Exploration d'arbres par des équipes de robots asynchrones et sans mémoire

Paola Flocchini[1 †], David Ilcinkas[2‡], Andrzej Pelc[2†‡] et Nicola Santoro[3†]

[1] *SITE, University of Ottawa, Ottawa, ON K1N 6N5, Canada. E-mail:* flocchin@site.uottawa.ca
[2] *Département d'informatique, Université du Québec en Outaouais, Gatineau, Québec J8X 3X7, Canada. E-mail:* ilcinkas@lri.fr, pelc@uqo.ca.
[3]*School of Computer Science, Carleton University, Ottawa, Ontario, K1S 5B6, Canada. E-mail:* santoro@scs.carleton.ca

---

Une équipe d'entités mobiles (robots), identiques et sans mémoire, doit explorer un arbre anonyme en visitant tous ses nœuds. Les robots démarrent depuis des nœuds différents et arbitraires de l'arbre et opèrent suivant des cycles Regarder-Calculer-Aller (Look-Compute-Move). A la fin, chaque nœud doit avoir été visité par au moins un robot, et tous les robots doivent s'arrêter. Dans un cycle, un robot prend une photographie de la configuration courante (Regarder), prend la décision de rester immobile ou de se déplacer vers un des nœuds adjacents (Calculer), et dans ce dernier cas se déplace instantanément vers ce voisin (Aller). Les cycles sont exécutés de façon asynchrone par chaque robot. Nous présentons un algorithme d'exploration pour tous les arbres à n nœuds de degré maximum 3 utilisant $O(\log n/\log\log n)$ robots, et nous prouvons que pour de tels arbres $\Omega(\log n/\log\log n)$ robots sont nécessaires pour les explorer. Nous montrons par ailleurs que l'exploration de certains arbres à n nœuds de degré maximum 4 nécessite $\Omega(n)$ robots.

Keywords: mobile agent, robot, oblivious, asynchronous, tree, exploration

---

1 Introduction

1.1 The problem and the model

We study the problem of exploration of trees by a team of mobile entities, called robots. These entities, initially located at arbitrary different nodes of the tree, have to explore it, by collectively visiting all nodes. At the end, every node must be visited by at least one robot and all robots must stop. We study the exploration problem in a very weak model that makes coordination of robots' actions particularly hard, as robots cannot communicate directly but have to make decisions about their moves only by observing the environment. Moreover, they operate asynchronously and do not have memory of past observations.

Consider an unoriented anonymous tree. Neither nodes nor links of the tree have any labels. Initially, some nodes of the tree are occupied by robots and there is at most one robot in each node. Robots operate in Look-Compute-Move cycles. In one cycle, a robot takes a snapshot of the current configuration (Look), then, based on the perceived configuration, makes a decision to stay idle or to move to one of its adjacent nodes (Compute), and in the latter case makes an instantaneous move to this neighbor (Move). Cycles are performed asynchronously for each robot. This means that the time between Look, Compute, and Move operations is finite but unbounded, and is decided by the adversary for each action of each robot. The only constraint is that moves are instantaneous, and hence any robot performing a Look operation sees all other robots at nodes of the tree and not on edges, while performing a move. However a robot $\mathcal{R}$ may perform a Look operation at some time t, perceiving robots at some nodes, then Compute a target neighbor at some time $t' > t$, and Move to this neighbor at some later time $t'' > t'$ in which some robots

---

†Partially supported by NSERC discovery grant.
‡This work was done during the stay of David Ilcinkas at the Research Chair in Distributed Computing of the Université du Québec en Outaouais and at the University of Ottawa, as a postdoctoral fellow. Andrzej Pelc was partially supported by the Research Chair in Distributed Computing at the Université du Québec en Outaouais.

are in different nodes from those previously perceived by $\mathcal{R}$ because in the meantime they performed their Move operations. Hence robots may move based on significantly outdated perceptions, which adds to the difficulty of exploration. It should be stressed that robots are oblivious, i.e., they do not have any memory of past observations. Thus the target node (which is either the current position of the robot or one of its neighbors) is decided by the robot during a Compute operation solely on the basis of the location of other robots perceived in the previous Look operation. Robots are anonymous and execute the same deterministic algorithm. They cannot leave any marks at visited nodes, nor send any messages to other robots.

This very weak scenario, introduced in [KMP06] and similar to that considered in [AOSY99, FPSW05, Pre05], is justified by the fact that robots may be very small, cheap and mass-produced devices. Adding distinct labels, memory, or communication capabilities would make such robots larger and more expensive, which is not desirable. Thus it is interesting to consider such a scenario from the point of view of applications. On the theoretical side, this weak model increases the difficulty of collective exploration by making the problem of coordinating actions of robots particularly hard, and thus provides an interesting insight to the general problem of organizing collective actions of mobile entities in a distributed environment.

An important and well studied capability that may be available to oblivious robots is the *multiplicity detection* [FPSW05, KMP06, Pre05]. This is the ability of the robots to perceive, during the Look operation, if there is one or more robots in a given location. In our case, it is easy to see that without this capability, exploration is always impossible. Indeed, there must be a configuration in which robots stop. If the initial configuration occupied precisely these nodes, only with one robot per node, there would be no way to perceive any difference from the stopping configuration (since the total number of robots is not known to them), and thus robots would stop immediately, without exploring any new node. Thus we assume the capability of multiplicity detection in our further considerations. It should be stressed that, during a Look operation, a robot can only tell if at some node there are no robots, there is one robot, or there are more than one robots: a robot does not see a difference between a node occupied by a or b robots, for distinct $a,b > 1$.

One final precision has to be added, concerning the decisions of robots made during the Compute action. Every such decision is based on the snapshot obtained during the last Look action. However it may happen that two or more ports at a node v currently occupied by the deciding robot look identical in this snapshot, i.e., there is an automorphism of the tree which fixes v, carries empty nodes to empty nodes, occupied nodes to occupied nodes, and multiplicities to multiplicities, and carries one port to the other. In this case if the robot decides to take one of these ports, it may take any of the identically looking ports. We assume the worst-case decision in such cases, i.e., that the actual port among the identically looking ones is chosen by an adversary. This is a natural worst-case assumption and it is important in some impossibility arguments: in some cases the adversary may prevent exploration by directing a robot to an already explored part of the tree, instead of the yet unexplored part.

1.2 Related Work

Algorithms for graph exploration by mobile agents (robots) have been recently studied by many authors. Most of the research is concerned with the case of a single robot exploring the graph. In the case of anonymous graphs it is impossible to explore arbitrary graphs by a single robot, if no marking of nodes is allowed. Hence the scenario adopted in [DJMW91] allows the use of a *pebble* which the robot can drop on nodes to recognize already visited ones, and then remove it and drop on other nodes. In trees however, pebbles are not necessary but the robot needs unbounded memory to perform exploration [DFKP04].

Exploration by many robots has been investigated mostly as a graph optimization problem, in the context when moves of the robots are centrally coordinated. In [FHK78], approximation algorithms are given for the collective exploration problem in arbitrary graphs. In [AB97] the authors construct approximation algorithms for the collective exploration problem in weighted trees. On the other hand, in [FGKP04] the authors study the problem of distributed collective exploration of trees of unknown topology. However, the robots performing exploration have memory of past actions and can directly communicate with each other.

The very weak assumption of asynchronous identical robots that cannot send any messages and communicate with the environment only by observing it, has been first used to study the problem of gathering robots in one location. Most of the research in this area concerned the case of robots moving freely in the plane [AOSY99, Cie04, FPSW05, Pre05, SY99]. Our scenario has been recently introduced in [KMP06] to study the gathering problem in the ring. This scenario is very similar to the asynchronous model used in [FPSW05, Pre05]. The only difference with respect to [FPSW05, Pre05] is in the execution of Move operations. This has been adapted to the context of graphs: moves of the robots are executed instantaneously from a node to its neighbor, and hence robots always see other robots at nodes. All possibilities of the adversary concerning interleaving operations performed by various robots are the same as in the model from [FPSW05, Pre05], and the characteristics of the robots (anonymity, obliviousness, multiplicity detection) are also the same. Very recently (cf. [FIPS07]) we used the model introduced in [KMP06] to study the exploration problem in the ring. It should be noted that in this very weak model, the exploration problem is significantly more difficult than gathering. This is due to the fact that in gathering the accomplishment of the task is readily seen in a snapshot: all robots are in one node. By contrast, in order to complete exploration, robots have to "remember" which nodes were visited. Since they do not have any memory of past events, this recollection must be coded in the dynamically changing configurations, and the design of this coding is one of the main challenges of exploration by oblivious robots.

1.3 Our results

We present an exploration algorithm, using $O(\log n/\log\log n)$ robots, for arbitrary n-node trees of maximum degree 3, and we prove that for some such trees, namely for complete binary trees, $\Omega(\log n/\log\log n)$ robots are necessary to explore them. Hence we show that the minimum number of robots sufficient to explore all n-node trees of maximum degree 3 is $\Theta(\log n/\log\log n)$. None of the two assumptions used in our positive result can be removed. The assumption about maximum degree 3 is crucial because one can show that in order to explore some n-node trees of maximum degree 4, $\Omega(n)$ robots are necessary. On the other hand, the assumption that the explored graph is a tree cannot be removed either. Indeed, we showed in [FIPS07] that $\Omega(\log n)$ robots are necessary to explore some n-node rings. One can finally show that the difficulty in tree exploration comes in fact from the symmetries of the tree. Indeed in order to explore trees that do not have any non-trivial automorphisms, 4 robots are always sufficient and often necessary.

2 Exploration of trees of maximum degree 3

2.1 Algorithm Tree-exploration

We start with the following upper bound on the size of the team of robots capable to explore all n-node trees of maximum degree 3.

Theorem 1 *There exists a team of $O(\log n/\log\log n)$ robots that can explore all n-node trees of maximum degree 3, starting from any initial configuration.*

This result is proved by showing an exploration algorithm using $O(\log n/\log\log n)$ robots.

The main idea of the algorithm is the following. The entire tree is partitioned into two or three subtrees, the number of parts depending on the shape of the tree. Parts are explored one after another by a team of three robots that sequentially visit leaves of this part. Since individual robots do not have memory, a specially constructed, dynamic configuration of robots, called the "brain", keeps track of what has been done so far. More precisely, the brain counts the number of already visited leaves and indicates the next leaf to be visited. It is also the brain that requires most of the robots used in the exploration process. The reason why $\Theta(\log n/\log\log n)$ robots are sufficient for exploration, is that the counting process is efficiently organized. The counting module of the brain consists of disjoint paths of logarithmic lengths, which are appropriately marked by groups of robots of bounded size (towers). Paths are of logarithmic lengths because this is the maximum length guaranteed in any tree of maximum degree 3. Inside each of these paths a tower moves, indicating a numerical value by its position in the path. The combination of these values yields the current value of the number of visited leaves. Since the number of leaves may be $\Theta(n)$, we need a number x of paths, which can produce $\Theta(n)$ combinations of values, i.e., such that $(\Theta(\log n))^x = \Theta(n)$. This is the reason of constructing $\Theta(\log n/\log\log n)$ paths and thus using $\Theta(\log n/\log\log n)$ robots. We show how to construct these paths in any tree of maximum degree 3, and how to organize the counting process. The latter is complicated by the asynchronous behavior of the robots. During the switch of the counter from value i to

$i+1$ robots move in the paths and a snapshot taken during the transition period shows a "blurred" picture: the old value is already destroyed while the new one is not yet created. This could confuse the robots and disorganize the process. Thus we use two counters acting together. They both indicate value i, then one of them keeps this value and the other transits to $i+1$. When this is completed, the first counter transits to $i+1$ and so on. This precaution permits to keep track of the current value during the process of incrementation. During the exploration of one part of the tree, the brain is located in another part and controls exploration remotely. After completing the exploration of one part, the brain is moved to the already explored part in order to let the exploring agents visit the rest of the tree.

There are two main difficulties in our algorithm. The first is to break symmetries that can exist in configurations of robots, in order to let them act independently and reach appropriate target nodes, in particular during the construction of the brain. To ensure that at least one robot has a different view from the others in the initial configuration, the number k of robots is chosen such that $k \equiv 5 \pmod 6$. The second challenge is the construction and relocation of the brain, as well as organizing its proper functioning by coordinating the two counters, regardless of the behavior of the adversary that controls asynchrony.

2.2 The lower bound on the number of robots

We now give a lower bound on the number of robots necessary for exploration of complete binary trees, that matches the upper bound given by Algorithm Tree-exploration.

Theorem 2 $\Omega(\frac{\log n}{\log\log n})$ *robots are required to explore n-node complete binary trees.*

References

[AOSY99] H. Ando, Y. Oasa, I. Suzuki, M. Yamashita, Distributed memoryless point convergence algorithm for mobile robots with limited visibility, IEEE Trans. on Rob. and Aut. 15 (1999), 818-828.

[AB97] I. Averbakh and O. Berman, $(p-1)/(p+1)$-approximate algorithms for p-traveling salesmen problems on a tree with minmax objective, Discr. Appl. Mathematics 75 (1997), 201-216.

[Cie04] M. Cieliebak, Gathering non-oblivious mobile robots, Proc. 6th Latin American Symposium on Theoretical Informatics (LATIN 2004), 577-588.

[DFKP04] K. Diks, P. Fraigniaud, E. Kranakis and A. Pelc., Tree Exploration with Little Memory. Journal of Algorithms 51(1):38-63, 2004.

[DJMW91] G. Dudek, M. R. M. Jenkin, E. E. Milios, and D. Wilkes. Robotic exploration as graph construction. *IEEE Transactions on Robotics and Automation*, 7(6):859–865, December 1991.

[FGKP04] P. Fraigniaud, L. Gasieniec, D. Kowalski, A. Pelc, Collective tree exploration, Proc. Latin American Theoretical Informatics (LATIN 2004), LNCS 2976, 141-151.

[FHK78] G. N. Frederickson, M. S. Hecht and C. E. Kim, Approximation algorithms for some routing problems. SIAM J. on Computing 7 (1978), 178-193.

[FIPS07] P. Flocchini, D. Ilcinkas, A. Pelc, N. Santoro, Ring and line exploration by a team of asynchronous oblivious robots, Technical Report TR-2007-01, University of Ottawa, SITE, 2007.

[FPSW05] P. Flocchini, G. Prencipe, N. Santoro, P. Widmayer, Gathering of asynchronous robots with limited visibility. Theoretical Computer Science 337 (2005) 147-168.

[KMP06] R. Klasing, E. Markou, A. Pelc, Gathering asynchronous oblivious mobile robots in a ring, Proc. 17th International Symposium on Algorithms and Computation (ISAAC 2006), LNCS 4288, 744-753.

[Pre05] G. Prencipe, On the feasibility of gathering by autonomous mobile robots. Proc. 12th Int. Col. on Structural Information and Communication Complexity (SIROCCO 2005), LNCS 3499, 246-261.

[SY99] I. Suzuki, M. Yamashita, Distributed anonymous mobile robots: formation of geometric patterns. SIAM J. Comput. 28 (1999) 1347-1363.