Proceedings of the

Fourth European Research Seminar on Advances in Distributed Systems

ERSADS 2001

Ozalp Babaoglu (Ed.)

14–18 May 2001

Bertinoro (Forlì), Italy

Corda: Distributed Coordination of a Set of Autonomous Mobile Robots

Giuseppe Prencipe
Dipartimento di Informatica, Università di Pisa
Corso Italia, 40, 56100 Pisa - Italy
Tel. +39 050 887273, Fax +39 050 887226
E-mail: prencipe@di.unipi.it

Abstract

The distributed coordination and control of a set of autonomous mobile robots is a problem which has been extensively studied in several fields: engineering, artificial intelligence, artificial life, robotics. In contrast with the common empirical approach used in these areas, a new line of investigations has recently being proposed, which approaches the problem from a *distributed computing* point of view. In this paper, we describe the current status of these investigations.

1 Introduction

The problem of controlling mobile robots that can freely move on a plane, with no central control, is a problem widely studied in several fields, such as engineering and artificial intelligence. This problem gained even more popularity in the last decade, when the interest shifted from studying the control of single powerful units, to the design of set of simple units that were asked to coordinate themselves in order to achieve a given task. Several reasons can be addressed to motivate this shift: the advantages that can arise from a distributed and parallel solution to the given problems, such as a faster computation; the ability to perform tasks which are unable to be executed by a single agent [7, 16]; for fault tolerance considerations; the decreased cost through simpler individual robot design. In a system consisting of a set of totally distributed agents the goal is generally to exploit the multiplicity of the elements in the system so that the execution of a certain number of predetermined tasks occurs in a coordinated and distributed way.

Generally, the problem of distributively coordinate a set of mobile units has been approached mostly from an empirical point of view. Among the studies conducted in the engineering area, we can cite the Cellular Robotic System (CEBOT) of Kawaguchi et al [13], the Swarm Intelligence of Beni et al. [3], the Self-Assembly Machine ("fructum") of Murata et al. [15], etc. A number of remarkable studies has been done also in the AI community, eg., on social interaction leading to group behavior by Matarić [14], on selfish behavior of cooperative robots in animal societies by Parker [16], on primitive animal behavior in pattern formation by Balch and Arkin [2], to cite just a few (see [4] for a survey).

In all these investigations, however, algorithmic aspects were somehow implicitly an issue, but clearly not a major concern. An investigation with an algorithmic flavor has been undertaken within the AI community by Durfee [8], 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.

In our study, we approach the problem from a different perspective: from a *distributed computing* point of view. In other words, we aim to understand the relationship between the capabilities of the robots and the solvability of the tasks they are given. In particular, we analyze the impact of the *knowledge* of the environment: can the robots form an arbitrary geometric pattern if they share 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 work of I. Suzuki *et. al.* [1, 19] is the closest to our study (and, with this focus, a rarity in the mobile robots literature). It approaches the algorithmic issues related to the pattern formation for robots, under several assumptions on the power of the individual robot. Our model, however, differs with respect to the assumptions on the robots capabilities (our approach is more general, and, in our opinion, better models the way a set of asynchronous mobile robots interacts in reality).

Since this investigation deals with the distributed control of a set of entities that have to coordinate themselves in order to achieve a common goal, this study is also interesting for other distributed systems. In fact, we study which problems these entities can achieve and under which conditions; we design distributed algorithms that the entities concurrently and asynchronously execute, and that let them to reach their common goal in finite time. In contrast with the traditional approach, however, our entities can freely move on a plane, and not strictly on a graph.

In Section 2 the formal definition of the model under study is presented. In Section 3 we present some results related to problems that are generally analyzed when studying the control and coordination of mobile robots. Finally, in Section 4 we draw some conclusions and present suggestions for further study.

2 The CORDA Model

The robots we consider are [9, 11]: *homogeneous* (they all follow the same set of rules), *autonomous* (there is no a priori central authority, and each robot's computing capabilities are independent from the others), *asynchronous* (there is no central clock, no a priori synchronization, no a priori bounds on processing or motorial speed), *mobile* (the robots are allowed to move on a plane), *anonymous* (they are a priori indistinguishable), *oblivious* (they do not explicitly remember the past). Moreover, there are no explicit direct means of communication: the communication occurs in a totally implicit manner, through the environment.

These assumptions make our robots simple and rather "weak" in light of current engineering technology. But, as already noted, we are interested in approaching the problem from a *computational* point of view; it is precisely by assuming the "weakest" robots, that it is possible to analyze the strengths and weaknesses of the distributed control. .

Each robot has its own *local view* of the world. This view includes a local Cartesian coordinate system with origin, unit of length, and the *directions* of two coordinate axes, identified as x axis and y axis, together with their *orientations*, identified as the positive and negative sides of the axes. Notice, however, that the robots do not necessarily have the same *handedness (chirality)* of the coordinate system (e.g., two robots can have their local y axis oriented in different directions), making impossible for the robots to agree on directions or on distances; we say that the robots have, in general, different *local views of the world*.

The robots observe the environment, compute a destination point based on their deterministic algorithm (all the robots share the same algorithm), and move; this is their only means of communication and of expressing a decision that they have taken. More formally, each robot, at any point in time, is in one of the following four states: *Wait*, *Look*, *Compute*, and *Move*. A robot is initially in *Wait*; a robot cannot stay infinitely idle. At any point in time, asynchronously and independently from the other robots, the robot observes the world by activating its sensors which will return a snapshot of the positions of all other robots with respect to its local coordinate system: each robot is viewed as a point, hence the result of the *Look* is just the set of their coordinates. Two different models can arise depending on whether we assume that a robot can see all the other robots in the system (called *Unlimited Visibility* model) or that a robot can see only the robots that are at most at some fixed distance from it (*Limited Visibility* model). After having observed, the robot performs a *local computation* according to its deterministic algorithm. The result of the computation is a destination point; if this point is the current location, the robot stays still (*null movement*). Otherwise, the robot moves towards the computed destination; this operation can terminate before the robot has reached it[1]. Moreover, the movement can not be infinite, nor infinitesimally small (see Assumption A2 below). Finally, the robot goes back to the *Wait* state. The sequence *Wait - Look - Compute - Move* will be called a *computation cycle* (or briefly *cycle*) of a robot.

In the model, there are only two limiting assumptions about time and space. Namely, **A1** The amount of time required by a robot r to complete a computational cycle is neither infinite nor infinitesimally small; and **A2** The distance traveled by a robot r in a move is neither infinite nor infinitesimally small: in particular, there exists an arbitrarily small constant $\delta_r > 0$, such that if the destination point is closer than δ_r, r will reach it; otherwise, r will move towards it of at least δ_r. As no other assumption on time exists, the resulting system is *fully asynchronous* and the duration of each activity (or inactivity) is unpredictable. As a result, the robots do not have a common notion of time, robots can be seen while moving, computations can be made based on obsolete observations.

In all our algorithms, the robots do not need to remember data from previous observations; that is, the algorithm they execute takes as input only the robots' positions observed in the last *Look* (hence, both the result of the computation and that of the *Look* will not be available to the robot at its next computational cycle). We say that the algorithm is *oblivious* (similarly, we say that the robots are *oblivious*).

The simplicity of the robots we model, allows us to formally highlight by an algorithmic and computational viewpoint the minimal capabilities the agents must have in order to accomplish basic tasks and produce interesting interactions. Furthermore, it allows us to understand better the limitations of the distributed control in an environment inhabited by mobile agents. The main motivations that prompted us to study the problem in this new perspective can be found in [10].

3 Main Results

As already mentioned, the work of I. Suzuki *et al.* is the closest to ours. There are, however, some aspects that render our approach and theirs quite different. In particular, in [1, 19] *instantaneous action* of the robots is modeled: every robots execute their cycle *atomically*. One consequence of this approach, is that it is not possible to model different motorial and computational speed of the robots. Moreover, a robot can not be seen while it is moving by robots that are observing. This departure in the way the asynchronicity is modeled, leads to the fact that all the algorithmic solution proposed in [1, 19] do not work, in general, in CORDA [18]. Moreover, let us denote by $\mathfrak{C}$ and $\mathfrak{Z}$ the class of problem that are solvable in CORDA and in the model presented in [1, 19], respectively. We have

Theorem 3.1 **([18]).** $\mathfrak{C} \subset \mathfrak{Z}$.

Hence, we studied in CORDA the problems that are commonly analyzed in robotics when dealing with the coordination and control of a set of mobile robots. The results are reported in the following sections.

3.1 Arbitrary Pattern Formation

In the *Arbitrary Pattern Formation* problem, the robots are given in input the same pattern, described as a set of points (given by their Cartesian coordinates) in the plane (clearly, each robots "sees" this pattern according to the direction and orientation of its local coordinate system). They are required to *form the pattern*: at the end of the computation, the positions of the robots coincide, in everybody's local view, with the points of the pattern, where the input pattern may be *translated*, *rotated*, *scaled*, and *flipped* into its mirror position in each local coordinate system. Initially, the robots are in arbitrary positions, with the only requirement that no two robots be

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

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 *pattern formation* problem has been extensively investigated in the literature [6, 19, 20], where generally the first step is to gather the robots together and then let them proceed in the desired formation. 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.

Also I. Suzuki and M. Yamashita solve the same problem in their model [19], characterizing what kind of patterns can be formed. But all their algorithms are *non-oblivious*; in fact, they require the capability of the robots to remember the past, while ours are totally oblivious.

In an attempt to understand the power of *common knowledge* for the coordination of robots, we have studied the pattern formation problem under several assumptions, obtaining a complete characterization of what can and what cannot be achieved. The following theorem summarizes the results holding for a set of n autonomous, anonymous, oblivious, mobile robots:

Theorem 3.2.

1. *With common knowledge of (i.e., the robots agree on) both x and y directions and orientations, the robots can form an arbitrary given pattern [9].*
2. *With common knowledge on only one axis direction and orientation, the pattern formation problem is unsolvable when n is even, while it can be solved if n is odd [9].*
3. *With common knowledge on only one axis direction and orientation, an even number of robots can form only symmetric patterns that have at least one axis of symmetry not passing through any vertex of the pattern [17].*
4. *With no common knowledge, the robots cannot form an arbitrary given pattern [9].*

3.2 Gathering

In the gathering problem, the robots, initially placed in arbitrary positions, are required to gather in a not predetermined point. In the unlimited visibility setting, one feature the robots must have in order to solve this problem, is the ability to detect *multiplicity*, that is the ability to detect if on a given point on the plane there is more than one robot [18] (recall that the robots are viewed as points). In this setting, this problem presents several difficulties. One of the problems arises from the difficulty to handle initial configurations where the n robots are placed on the n vertices of a regular polygon. Since in general the robots can all have the same local view of the world, no robot can deterministically be chosen to break the symmetry. If such a situation is forbidden at the beginning, a deterministic algorithm can be provided [5]. The complete characterization is still under study.

We solved, however, the gathering problem in the limited visibility setting, where the robots can sense only the portion of the plane at distance V from them. In particular, our algorithm requires that the robots agree on the direction and orientation of both the x and y axis (e.g., the robots have a *compass*). A necessary condition to solve this problem, is that there is no robot completely "isolated" from the others at the beginning of the computation. More formally, let C_i be the visibility area of a robot r_i. We define the *visibility graph* as follows:

Definition 3.1 (Visibility Graph). The *visibility graph* $G = (N, E)$ is a graph whose node set N is the set of the input robots and $(r_i, r_j) \in E$ iff $r_j \in C_i$ and $r_i \in C_j$, where r_i and r_j are two robots in their initial positions.

We can state the following:

Lemma 3.1 ([11]). *If the visibility graph is disconnected, the problem is unsolvable.*

Let $Left$ and $Right$ be the leftmost and rightmost vertical axis, respectively, where some robot initially lie. The idea of the algorithm is to make the robots move towards $Right$, in such a way that, after a finite number of steps, they will reach it and gather at the bottom most position occupied by a robot at that time.

Theorem 3.3 ([11]). *There exists a deterministic oblivious algorithm that let the robots gather in one point in a finite number of movements, in the limited visibility setting and assuming common knowledge on direction and orientation of both axes.*

This same problem has been investigated also in [1], where the authors have presented a procedure that let the robots *converge* to, but not *reach*, the point.

3.3 Flocking

In this problem, there are two kinds of robots in the environment: the *leader L*, and the *followers*. The leader acts independently from the others, and we can assume that it is driven by an human pilot. The followers are required to follow the leader wherever it goes, while keeping a formation they are given in input (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).

We analyzed the problem, assuming that the leaders have no agreement on the orientation and direction of the x and y axis. Let us denote with v_L and v_f respectively the maximum linear velocity of the leader and of the generic follower f, and with ω_L the maximum angular velocity of L. The following theorem states the condition under which the followers can successfully follow the leader.

Theorem 3.4 ([12]). *In order for the formation to be kept in movement, it is necessary for the leader to have* $\omega_L \cdot r + v_L < \min_f v_f$.

We presented an oblivious algorithm that allows the robots to keep formations that are symmetric with respect to the direction of movement of L [12]. We tested our algorithm with computer simulation. All our 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 ship along her route. Moreover, the obliviousness of the algorithm contributes to this result, since the followers do not base their computation on past leader's positions.

4 Conclusions and Discussion

The purpose of our study is to gain a better understanding of the power of the distributed control of a set of mobile robots from an algorithmic point of view. We described a model, CORDA, consisting of a set of autonomous, anonymous, memoryless, mobile robots - features that render our robots "weak" - and we have outlined the current status of the investigation.

There are many issues which merit further research. First, the open problems. The result of Theorem 3.1 has been proven in the non-oblivious setting. In the oblivious setting, we only proved in [18] that the algorithmic solutions of [1, 19] do not work in CORDA; hence, an open problem is to formally understand the relationship between CORDA and the model in [1, 19] in the oblivious case. Another open problem regards the design of an oblivious algorithm that can handle the totally symmetric initial configurations in the gathering problem.

Then, it would be interesting to look at models where robots have different features. For instance, we could use a *non-oblivious* model, that is, one with an unlimited amount of memory that each robot could use. Alternatively, we could equip the robots with just a bounded amount of memory (*semi-obliviousness*), and see if this added "power" can be useful in solving problems otherwise unsolvable, or if it could be used to design faster algorithm. We could add a dimension to the robots, and stationary obstacles to the environment, thus adding the possibility of collision between robots or between moving robots and obstacles. We could also explore robots that have some kind of direct communication, and we could assume different kind of robots that move in the environment (like in the flocking problem).

Relationship between memory and ability of the robots to complete given tasks, dimensional robots, obstacles in the environment, suggest that the algorithmic nature of distributed coordination of autonomous, mobile robots merits further investigation.

References

[1] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita. A Distributed Memoryless Point Convergence Algorithm for Mobile Robots with Limited Visibility. *IEEE Trans. on Robotics and Automation*, 15(5):818–828, 1999.

[2] T. Balch and R. C. Arkin. Behavior-based Formation Control for Multi-robot Teams. *IEEE Trans. on Robotics and Automation*, 14(6), 1998.

[3] G. Beni and S. Hackwood. Coherent Swarm Motion Under Distributed Control. In *Proc. DARS'92*, pages 39–52, 1992.

[4] Y. U. Cao, A. S. Fukunaga, A. B. Kahng, and F. Meng. Cooperative Mobile Robotics: Antecedents and Directions. In *Int. Conf. on Intel. Robots and Sys.*, pages 226–234, 1995.

[5] M. Celiebak and G. Prencipe. Point Formation for Oblivious Robots. manuscript, 2001.

[6] Q. Chen and J. Y. S. Luh. Coordination and Control of a Group of Small Mobile Robots. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2315–2320, 1994.

[7] B. R. Donald, J. Jennings, and D. Rus. Information Invariants for Distributed Manipulation. *The Int. Journal of Robotics Research*, 16(5), 1997.

[8] E. H. Durfee. Blissful Ignorance: Knowing Just Enough to Coordinate Well. In *ICMAS*, pages 406–413, 1995.

[9] 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 *ISAAC '99*, pages 93–102, 1999.

[10] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Distributed Coordination of a Set of Autonomous Mobile Robots. In *IVS*, pages 480–485, 2000.

[11] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Gathering of Asynchronous Mobile Robots with Limited Visibility. In *STACS*, 2001.

[12] V. Gervasi and G. Prencipe. Need a Fleet? Use The Force. In *Fun with Algorithms 2*, 2001.

[13] Y. Kawauchi and M. Inaba and. T. Fukuda. A Principle of Decision Making of Cellular Robotic System (CEBOT). In *Proc. IEEE Conf. on Rob. and Autom.*, pages 833–838, 1993.

[14] M. J Matarić. *Interaction and Intelligent Behavior*. PhD thesis, MIT, 1994.

[15] S. Murata, H. Kurokawa, and S. Kokaji. Self-Assembling Machine. In *Proc. IEEE Conf. on Robotics and Autom.*, pages 441–448, 1994.

[16] L. E. Parker. On the Design of Behavior-Based Multi-Robot Teams. *Journal of Advanced Robotics*, 10(6), 1996.

[17] G. Prencipe. Achievable Patterns by an Even Number of Mobile Robots. Technical Report TR-00-11, Univ. di Pisa, 2000.

[18] G. Prencipe. *Istantaneous Action* vs. *Full Asynchronicity*: Controlling and Coordinating a Set of Autonomous Mobile Robots. submitted, 2001.

[19] I. Suzuki and M. Yamashita. Distributed Anonymous Mobile Robots: Formation of Geometric Patterns. *Siam J. Comput.*, 28(4):1347–1363, 1999.

[20] P. K. C. Wang. Navigation Strategies for Multiple Autonomous Mobile Robots Moving in Formation. *Journal of Robotic Systems*, 8(2):177–195, 1991.