly on it. A very interesting aspect of the study, is that [15] addresses the issue of *local algorithms*: each robots decides where to move based on the positions of its close neighbors. In particular, in the case of the line, the protocol is quite simple: each robot r observes its left and right neighbor. If r does not see any robot, it simply does not move; otherwise, it moves to the median point between its two neighbors. The authors prove its convergence in Ssync. Furthermore, if each robot knows the exact number of robots at each of its sides, it is possible to achieve the spreading in one dimension in a finite number of cycles.

In the Circle Formation problem, the robots want to place themselves on the plane to form a non degenerated circle of a given diameter.[3] One of the first discussion on circle formation by a group of mobile entities was by Debest [20], who introduced it as an illustration of self-stabilizing distributed problems, but did not provide an algorithm. This problem was first studied by Sugihara and Suzuki [56]. They presented an heuristic distributed protocol,

[3] If the diameter is not fixed a priori, the problem becomes trivial, even in Async: each robot computes the smallest circle enclosing all the robots' positions and moves on the circumference of such a circle.

successively improved by Tanaka [58], that allowed the robots to form an approximation of a circle (more similar to a Reuleaux triangle) having a given diameter D. A variant of this problem is the Uniform Circle Formation problem: the n robots on the plane must be arranged at regular intervals on the boundary of a circle. This kind of formation can be usefully deployed in surveillance tasks: the robots are placed on the border of the area (or around the target) to surveil (e.g., see [34]). Both problems have since been extensively investigated in Ssync and Async [8,21–24,39,52,58].

Gathering. In the Gathering problem, the robots, initially placed in arbitrary and distinct positions, are required to gather in a single location within finite time. This problem is also called *point formation*, *homing*, or *rendezvous*. A problem closely related to Gathering is that of Convergence, where the robots need to be arbitrarily close to a common location, without the requirement of ever reaching it.

In spite of their apparent simplicity, these problems have been investigated extensively both in Ssync and in Async under a variety of assumptions on the robots' capabilities: in fact, several factors render this problem difficult to solve. First of all, some basic results about Gathering: It is possible in Fsync, with an algorithm that exploits the properties of the center of gravity of the team [13]; it is impossible without additional assumptions in Ssync, hence in Async [51,57], and trivially achievable even in Async with totally agreement on the coordinate systems (gather at the position occupied by the rightmost and topmost robot).

Rendezvous. When the system contains only *two* robots, the Gathering problem is very special, and it is often called Rendezvous. We have just stated that, with a common coordinate system, there is an easy solution to Gathering, and hence to Rendezvous even in Async. In absence of a common coordinate system the problem is not solvable even in Ssync. Hence, with $n = 2$, the focus is on gathering in Fsync, and on the Convergence problem.[4]

The Rendezvous has been extensively studied by assuming different level of agreement on the compass systems of the robots. In particular, the problem is solvable in Async when the robots agree on chirality, but the axis are however tilted up to a $\phi < \frac{\pi}{2}$ degrees [37], and the tilt is fixed. If the robots still agree on chirality, but the tilt of their compasses might be variable, rendezvous can be achieved in Ssync with *fully variable* compasses if and only if $\phi < \frac{\pi}{4}$, and in Async with *semi-variable* compasses[5] if and only if $\phi < \frac{\pi}{6}$ [37].

[4] Notice that Rendezvous has a trivial solution in Fsync: a robot moves to the halfway point to the other robot. In both Ssync [57] and Async [13], this *move-to-half* strategy guarantees only *convergence*.

[5] The tilted compasses are said to be *fully variable* if the actual tilt of each compass may vary at any time (but always with no more than ϕ from the global coordinate system); they are *semi-variable* if the tilt of each compass may vary (but no more than ϕ) between successive cycles, but it does not change during a cycle.

Gathering and Convergence. The GATHERING problem has been extensively investigated both experimentally and theoretically in the *unlimited visibility* setting, that is assuming that the entities are capable to sense the entire space. As stated above, when no additional assumptions are made in the model, there is no deterministic solution to the GATHERING problem in SSYNC. However, CONVERGENCE is possible even in ASYNC: The robots get closer to a gathering point, but never reach it in finite time. One quite simple and effective convergence solution in ASYNC exploits the *Center of Gravity* of the robots [13]. With the strongest assumption of *unlimited mobility* (all robots always reach their destinations when performing a *Move*), convergence time in ASYNC can be improved [17].

Thus, the GATHERING problem has a solution only adding additional assumptions. The most common assumption is that of *multiplicity detection*: a robot is able to detect whether a point on the plane is occupied by more than one robot. With this assumption, there exists solutions in both SSYNC [57] and ASYNC [12]. Another capability that has also been considered is a stronger form of multiplicity detection, where robots can detect the exact number of robots located at a given position [25]. Adding this capability, it is impossible to solve the problem for all possible initial configurations containing an even number of robots; however the robots can gather from an arbitrary configuration with n robots, when n is odd. In this case, *initial* configurations include also configurations containing more than one robot on the same point. Note that, since this algorithm is correct starting from all possible configurations provided n is odd (even the ones containing more than one robot), it is truly self-stabilizing.

In contrast, the multiplicity detection is not used in the solution described in [11]; however, it is assumed that the robots can rely on an unlimited amount of memory: the robots are said to be *non-oblivious*. In other words, the robots have the capability to store the results of all computations since the beginning, and freely access to these data and use them for future computations.

Furthermore, in SSYNC agreement on chirality and unlimited mobility suffice for making the problem solvable, even with variable tilted compasses, if the tilt of the local compasses is $\phi < \frac{\pi}{4}$ [36];

A different setting that has been studied is when robots have *limited visibility*: in this scenario, an obvious necessary condition to solve the problem, is that at the beginning of the computation the *visibility graph* (having the robots as nodes and an edge (r_i, r_j) if r_i and r_j are within viewing distance) is connected [2,30]. In [2] the proposed protocol solves the CONVERGENCE problem. In [30], the authors present an algorithm that let the robots to gather in a finite number of cycles. However, in this case the robots can rely on the presence of a common coordinate system: that is, they share a compass.

With limited visibility, the CONVERGENCE problem has been studied in FSYNC when the robots operate in a *non-convex* region (of which they have no map) [33]; in ASYNC with a limited form of asynchrony [42], where the time spent by a robot in the *Look*, and *Compute* states is bounded by a globally predefined amount, while the time spent in the *Move* state is bounded by a locally predefined quantity (not necessarily the same for each robot); and in ASYNC under a *1-fair scheduler* [40]: Between two successive activations of each robot r, all other robots have been activated at most once (as a consequence, from the moment r observes the current situation to the moment it finishes its movement, no other robot performs more than one *Look*).

The GATHERING problem has been also investigated in the context of robots *failures*. In this context, the goal is for the non-faulty robots to gather regardless of the action taken by the faulty ones. Two types of robot faults were investigated by Peleg et al. [1]: *crash* failure, in which the robot stops any activity and will no longer execute any computational cycle; and the *byzantine* failure, in which the robot acts arbitrarily and possibly maliciously.

In [14] it is analyzed the case of systems where the robots have inaccuracies in sensing the positions of other robots, in computing the next destination point, and in moving towards the computed destination. The authors provide a set of limitations on the amount of inaccuracies allowing convergence; hence, they present an algorithm for convergence under bounded measurement, movement and calculation errors. In [43], the case of *radial errors* has also been considered.

Finally, beside the inaccuracies in the compasses that have already been cited above (*tilted* compasses), with *eventually consistent compasses* (i.e., transient errors on the compasses), the GATHERING problem has also been studied in SSYNC, with robots that agree on chirality: in this case, it has been proven that the robots can gather in finite time [53].

Near-Gathering A problem that is very close to the CONVERGENCE problem is NEAR-GATHERING, where a set of robots with limited visibility, at the beginning arbitrarily placed in the plane on distinct positions, are required to get close enough to each other, without any collisions. In particular, in finite time, the robots are required to move within distance ε from each other for some predefined ε. This problem is particularly useful to overcome the limitations introduced by having robots with limited sensing capabilities: in fact, once they are *close enough*, all robots can see each other, hence they can operate as they had unlimited visibility power. This problem has been recently solved in ASYNC for robots with consistent compass [48].

4 Dynamic Problems: Flocking and Capture

In this set of problems, the robots dynamically move, and there is really no ending in the robots' tasks. Let us consider the FLOCKING problem first: There are mainly two versions of this problem. In the first one, there are two kinds of robots in the environment: the *leader* L, and the *followers* (this scenario is also called *guided flocking*). The leader acts independently from the others, and it can be assumed that it is driven by an human pilot. The followers are required to follow the leader wherever it goes (*following*), while keeping a formation they are given in input (*flocking*). In this context, a formation is simply a pattern described as a set of points in the plane, and all the robots have the same formation in input.

In [35], an algorithm solving this problem in ASYNC has been tested by using computer simulation; the algorithm assumes no agreement. All the experiments demonstrated that the algorithm is well behaved, and in all cases the followers were able to assume the desired formation and to maintain it while following the leader along its route. Moreover, the obliviousness of the algorithm contributes to this result, since the followers do not base their computation on past leader's positions.

In the second version of the problem, also known as *homogeneous flocking*, there is no exogenous source (i.e., no guide) and every robot knows the trajectory: The path along which the flock has to move is known in advance to every robot (e.g., [6,54,55]).

Finally, if the leader is considered an "enemy" or "intruder", and the pattern surrounds it, the problem is known as CAPTURE (or *intruder*). A protocol that assumes no agreement and solves the problem in ASYNC has been presented in [34]. The proposed algorithm exhibits remarkable robustness, and numeric simulations indicate that the intruder is efficiently captured in a relatively short time and kept surrounded after that, as desired. Furthermore, the solution is self-stabilizing. In particular, any external intervention (e.g., if one or more of the cops are stopped, slowed down, knocked out, or simply faulty) does not prevent the completion of the task.

5 New Directions

Computing with Colors. A new direction of investigation that just started being explored is the introduction in the model of some form of direct communication. The first attempt in this direction is in [19], where the robots make *visible* to their fellows $O(1)$ persistent bits [19]: Each robot is equipped with a light bulb that can display a constant numbers of different colors; the colors are visible to all other robots, and are persistent, that is, the light bulbs are not automatically reset at the end of each cycle. Thus, they can be used to remember states and to communicate. Apart from these lights, the robots are oblivious in all other respects.

Studies in this direction just started, and here is a brief summary of the major results obtained so far.

Colored ASYNC *versus* SSYNC. The presence of lights with visible colors is undoubtably a very powerful computational tool even if just constant in number. Indeed, it can overcome the limitations of ASYNC making the robots strictly more powerful than traditional SSYNC robots, as we see in the following. In fact, it has been shown that asynchronous robots with lights are *at least as powerful as* semi-synchronous ones: the proof consists of a protocol that allows to execute any semi-synchronous algorithm in an asynchronous setting, each robot using a light with a constant number of colours [19].

There are problems that robots cannot solve *without* visible bits, even if they are semi-synchronous, but can be solved with $O(1)$ visible bits even if the robots are asynchronous [19]. One such a problem is *rendezvous*, i.e., the gathering of two robots; from previous Sect. 3, we know that this problem is not solvable in SSYNC. However, this problem can be solved if the robots have $O(1)$ colors.

Hence, these two results lead to conclude that asynchronous robots endowed with $O(1)$ *visible lights* are strictly more powerful than semi-synchronous robots without any light [19].

Colored ASYNC *versus* FSYNC. The relationship between FSYNC and *Colored* ASYNC is less understood. What is known is that asynchronous robots, if empowered with both a constant number of visible lights and the ability to remember a single snapshot from the past, become at least as powerful as traditional fully synchronous robots [19].

Interestingly, there are problems that can be solved in ASYNC with three colours and one past snapshot, but are not solvable in FSYNC without additional information. This is the case, for example, of the BLINKING problem, which requires $n > 2$ robots to perform subtasks T_1 and T_2 repeatedly in alternation. In T_1, the robots must form a circle, i.e. each robot lies on a distinct point on the same circle $\mathcal{C}$ of radius $Rad > 0$; while in T_2, the robots must gather at a single point.

The presence of a problem not solvable in FSYNC but solvable in ASYNC with lights and one past snapshot, leads to the following conclusion: Asynchronous robots, endowed with $O(1)$ *visible lights* and able to remember a single *snapshot*, are strictly more powerful than fully-synchronous oblivious robots without any lights [19].

This is to be contrasted with the fact that, without lights, ASYNC robots are not even as powerful as SSYNC, even if they remember an unlimited number of previous snapshots [50].

Solid Robots. In the standard model, the robots are viewed as points, i.e., they are *dimensionless*. An interesting variant of the model is to consider entities that occupy a physical space of some size; that is, the entities have a solid dimension. These robots, called *solid* (or *fat*), are assumed to have a common unit distance and are viewed as circular disks of a given diameter. The disks of two robots can touch but cannot overlap. Moreover, it is assumed that, if during its movement a robot collide with another, its movement stops (*fail-stop collision*).

The robots' visibility might be affected by their solid dimension. If so, two robots r_1 and r_2 can see each other if there exist points x and y in the visibility radius of r_1 and r_2 respectively, such that the segment $[xy]$ does not contain any point of any other robot. Note that if a robot r_1 can see robot r_2, it can see some non-zero arc of its bounding circle and thus it can always compute its centre. Otherwise, if no visibility obstruction occurs, the robots are said to be *transparent*.

Very few problems have been investigated for solid robots. One of these is the GATHERING. Obviously, in the case of solid robots, the definition of gathering needs to be modified.

The robots are said to form a connected configuration in the plane if between any two points of any two robots there exists a polygonal line each of whose points belongs to some robot. Gathering is accomplished if the robots form some connected configuration and they are all visible to each other (and thus are aware that a connected configuration is achieved).

Adding a physical dimension to the robots significantly complicates the task, mainly because of the fact that their "body" can obstruct visibility. An example that shows one of the difficulty is given by a team of 4 robots whose centres are situated on two intersecting non-perpendicular lines, one robot in each of the four half- lines. The obvious algorithm that would work if the robots were points would be to have them move towards the intersection of the two lines, which is invariant under straight moves. However, it is easy to see that an adversary might have two robots meet in their move toward the centre, thus obstructing the view to the other two, without forming a connected configuration. In general, the lack of full visibility due to obstruction, prevent the robots from being able to compute easily an invariant point.

For the gathering of solid robots, currently there are only solutions for very small teams; in fact, no gathering algorithm is known for $n > 4$ non-transparent solid robots [18]. Furthermore, these algorithms are not collision-free and they rely on the fail-stop collision assumption to work.

In [9] it is presented an algorithm that works for $n \geq 5$ robots that are solid but transparent. The robots must be initially placed in an asymmetry configuration (so that a leader can be elected) and the desired gathering pattern is a circular layered structure of robots with the elected leader in the center.

In [16] gathering by solid robots is considered in a different setting. Each robot is given in input the position of the gathering point in its own coordinate system. All robots have the same dimension *dim*, and they are said to be gathered when they form a sphere with minimum radius around the predefined gathering point. Robots have *limited visibility*, large enough to avoid collisions (thus, a visibility radius $V \geq 2 \cdot dim$ is sufficient), and they operate in FSYNC.

Solid robots have been also studied in the context of circle formation, in [56, 58] for robots with unlimited visibility, and in [45] for mobile robots whose vision is not only limited but also *directional*.

Simulation Environments. A promising area of research on these topics is represented by the development of computer simulation environments dedicated to autonomous mobile robots. Several studies can be found in the literature right on this track [2,34,35,56,58]. All these simulation environments are specifically designed and developed for a particular problem: for instance, the one in [58] for the circle formation; the one in [35] for the flocking problem; the one in [34] for the intruder problem.

Recently, there has been a first attempt in designing a modular simulation environment to test and execute generic protocols for the autonomous mobile robots addressed in this paper: SYCAMORE [59]. In this environment, the protocol of a robot is defined as a plugin given in input to the simulation engine, and it can be easily set to simulate both 2D an 3D scenarios.

6 Conclusions

In this paper, we surveyed a number of recent results on the interplay between robots capabilities and solvability of problems. The goal of these studies is to gain a better understanding of the power of distributed control from an algorithmic point of view. The area is quite young, thus still offers many research quests. First, one outstanding theoretical open problem: no solution is still known for the Gathering problem where the robots have limited visibility and no agreement; actually, it is not even clear whether the problem is solvable (a similar problem stands for the Near-Gathering).

Then, operating capabilities of our robots are quite limited: New research directions can be taken by expanding the capabilities of the robots, in the attempt of better modeling the real robots. It would be interesting to look at models where the robots have more complex capabilities, e.g.: the robots have some kind of direct communication capabilities (besides the use of lights); the robots are distinct and externally identifiable; etc. Little is known about the solvability of other problems like spreading and exploration (used to build maps of unknown terrains), about the physical aspects of the models, such as those related to energy saving issues, and about the relationships between geometric problems and classical distributed computations. In the area of reliability and fault-tolerance, lightly faulty snapshots, a limited and directional (i.e., not 360°) range of visibility, obstacles that limit the visibility and that moving robots must avoid or push aside, as well as robots that appear and disappear from the scene clearly are all topics that have not yet been studied.

We believe that investigations in these areas will provide useful insights on the ability of weak robots to solve complex tasks.

Acknowledgements. The author would like to thank Paola Flocchini and Nicola Santoro for their help and suggestions in the preparation of this paper.

References

1. Agmon, N., Peleg, D.: Fault-tolerant gathering algorithms for autonomous mobile robots. SIAM J. Comput. **36**, 56–82 (2006)
2. Ando, H., Oasa, Y., Suzuki, I., Yamashita, M.: A distributed memoryless point convergence algorithm for mobile robots with limited visibility. IEEE Trans. Robot. Autom. **15**(5), 818–828 (1999)
3. Balch, T., Arkin, R.C.: Behavior-based formation control for multi-robot teams. IEEE Trans. Robot. Autom. **14**, 926–939 (1998)
4. Beckers, R., Holland, O.E., Deneubourg, J.L.: From local actions to global tasks: stigmergy and collective robotics. In: Artificial Life IV, 4th International Workshop on the Synthesis and Simulation of Living Systems (1994)
5. Beni, G., Hackwood, S.: Coherent swarm motion under distributed control. In: Proceedings of the DARS'92, pp. 39–52 (1992)
6. Canepa, D., Potop-Butucaru, M.G.: Stabilizing flocking via leader election in robot networks. In: Masuzawa, T., Tixeuil, S. (eds.) SSS LNCS. 2007, vol. 4838, pp. 52–66. Springer, Heidelberg (2007)

7. Cao, Y.U., Fukunaga, A.S., Kahng, A.B., Meng, F.: Cooperative mobile robotics: antecedents and directions. In: International Conference on Intelligent Robots and System, pp. 226–234 (1995)
8. Chatzigiannakis, I., Markou, M., Nikoletseas, S.E.: Distributed circle formation for anonymous oblivious robots. In: Ribeiro, C.C., Martins, S.L. (eds.) WEA 2004. LNCS, vol. 3059, pp. 159–174. Springer, Heidelberg (2004)
9. Chaudhuri, S.G., Mukhopadhyaya, K.: Gathering asynchronous transparent fat robots. In: Janowski, T., Mohanty, H. (eds.) ICDCIT 2010. LNCS, vol. 5966, pp. 170–175. Springer, Heidelberg (2010)
10. Chen, Q., Luh, J.Y.S.: Coordination and control of a group of small mobile robots. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2315–2320 (1994)
11. Cieliebak, M., Gathering non-oblivious mobile robots. In: Proceedings of the 6th Latin American Symposium on Theoretical Informatics, pp. 577–588 (2004)
12. Cieliebak, M., Flocchini, P., Prencipe, G., Santoro, N.: Distributed computing by mobile robots: gathering. SIAM J. Comput. **41**(4), 829–879 (2012)
13. Cohen, R., Peleg, D.: Convergence properties of the gravitational algorithm in asynchronous robot systems. SIAM J. Comput. **34**, 1516–1528 (2005)
14. Cohen, R., Peleg, D.: Convergence of autonomous mobile robots with inaccurate sensors and movements. In: Durand, B., Thomas, W. (eds.) STACS 2006. LNCS, vol. 3884, pp. 549–560. Springer, Heidelberg (2006)
15. Cohen, R., Peleg, D.: Local spreading algorithms for autonomous robot systems. Theor. Comput. Sci. **399**, 71–82 (2008)
16. Cord-Landwehr, A., et al.: Collisionless gathering of robots with an extent. In: Černá, I., Gyimóthy, T., Hromkovič, J., Jefferey, K., Královič, R., Vukolić, M., Wolf, S. (eds.) SOFSEM 2011. LNCS, vol. 6543, pp. 178–189. Springer, Heidelberg (2011)
17. Cord-Landwehr, A., et al.: A new approach for analyzing convergence algorithms for mobile robots. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011, Part II. LNCS, vol. 6756, pp. 650–661. Springer, Heidelberg (2011)
18. Czyzowicz, J., Gasieniec, L., Pelc, A.: Gathering few fat mobile robots in the plane. Theor. Comput. Sci. **410**(6–7), 481–499 (2009)
19. Das, S., Flocchini, P., Prencipe, G., Santoro, N., Yamashita. M.: The power of lights: synchronizing asynchronous robots using visible bits. In: Proceedings of the 32nd International Conference on Distributed Computing Systems (ICDCS), pp. 506–515 (2012)
20. Debest, X.A.: Remark about self-stabilizing systems. Comm. ACM **238**(2), 115–117 (1995)
21. Défago, X., Konagaya, A.: Circle formation for oblivious anonymous mobile robots with no common sense of orientation. In: Workshop on Principles of Mobile Computing, pp. 97–104 (2002)
22. Défago, X., Souissi, S.: Non-uniform circle formation algorithm for oblivious mobile robots with convergence toward uniformity. Theor. Comput. Sci. **396**(1–3), 97–112 (2008)
23. Dieudonné, Y., Labbani-Igbida, O., Petit, F.: Circle formation of weak mobile robots. ACM Trans. Auton. Adapt. Syst. **3**(4), 16:1–16:20 (2008)
24. Dieudonné, Y., Petit, F.: Circle formation of weak robots and Lyndon words. Inf. Process. Lett. **4**(104), 156–162 (2007)
25. Dieudonné, Y., Petit, F.: Self-stabilizing gathering with strong multiplicity detection. Theor. Comput. Sci. **428**(13), 47–57 (2012)

26. Donald, B.R., Jennings, J., Rus, D.: Information invariants for distributed manipulation. Int. J. Robot. Res. **16**(5), 63–73 (1997)
27. Durfee, E.H.: Blissful ignorance: knowing just enough to coordinate well. In: ICMAS, pp. 406–413 (1995)
28. Flocchini, P., Prencipe, G., Santoro, N.: Distributed Computing by Oblivious Mobile Robots. Morgan & Claypool, San Rafeal (2012)
29. Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Hard tasks for weak robots: the role of common knowledge in pattern formation by autonomous mobile robots. In: Proceedings of the 10th International Symposium on Algorithm and Computation, pp. 93–102 (1999)
30. Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Gathering of asynchronous robots with limited visibility. Theor. Comput. Sci. **337**, 147–168 (2005)
31. Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Arbitrary pattern formation by asynchronous oblivious robots. Theor. Comput. Sci. **407**(1–3), 412–447 (2008)
32. Fukuda, T., Kawauchi, Y., Asama, H., Buss. M.: Structure decision method for self organizing robots based on cell structures-CEBOT. In: Proceedings of the IEEE International Conference on Robotics and Automation, vol.2, pp. 695–700 (1989)
33. Ganguli, A., Cortés, J., Bullo, F.: Multirobot rendezvous with visibility sensors in nonconvex environments. IEEE Trans. Robot. **25**(2), 340–352 (2009)
34. Gervasi, V., Prencipe, G.: Robotic Cops: the intruder problem. In: Proceedings of the IEEE Conference on Systems, Man and Cybernetics, pp. 2284–2289 (2003)
35. Gervasi, V., Prencipe, G.: Coordination without communication: the case of the flocking problem. Discrete Appl. Math. **143**, 203–223 (2004)
36. Izumi, T., Katayama, Y., Inuzuka, N., Wada, K.: Gathering autonomous mobile robots with dynamic compasses: an optimal result. In: Pelc, A. (ed.) DISC 2007. LNCS, vol. 4731, pp. 298–312. Springer, Heidelberg (2007)
37. Izumi, T., Souissi, S., Katayama, Y., Inuzuka, N., Defago, X., Wada, K., Yamashita, M.: The gathering problem for two oblivious robots with unreliable compasses. Siam J. Comput. **41**(1), 26–46 (2012)
38. Jung, D., Cheng, G., Zelinsky, A.: Experiments in realising cooperation between autonomous mobile robots. In: ISER (1997)
39. Katreniak. B.: Biangular circle formation by asynchronous mobile robots. In: Proceedings of the 12th International Colloquium on Structural Information and Communication Complexity (2005)
40. Katreniak, B.: Convergence with limited visibility by asynchronous mobile robots. In: Kosowski, A., Yamashita, M. (eds.) SIROCCO 2011. LNCS, vol. 6796, pp. 125–137. Springer, Heidelberg (2011)
41. Kawauchi, Y., Inaba, M., Fukuda, T.: A principle of decision making of cellular robotic system (CEBOT). In: Proceedings of the IEEE Conference on Robotics and Automation, pp. 833–838 (1993)
42. Lin, J., Morse, A.S., Anderson, B.D.O.: The multi-agent rendezvous problem. Part 2: the asynchronous case. SIAM J. Control Optim. **46**(6), 2120–2147 (2007)
43. Martínez, S.: Practical multiagent rendezvous through modified circumcenter algorithms. Automatica **45**(9), 2010–2017 (2009)
44. Matarić, M.J.: Interaction and intelligent behavior. Ph.D. thesis. MIT, May 1994
45. Miyamae, T., Ichikawa, S., Hara, F.: Emergent approach to circle formation by multiple autonomous modular robots. J. Robot. Mechatron. **21**(1), 3–11 (2009)
46. Murata, S., Kurokawa, H., Kokaji, S.: Self-assembling machine. In: Proceedings of the IEEE Conference on Robotics and Automation, pp. 441–448 (1994)
47. Noreils, F.R.: Toward a robot architecture integrating cooperation between mobile robots: application to indoor environment. Int. J. Robot. Res. **12**, 79–98 (1993)

48. Pagli, L., Prencipe, G., Viglietta, G.: Getting close without touching. In: Even, G., Halldórsson, M. (eds.) SIROCCO 2012. LNCS, vol. 7355, pp. 315–326. Springer, Heidelberg (2012)
49. Parker, L.E.: On the design of behavior-based multi-robot teams. J. Adv. Robot. **10**(6), 547–578 (1996)
50. Prencipe, G.: The effect of synchronicity on the behavior of autonomous mobile robots. Theory Comput. Syst. **38**, 539–558 (2005)
51. Prencipe, G.: Impossibility of gathering by a set of autonomous mobile robots. Theor. Comput. Sci. **384**(2–3), 222–231 (2007)
52. Samia, S., Défago, X., Katayama, T.: Convergence of a uniform circle formation algorithm for distributed autonomous mobile robots. In: Journés Scientifiques Francophones (JSF), Tokio, Japan (2004)
53. Souissi, S., Défago, X., Yamashita, M.: Using eventually consistent compasses to gather memory-less mobile robots with limited visibility. ACM Trans. Auton. Adapt. Syst. **4**(1), 1–27 (2009)
54. Souissi, S., Yang, Y., Défago. X.: Fault-tolerant flocking in a k-bounded asynchronous system. In: Proceedings of 12th International Conference on Principles of Distributed Systems (OPODIS), pp. 145–163 (2008)
55. Souissi, S., Yang, Y., Défago, X., Takizawa, M.: Fault-tolerant flocking for a group of autonomous mobile robots. J. Syst. Softw. **84**, 29–36 (2011)
56. Sugihara, K., Suzuki, I.: Distributed algorithms for formation of geometric patterns with many mobile robots. J. Robot. Syst. **13**, 127–139 (1996)
57. Suzuki, I., Yamashita, M.: Distributed anonymous mobile robots: formation of geometric patterns. Siam J. Comput. **28**(4), 1347–1363 (1999)
58. Tanaka, O.: Forming a circle by distributed anonymous mobile robots. Technical report, Department of Electrical Engineering, Hiroshima University, Hiroshima, Japan (1992)
59. Volpi, V.: Sycamore: a 2D–3D simulation environment for autonomous mobile robots algorithms. https://code.google.com/p/sycamore/
60. Wang, P.K.C.: Navigation strategies for multiple autonomous mobile robots moving in formation. J. Robot. Syst. **8**(2), 177–195 (1991)

http://www.springer.com/978-3-642-45345-8

Algorithms for Sensor Systems
9th International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics, ALGOSENSORS 2013, Sophia Antipolis, France, September 5-6, 2013, Revised Selected Papers

Flocchini, P.; Gao, J.; Kranakis, E.; Meyer auf der Heide, F. (Eds.)

2014, X, 307 p. 91 illus., Softcover
ISBN: 978-3-642-45345-8