Distributed Coordination Algorithms for Mobile Robot Swarms: New Directions and Challenges

David Peleg*

Department of Computer Science,
The Weizmann Institute of Science,
Rehovot, Israel
david.peleg@weizmann.ac.il

Abstract. Recently there have been a number of efforts to study issues related to coordination and control algorithms for systems of multiple autonomous mobile robots (also known as *robot swarms*) from the viewpoint of distributed computing. This paper reviews the literature in the area and discusses some open problems and future research directions.

1 Introduction

Mobile robots have been developed for over half a century, beginning in the 1950's with pioneering projects such as Shannon's electromechanical mouse Theseus, Grey Walter's tortoise and Stanford's Shakey (cf. [4]). Applications for such robots abound, including industrial tasks (e.g., moving materials around), military operations (e.g., surveillance or automated supply lines), search and rescue missions, space exploration (e.g., Sojourner's Mars Pathfinder mission in 1997 or the recent automated transfer vehicle project of the European Space Agency), as well as a variety of home applications, from babysitters and pets to smart appliances such as vacuum cleaners and lawn mowers. Mobile robots come in all shapes, sizes and designs, and vary in their motion type, sensors, handling mechanisms, computational power and communication means.

Systems of multiple autonomous mobile robots (often referred to as *robot swarms*) have been extensively studied throughout the past two decades (cf. [17, 8, 24, 27, 12, 5, 41]). The motivating idea is that for certain applications it may be preferable to abandon the use of a single, strong and costly robot in favor of a group of tiny, functionally simple and relatively cheap robots. For instance, it may be possible to use a multiple robot system in order to perform certain tasks that require spreading over a large area, and thus cannot be performed by a single robot. Also, robot swarms may be the preferred alternative in hazardous environments, such as military operations, chemical handling and toxic spill cleanups, search and rescue missions or fire fighting. In such situations, one may also be willing to accept the possibility of losing a fraction of the units in the swarm. Multiple robot systems may also be used for simple repetitive tasks that humans find extremely boring, tiresome of repelling.

* Supported by the Israel Science Foundation (grant No. 693/04).

A. Pal et al. (Eds.): IWDC 2005, LNCS 3741, pp. 1–12, 2005.

Autonomous mobile robot systems have been studied in a number of different disciplines in engineering and artificial intelligence. Some notable examples for directions taken include the Cellular Robotic System [23], swarm intelligence [7], the self-assembly machine [26], social interaction and intelligent behavior [25], behavior based robot systems [27, 28, 6], multi robot learning [29, 30], and ant robotics [42]. See [8] for a survey of the area.

Robot swarms typically consist of robots that are very small, very simple and very limited in their capabilities. More specifically, they have weak energy resources, limited means of communication and limited processing power. In fact, a common and recurring metaphor is that of insect swarms, and a number of algorithms and methodologies developed for robot swarms draw their inspiration from this metaphor.

While most of the research efforts invested in mobile robots to date were dedicated to engineering aspects (focusing on mobility and function), it is clear that the transition from a single robot to a swarm of robots necessitates some changes also in the approach taken towards the control and coordination mechanisms governing the behavior of the robots. In particular, dealing with the movements of robots in a swarm raises some algorithmic problems that do not exist when considering a single mobile robot. The individual robots must coordinate their movements at least partially, in order to avoid colliding or constricting each other, and to optimize the performance of the entire swarm.

Typical coordination tasks studied in the literature include the following. *Gathering* is the task where starting from any initial configuration, the robots should gather at a single point (within a finite number of steps). A closely related problem is *convergence*, requiring the robots to converge to a single point, rather than gather at it (namely, for every $\epsilon > 0$ there must be a time t_ϵ by which all robots are within distance of at most ϵ of each other). *Pattern formation* requires the robots to arrange themselves in a simple geometric form such as a circle, a simple polygon or a line segment. *Flocking* is the task of following a designated leader. Additional coordination tasks include partitioning, spreading, exploration and mapping, patrolling and searching, and avoiding collisions or bottlenecks.

Most of the experimental studies of multiple robot systems dealt with a fairly small group of robots, typically less than a dozen. A system of that size can usually be controlled centrally, relying on ad-hoc heuristic protocols. Indeed, algorithmic aspects were usually handled in such systems in an implicit manner, mostly ignoring issues such as correctness proof or complexity analysis. However, multi-robot systems envisioned for the future will consist of tens of thousands of small individual units, and such systems can no longer be controlled by a central entity in an efficient way. While hierarchical approaches may be developed, it seems that certain tasks may need to be managed in a fully decentralized manner.

Subsequently, over the last decade there have been a number of efforts to study issues related to the coordination and control of robot swarms from the point of view of distributed computing (cf. [31, 39, 40, 37, 3]), and in particular, to model an environment consisting of mobile autonomous robots and study the capabilities the robots must have in order to achieve their common goals.

This development is fascinating in that it provides the "distributed computing" community with a distributed model that is fundamentally different in some central ways from most of the traditional distributed models, including the model assumptions, the research problems one is required to solve, and the typical concerns one is faced with in trying to solve those problems.

The current paper reviews this exciting area of research and its main developments over the last decade, discusses some of the central obstacles and difficulties, and outlines two main directions for future research.

2 Review of the Literature

2.1 Common Models for Distributed Coordination Algorithms

A number of computational models for robot swarms were proposed in the literature, and several studies dealt with characterizing the influence of the chosen model on the ability of a robot swarm to perform certain basic tasks under different constraints. The general setting consists of a group of mobile robots which all execute the same algorithm in order to perform a given coordination task.

Robot operation cycle: Each robot operates individually in cycles consisting of the following three steps.

- **Look**: identify the locations of the other robots and form a map of the current configuration on your private coordinate system (the model may assume either a perfect vision or a limited visibility range),
- **Compute**: execute the given algorithm, obtaining a *goal point* p_G,
- **Move**: move towards the point p_G. (It is sometimes assumed that the robot might stop before reaching its goal point p_G, but is guaranteed to traverse at least some minimal distance, unless reaching the goal first.)

The "look" and "move" steps are carried out identically in every cycle, independently of the algorithm used; algorithms differ only in their "compute" step.

In most papers in this area (cf. [38, 39, 21, 12]), the robots are assumed to be *oblivious* (or memoryless), namely, they cannot remember their previous states, their previous actions or the previous positions of the robots. Hence the algorithm employed by the robots for the "compute" step cannot rely on information from previous cycles, and its only input is the current configuration.

The robots are also assumed to be indistinguishable, so when looking at the current configuration, each robot knows its own location but does not know the identity of the robots at each of the other points. Furthermore, the robots are assumed to have no means of directly communicating with each other.

The synchronization model: With respect to time, three main models have been considered. The first [37, 40], hereafter referred to as the *semi-synchronous* model, is partially synchronous: all robots operate according to the same clock cycles, but not all robots are necessarily active in all cycles. Robots that are awake at a given cycle may measure the positions of all other robots and then make a local computation and move instantaneously accordingly. The activation of the different robots can be thought of as managed by a hypothetical "scheduler", whose only "fairness" obligation is that each robot must be activated and given a chance to operate infinitely often in any infinite execution. The second, closely related model of [31, 32, 34], hereafter referred to as the *asynchronous* model, differs from the semi-synchronous model in that each robot acts independently in a cycle composed of *four* steps: Wait, Look, Compute, Move. The length of this cycle is finite but not bounded. Consequently, there is no bound on the length of the walk in a single cycle, and different cycles of the same robot may vary in length. The third model is the *synchronous* model [40], in which robots operate by the same clock and all robots are active on all cycles.

2.2 Known Results on Distributed Coordination Algorithms

Much of the theoretical research on distributed algorithms for mobile robots was focused on attempting to answer the question: "how restricted can the robots be and still be able to accomplish certain cooperative tasks?" In other words, the primary motivation of the studies presented, e.g., in [37, 40, 31, 32, 39] was to identify the *minimal* capabilities a collection of distributed robots must have in order to accomplish certain basic tasks and produce interesting interaction.

Various aspects of coordination in autonomous mobile robot systems have been studied in the literature. A basic task that has received considerable attention is the gathering problem. This problem was discussed in [39, 40] in the semi-synchronous model, where it was shown that gathering *two* oblivious autonomous mobile robots without common orientation is impossible. In contrast, an algorithm for gathering $N \geq 3$ robots was presented in [40]. In the fully asynchronous model, a gathering algorithm for $N = 3, 4$ robots is given in [33, 12], and for arbitrary $N \geq 5$ the problem is solved in [11]. Gathering was studied also (in both the semi-synchronous and asynchronous models) in an environment of limited visibility. Visibility conditions are modeled via a (symmetric) *visibility graph* representing the visibility relation between the robots. The problem was proven to be unsolvable when the visibility graph is not connected [21]. A convergence algorithm for any N in limited visibility environments is presented in [2]. A gathering algorithm in the asynchronous model is described in [21], under the assumption that all robots share a compass (i.e., agree on a direction in the plane). The natural gravitational algorithm based on going to the center of gravity, and its convergence properties, were studied in [15, 14] in the semi-synchronous and asynchronous models respectively. Gathering without the ability to detect multiplicity but with unlimited memory is studied in [10], and gathering without both capabilities is shown to be impossible in the asynchronous model in [35].

Formation of geometric patterns was studied in [3, 37, 39, 40, 16, 19, 9, 22]. The algorithms presented therein enable a group of robots to self-arrange and spread itself nearly evenly along the form shaped. The task of flocking, requiring the robots to follow a predefined leader, was studied in [33].

Searching a (static or moving) target in a specified region by a group of robots in a distributed fashion is a natural application for mobile robot systems. Two important related tasks, studied in [37], are even distribution, namely, requiring the robots to spread out uniformly over a specified region, and partitioning, where the robots must split themselves into a number of groups. Finally, the wake-up task requires a single initially awake robot to wake up all the others. A variant of this problem is the Freeze-Tag problem studied in [5, 41].

3 Future Directions

3.1 Modifications in the Robot Model

The existing body of literature on distributed algorithms for autonomous mobile robot systems represents a significant theoretical base containing a rich collection of tools and techniques. The main goals of initial research in this area were to obtain basic understanding and develop a pool of common techniques and methodologies, but equally importantly, to explore and chart the border between the attainable and the unattainable under the most extreme model, representing the weakest possible type of robots in the harshest possible external environment.

Consequently, the models adopted in these studies assume the robots to be very weak and simple. In particular, these robots are generally (although not always) assumed to be oblivious. They are also assumed to have no common coordinate system, orientation, scale or compass, and no means of explicit communication (not even of a limited type, such as receiving broadcasts from a global beacon). It is also assumed that these robots are anonymous, namely, have no identifying characteristics. Also, the robots are usually taken to be dimensionless, namely, treated as points. This implies that robots do not obstruct each other's visibility or movement, i.e., two robots whose timed trajectories intersect will simply pass "through" each other. (This is not necessarily a "weak" property, but it is an unrealistic assumption nontheless.)

These assumptions lead to challenging "distributed coordination" problems since the only means of communication is through using "positional" or "geometric" information, yielding a novel variant of the classical distributed model (which is based on direct communication). The resulting questions are interesting from a theoretical point of view, as they allow us to explore the theoretical limits of robot swarms. Moreover, it is often advantageous to develop algorithms for the weakest robot types possible, as an algorithm that works correctly for weak robots will clearly work correctly in a system of stronger robot types.

On the other hand, the extremely weak model often leads to cumbersome, artificial and sometimes impractical algorithmic solutions. Moreover, towards the practical application of such algorithmic techniques, it is necessary to develop a methodology supporting modularity and allowing multi-phase processes. This becomes difficult if the robots are assumed to be completely memoryless. In fact, it seems that tasks even slightly more involved than the basic ones studied in the literature might pose insurmountable barriers under such weak assumptions. Consider a two-stage project requiring the robots to gather and then perform some follow-up task. The feasibility of such a project is unclear: as the robots are deaf, mute, and forgetful, it seems doubtful that they can accomplish much once they do meet each other. Furthermore, even if they do try to embark on the follow-up task after gathering, their obliviousness will repeatedly force them to immediately resume their attempts to gather.

It is thus clear that the focus on extremely weak robots limits the practicality of many of the distributed algorithms presented in the literature for autonomous mobile robot systems, despite their importance as a base of algorithmic ideas, paradigms and techniques for multi-robot coordination. Subsequently, future research in this area should focus on modifying the model in order to allow a more accurate representation, taking into account the fact that actual robots are usually not so helpless. It is expected that a rigorous algorithmic theory based on accurate assumptions and realistic models may lead to simpler and more practical algorithms which can be readily used within experimental and real systems.

Understandably, it does not make sense to expect the emergence of a single unifying model covering the entire spectrum of possible applications. Nevertheless, let us outline some of the main characteristics a realistic model should have, with a number of possible variations in certain aspects.

A central modification in the model that has to be examined involves the effects of equipping each robot with a small amount (say, $O(1)$ bits) of stable memory. The most immediate benefit is that this will allow the (possibly significant) simplification of most existing algorithms for robot coordination. The reason for this is that many of the complications present in those algorithms were necessary to overcome this lack of memory, and once robots can save state, those complications can be dispensed with. The effect of this change should be systematically investigated across all coordination tasks studied in the literature. A second advantage of introducing memory is that allowing the robots some stable memory may facilitate the modular composition of a number of sub-procedures into a single algorithm, since this stable memory may allow the robots to recognize the computational phase they're in at any given moment. It may be interesting to consider also partial changes along this line, such as allowing the robot to maintain partial history (say, remember the last k cycles).

A second modification concerns the assumption that the robots in a swarm lack common orientation. In many natural settings, the robots may enjoy at least a partial agreement on their orientation. For instance, they may agree on the North, or use a common unit of distance or a common point of reference. It could be interesting to examine the effects of such partial orientation agreements on the solvability and computational complexity of simple coordination tasks. Our initial studies in this direction indicate that with respect to the gathering problem, each of these assumptions may suffice to improve the situation, either by making the problem solvable in settings where impossibility holds otherwise, or by facilitating a simpler solution.

Another interesting question concerns examining which problems can be solved more efficiently or in a simpler manner when the robots are allowed a partial means of explicit communication. This relaxation is also expected to cause a dramatic change in the efficient solvability of various coordination problems. Since the robots are expected to operate in difficult environments and on rugged terrains, it makes sense to focus on restricted communication forms. For example, in certain scenarios a robot may be allowed to communicate only with robots within a limited range (say, radius r from its location), or only with robots to which its line of sight is unobstructed.

Even in settings where explicit communication is infeasible or prohibitively expensive, it may be possible (and desirable) to incorporate in the model some simple means of identification and signalling, such as marking (at least some of) the robots with colors, flags or visible indicator lights. Such modifications may be simple to implement and yet may positively affect the ease of solving some coordination problems, hence this direction deserves thorough examination.

Another assumption that may need to be discarded is that robots are dimensionless, and can pass each other without colliding. A more realistic assumption is that two (or more) robots moving towards each other will stop once meeting (say, by colliding) or shortly before (say, through some "soft halt" mechanism allowing robots to detect a near-collision and halt).

3.2 Introducing Fault Tolerance

While the classical model is rather restrictive on the one hand, it is perhaps somewhat "too optimistic" on the other, in that it assumes perfectly functioning robots. As future robot swarms are expected to comprise of cheap, simple and relatively weak robots and operate under harsh conditions, the issue of resilience to failure becomes crucial, since in such systems one cannot possibly rely on assuming fail-proof hardware or software.

When considering the issue of coping with faults, we may classify the problems that need to be dealt with into two types: problems that occur regularly during the normal operation of every robot as a result of its inherent imperfections, and problems resulting from the *malfunction* of some robots. Next we discuss these two fault types and possible ways to overcome them.

Overcoming Robot Imperfections. The common robot model makes the assumption that the configuration map obtained by a robot observing its surroundings is perfect. In fact, certain algorithmic solutions proposed in the literature rely critically on this assumption. In practice, however, the robot measurements suffer from nonnegligible inaccuracies in both distance and angle estimations. (For instance, the accuracy of range estimation in sonar sensors is about $\pm 1\%$ and the angular separation is about 3°, cf. [36].) The same applies to the precision of robot movements, as a variety of mechanical factors, including unstable power supply, friction and force control, make it hard to control the exact distance a robot traverses in a single cycle, or to predict it with high accuracy.

Another unrealistic assumption is that robots are capable of carrying out infinite precision calculations over the reals. For instance, this assumption underlies the distinction between the gathering and convergence problems. In fact, it is sometimes assumed that the robots have unlimited computational power. The fact that in reality robots cannot perform perfect precision calculations may seem insignificant, since floating point arithmetic can be carried to very high accuracy with modern computers. However, this may prove to be a serious problem. For instance, the point that minimizes the sum of distances to the robots' locations (also known as the Weber point) may be used to achieve gathering. However, this point is not computable, due to its infinite sensitivity to location errors. More generally, the correctness of many of the distributed coordination algorithms presented in the literature is proven by relying on basic properties from Euclidean geometry. Unfortunately, these properties are often no longer valid when measurement or calculation errors occur. To illustrate this point, consider Algorithm `3-Gather` presented in [1], which gathers three robots using several simple rules. One of these rules states that if the robots form an obtuse triangle, then they move towards the vertex with the obtuse angle. Thus, as shown in [13], this algorithm might fail to achieve even convergence in the presence of angle measurement errors of at least $15°$. Similar problems arise with other algorithms described in the literature.

Subsequently, for the "next-generation" model of robot swarms, it is desirable to discard these unrealistic assumptions and examine whether efficient algorithmic solutions can still be obtained for coordination problems of interest.

An initial study [13] examines a model in which the robot's location estimation and movements are imprecise, with imprecision bounded by some accuracy parameter ϵ known at the robot's design stage. The measurement imprecisions can affect both distance and angle estimations. Formally, the robot's distance estimation is *ϵ-precise* if, whenever the real distance to an observed point in the robot's private coordinate system is D, the measurement d taken by the robot for that distance satisfies $(1 - \epsilon)d < D < (1 + \epsilon)d$. A similar imprecision is allowed for angle estimations.

Several impossibility results are established in [13], limiting the maximum inaccuracy that still allows convergence. Specifically, it is shown that gathering is impossible for any number of robots assuming inaccuracies in both distance and angle measurements, even in a fully synchronous model and when the robots have unlimited memory and are allowed to use randomness. (If angle measurements are always exact, then impossibility of gathering is known only for $N = 2$ robots, and is conjectured for any N.) Hence at best, only the weaker requirement of convergence can be expected. Actually, it seems reasonable to conjecture that even convergence is impossible for robots with large measurement errors. The exact limits are not completely clear. Some rather weak limits on the possibility of convergence are given in [13], where it is shown that for a configuration of $N = 3$ robots having an error of $\pi/3$ or more in angle measurement, there is no deterministic algorithm for convergence even assuming exact distance estimation, fully synchronous model and unlimited memory. On the other hand, an algorithm is presented in [13] for convergence under bounded imprecision (specifically, $\epsilon < 0.2$ or so) in the synchronous and semi-synchronous models.

Some natural questions to be explored further include the following. First, the precision required of the robots for the algorithm of [13] to work correctly is still significant, and improved techniques are necessary for overcoming this. Second, it would be interesting to obtain similar results in the asynchronous setting. Third, similar techniques should be developed for other coordination tasks, such as pattern formation, search, etc.

It may also be interesting to examine distributed coordination algorithms with an eye towards complexity, trying to develop variants that are both simple and resource efficient in terms of internal computation costs at each robot. One specific aspect of this is discarding the assumption of infinite precision in real computations, and settling for approximations. This may necessitate some relaxations in the definitions of certain common tasks (such as gathering at a single point or forming perfect geometric objects) to fit these weaker assumptions.

Overcoming Robot Malfunctions. Robot swarms are intended to operate in tough and hazardous environments, so it is to be expected that certain robots may malfunction. Indeed, one of the main attractive features of robot swarms is their potential for enhanced fault tolerance through inherent redundancy. For example, a fault tolerant algorithm for gathering should be required to ensure that even if some fraction of the robots fails in any execution, all the *nonfaulty* robots still manage to gather at a single point within a finite time, regardless of the actions taken by the faulty ones.

Perhaps surprisingly, however, this aspect of multiple robot systems has been explored to very little extent so far. In fact, almost all the results reported in the literature rely on the assumption that all robots function properly and follow their protocol without any deviation.

One exception concerns *transient* failures. As observed in [40, 37, 20], any algorithm that works correctly on oblivious robots is necessarily *self-stabilizing*, i.e., it guarantees that after any transient failure the system will return to a correct state and the goal will be achieved. Another fault model studied in [37] considers restricted sensor and control failures, and assumes that whenever failures occur in the system, the identities of the faulty robots become known to all robots. Unfortunately, this assumption might not hold in many typical settings, and in case unidentified faults do occur in the system, it is no longer guaranteed that the algorithms of [37, 40] remain correct.

Following traditional approaches in the field of distributed computing, it is interesting to study robot algorithms under the *crash* and *Byzantine* fault models. In order to pinpoint the effect of faults, all other aspects of the model can be left unchanged, following the basic models of [37, 31]. In the Byzantine fault model it is assumed that a faulty robot might behave in arbitrary and unforeseeable ways. It is sometimes convenient to model the behavior of the system by means of an *adversary* which has the ability to control the behavior of the faulty robots, as well as the "undetermined" features in the behavior of the nonfaulty processors (e.g., the distance to which they move). In the crash fault model, it is assumed that the only faulty behavior allowed for a faulty robot is to crash, i.e., stop functioning. This may happen at any point in time during the cycle, including any time during the movement towards the goal point.

In [43], an algorithm is given for the *Active Robot Selection Problem (ARSP)* in the presence of *initial* crash faults. The ARSP creates a subgroup of nonfaulty robots from a set that includes also initially crashed robots and enables the robots in that subgroup to recognize one another.

A systematic study of the gathering problem in failure-prone robot systems is presented in [1]. Under the crash fault model, it is shown in [1] that the gathering problem with at most one crash failure is solvable in the semi-synchronous model. Considering the Byzantine fault model, it is shown that it is impossible to perform a successful gathering in the semi-synchronous or asynchronous model even in the presence of a single fault. For the synchronous model, an algorithm is presented for solving the gathering problem in N-robot systems whenever the maximum number of faults f satisfies $3f + 1 \leq N$.

In general, the design of fault-tolerant distributed control algorithms for multiple robot systems is still a largely unexplored direction left for future study. Particularly, a number of questions are left open in [1]. In the synchronous model, while the algorithm of [1] does solve the problem even with Byzantine faults, its complexity is prohibitively high, rendering it impractical except maybe for very small systems. Hence it is desirable to look for a simpler and faster algorithm. In the asynchronous and semi-synchronous models, the techniques of [1] are inadequate for handling more than a single fault, again limiting their applicability rather drastically, and it is interesting to investigate approaches for extending these techniques to multiple failures. More generally, as the asynchronous model captures a more faithful representation of typical actual settings, we view the derivation of suitable algorithms for performing various coordination tasks in this model in the presence of multiple crash faults as one of the central directions of research in this area. Turning to Byzantine faults in the asynchronous and semi-synchronous models, as such faults make gathering impossible, a plausible alternative is to try to solve the slightly weaker problem of convergence.

Moreover, as the initial study of [1] was limited to the gathering problem, it would be interesting to investigate also the fault-tolerance properties of currently available algorithms for other tasks described above (e.g., formation of geometric patterns). Specifically, a central theme of both theoretical and practical significance concerns identifying the maximum number of faults under which a solution for a particular coordination problem is still feasible. It would be attractive to develop a general theory answering this question, similar to the theory developed for the analogous question in classical distributed systems.

References

1. N. Agmon and D. Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. In *Proc. 15th ACM-SIAM Symp. on Discrete Algo.*, 1063–1071, 2004.
2. H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita. A distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Trans. Robotics and Automation*, 15:818–828, 1999.

3. H. Ando, I. Suzuki, and M. Yamashita. Formation and agreement problems for synchronous mobile robots with limited visibility. In *Proc. IEEE Symp. of Intelligent Control*, 453–460, 1995.
4. R.C. Arkin. *Behavior-Based Robotics.* MIT Press, 1998.
5. E. Arkin, M. Bender, S. Fekete, J. Mitchell, and M. Skutella. The freeze-tag problem: How to wake up a swarm of robots. In *Proc. 13th ACM-SIAM Symp. on Discrete Algorithms*, 2002.
6. T. Balch and R. Arkin. Behavior-based formation control for multi-robot teams. *IEEE Trans. on Robotics and Automation*, 14, 1998.
7. G. Beni and S. Hackwood. Coherent swarm motion under distributed control. In *Proc. DARS'92*, 39–52, 1992.
8. Y.U. Cao, A.S. Fukunaga, and A.B. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4(1):7–23, 1997.
9. I. Chatzigiannakis, M. Markou, and S.E. Nikoletseas. Distributed circle formation for anonymous oblivious robots. In *Proc. 3rd Workshop on Experimental and Efficient Algorithms*, LNCS 3059, 159–174, 2004.
10. M. Cieliebak. On the feasibility of gathering by autonomous mobile robots. In *Proc. 6th Latin American Symp. on Theoret. Inform.*, LNCS 2976, 577–588, 2004.
11. M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro. Solving the robots gathering problem. In *Proc. 30th ICALP*, 1181–1196, 2003.
12. M. Cieliebak and G. Prencipe. Gathering autonomous mobile robots. In *Proc. 9th Int. Colloq. on Struct. Info. and Commun. Complex.*, 57–72, 2002.
13. R. Cohen and D. Peleg. Convergence of autonomous mobile robots with inaccurate sensors and movements. Tech. Rep. MSC 04-8, Weizmann Inst. of Science, 2004.
14. R. Cohen and D. Peleg. Convergence properties of the gravitational algorithm in asynchronous robot systems. In *Proc. 12th ESA*, LNCS 3221, 228–239, 2004.
15. R. Cohen and D. Peleg. Robot convergence via center-of-gravity algorithms. In *Proc. 11th Colloq. on Struct. Inf. and Comm. Complex.*, LNCS 3104, 79–88, 2004.
16. X. Defago and A. Konagaya. Circle formation for oblivious anonymous mobile robots with no common sense of orientation. In *Proc. 2nd ACM Workshop on Principles of Mobile Computing*, 97–104. ACM Press, 2002.
17. M. Erdmann and T. Lozano-Pdrez. On multiple moving objects. *Algorithmica*, 2:477–521, 1987.
18. M. Erdmann and T. Lozano-Perez. On multiple moving objects. In *Proc. IEEE Conf. on Robotics and Automation*, 1419–1424, 1986.
19. P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Hard tasks for weak robots: The role of common knowledge in pattern formation by autonomous mobile robots. In *Proc. 10th Int. Symp. on Algorithms and Computation*, 93–102, 1999.
20. P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Distributed coordination of a set of autonomous mobile robots. In *Proc. IEEE Intelligent Vehicles Symp.*, 480–485, 2000.
21. P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Gathering of autonomous mobile robots with limited visibility. In *Proc. 18th STACS*, 247–258, 2001.
22. B. Katreniak. Biangular circle formation by asynchronous mobile robots. In *Proc. 12th Colloq. on Struct. Info. and Commun. Complex.*, LNCS 3499, 185–199, 2005.
23. Y. Kawauchi, M. Inaba, and T. Fukuda. A principle of decision making of cellular robotic system (CEBOT). In *Proc. IEEE Conf. on Robotics and Automation*, 833–838, 1993.
24. Y. Kuniyoshi, S. Rougeaux, M. Ishii, N. Kita, S. Sakane, and M. Kakikura. Cooperation by observation - the framework and basic task patterns. In *Proc. Int. Conf. on Robotics and Automation*, 767–774, 1994.

25. M.J. Mataric. *Interaction and Intelligent Behavior.* PhD thesis, MIT, 1994.
26. S. Murata, H. Kurokawa, and S. Kokaji. Self-assembling machine. In *Proc. IEEE Conf. on Robotics and Automation*, 441–448, 1994.
27. L.E. Parker. Designing control laws for cooperative agent teams. In *Proc. IEEE Conf. on Robotics and Automation*, 582–587, 1993.
28. L.E. Parker. On the design of behavior-based multi-robot teams. *J. of Advanced Robotics*, 10, 1996.
29. L.E. Parker and C. Touzet. Multi-robot learning in a cooperative observation task. In *Distributed Autonomous Robotic Systems 4*, 391–401, 2000.
30. L.E. Parker, C. Touzet, and F. Fernandez. Techniques for learning in multi-robot teams. In T. Balch and L.E. Parker, editors, *Robot Teams: From Diversity to Polymorphism.* A. K. Peters, 2001.
31. G. Prencipe. CORDA: Distributed coordination of a set of autonomous mobile robots. In *Proc. 4th Eur. Res. Seminar on Adv. in Distr. Syst.*, 185–190, 2001.
32. G. Prencipe. Instantaneous actions vs. full asynchronicity: Controlling and coordinating a set of autonomous mobile robots. In *Proc. 7th Italian Conf. on Theoretical Computer Science*, 185–190, 2001.
33. G. Prencipe. *Distributed Coordination of a Set of Autonomous Mobile Robots.* PhD thesis, Universita Degli Studi Di Pisa, 2002.
34. G. Prencipe. The effect of synchronisity on the behavior of autonomous mobile robots. *Theory of Computing Systems*, 2004.
35. G. Prencipe. On the feasibility of gathering by autonomous mobile robots. In *Proc. 12th Colloq. on Struct. Info. and Commun. Complex.*, LNCS 3499, 246–261, 2005.
36. SensComp Inc. *Spec. of 6500 series ranging modules.* http://www.senscomp.com.
37. K. Sugihara and I. Suzuki. Distributed algorithms for formation of geometric patterns with many mobile robots. *J. of Robotic Systems*, 13(3):127–139, 1996.
38. I. Suzuki and M. Yamashita. Agreement on a common x-y coordinate system by a group of mobile robots. In *Proc. Dagstuhl Seminar on Modeling and Planning for Sensor-Based Intelligent Robots*, 1996.
39. I. Suzuki and M. Yamashita. Distributed anonymous mobile robots - formation and agreement problems. In *Proc. 3rd Colloq. on Struct. Info. and Commun. Complex.*, 313–330, 1996.
40. I. Suzuki and M. Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM J. on Computing*, 28:1347–1363, 1999.
41. M. Sztainberg, E. Arkin, M. Bender, and J. Mitchell. Analysis of heuristics for the freeze-tag problem. In *Proc. 8th Scand. Workshop on Alg. Theory*, 270–279, 2002.
42. I.A. Wagner and A.M. Bruckstein. From ants to a(ge)nts. *Annals of Mathematics and Artificial Intelligence*, 31, special issue on ant-robotics:1–5, 1996.
43. D. Yoshida, T. Masuzawa, and H. Fujiwara. Fault-tolerant distributed algorithms for autonomous mobile robots with crash faults. *Systems and Computers in Japan*, 28:33–43, 1997.