J. Parallel Distrib. Comput. 119 (2018) 92–106

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Uniform deployment of mobile agents in asynchronous rings[☆,☆☆]

Masahiro Shibata [a,*], Toshiya Mega [b], Fukuhito Ooshita [c], Hirotsugu Kakugawa [b], Toshimitsu Masuzawa [b]

[a] Kyushu Institute of Technology, 680-4, Kawadu, Iizuka, Fukuoka, 820-8502, Japan
[b] Graduate School of Information Science and Technology, Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
[c] Graduate School of Information Science, NAIST, Takayama 8916-5, Ikoma, Nara 630-0192, Japan

ARTICLE INFO

Article history:
Received 23 May 2017
Received in revised form 21 February 2018
Accepted 28 March 2018
Available online 20 April 2018

Keywords:
Distributed system
Mobile agent
Uniform deployment
Ring networks
Token
Symmetry degree

ABSTRACT

In this paper, we consider the uniform deployment problem of mobile agents in asynchronous unidirectional rings, which requires the agents to uniformly spread in the ring. The uniform deployment problem is in striking contrast to the rendezvous problem which requires the agents to meet at the same node. While rendezvous aims to break the symmetry, uniform deployment aims to attain the symmetry. It is well known that the symmetry breaking is difficult in distributed systems and the rendezvous problem cannot be solved from some initial configurations. Hence, we are interested in clarifying what difference the uniform deployment problem has on the solvability and the number of agent moves compared to the rendezvous problem. We consider two problem settings, with knowledge of k (or n) and without knowledge of k or n where k is the number of agents and n is the number of nodes. First, we consider agents with knowledge of k (or n since k and n can be easily obtained if one of them is given). In this case, we propose two algorithms. The first algorithm solves the uniform deployment problem with termination detection. This algorithm requires $O(k \log n)$ memory space per agent, $O(n)$ time, and $O(kn)$ total moves. The second algorithm also solves the uniform deployment problem with termination detection. This algorithm reduces the memory space per agent to $O(\log n)$, but uses $O(n \log k)$ time, and requires $O(kn)$ total moves. Both algorithms are asymptotically optimal in terms of total moves since there are some initial configurations such that agents require $\Omega(kn)$ total moves to solve the problem. Next, we consider agents with no knowledge of k or n. In this case, we show that, when termination detection is required, there exists no algorithm to solve the uniform deployment problem. For this reason, we consider the relaxed uniform deployment problem that does not require termination detection, and we propose an algorithm to solve the relaxed uniform deployment problem. This algorithm requires $O((k/l) \log(n/l))$ memory space per agent, $O(n/l)$ time, and $O(kn/l)$ total moves when the initial configuration has symmetry degree l. This means that the algorithm can solve the problem more efficiently when the initial configuration has higher symmetric degree (i.e., is closer to uniform deployment). Note that all the proposed algorithms achieve uniform deployment from any initial configuration, which is a striking difference from the rendezvous problem because the rendezvous problem is not solvable from some initial configurations.

1. Introduction

1.1. Background and motivation

A *distributed system* consists of a set of computers (*nodes*) connected by communication links. As a promising design paradigm of distributed systems, (mobile) agent systems have attracted a lot of attention [11,3]. Agents can traverse the system carrying information collected at visiting nodes and process tasks on each node using the information. In other words, agents can encapsulate the process code and data, which simplifies design of distributed systems [15,4].

☆ The conference version of this paper is published in the proceedings of 29th ACM Symposium on Principles of Distributed Computing (PODC 2016).

☆☆ This work was supported by JSPS KAKENHI Grant Numbers 24500039, 26280022, 26330084, 15H00816, and 16K00018.

* Corresponding author.

E-mail addresses: shibata@cse.kyutech.ac.jp (M. Shibata), f-oosita@is.naist.jp (F. Ooshita), kakugawa@ist.osaka-u.ac.jp (H. Kakugawa), masuzawa@ist.osaka-u.ac.jp (T. Masuzawa).

https://doi.org/10.1016/j.jpdc.2018.03.008
0743-7315/

In this paper, we consider the *uniform deployment* (or *uniform scattering*) *problem* as a fundamental problem for coordination of agents. This problem requires all agents to spread uniformly in the network. From a practical point of view, uniform deployment is useful for the network management. In a distributed system, it is necessary that regularly each node gets software updates and is checked whether some application installed on the node is running correctly or not [17,12]. Hence, considering agents with such services, uniform deployment guarantees that agents visit each node at short intervals and provide services. Uniform deployment might be useful also for a kind of the load balancing. That is, considering agents with large-size database replicas, uniform deployment guarantees that not all nodes need to store the database but each node can quickly access the database [16,19]. Hence, we can see the uniform deployment problem as a kind of the resource allocation problem.

1.2. Related works

There are several researches considering the uniform deployment problem in the *Look–Compute–Move* model. In this model, agents are assumed to be oblivious (or memoryless) but be able to observe multiple agents (and nodes in graph environments) within its visibility range. In the look phase, an agent takes a snapshot and gets the positions of all agents (and nodes in graph environments) within the visibility range. In the compute phase, based on the snapshot, the agent decides where to go in the next movement. In the move phase, the agent moves to the destination. Agents repeat such cycles until the given task is completed. In the Look–Compute–Move model, Flocchini et al. [9] considered the uniform deployment problem in cycle environments of length m (m is a real number). They considered two types of uniform deployment: *exact* and *ϵ-approximate*. In the exact uniform deployment, agents move in the ring so that the distance between any two consecutive agents is the same, say d. In the ϵ-approximate uniform deployment, agents move in the cycle so that the distance should be between $d - \epsilon$ and $d + \epsilon$. They showed that if agents do not have common sense of direction, agents cannot solve the exact uniform deployment problem even if agents have unlimited memory and visibility range. If agents have common sense of direction, they proposed an algorithm to solve the exact uniform deployment problem for agents with knowledge of d. In addition, for any $\epsilon > 0$ they proposed an algorithm to solve the ϵ-approximate uniform deployment problem for agents without knowledge of d.

Elor et al. [7] considered uniform deployment in ring networks. They considered agents without knowledge k or n, where k is the number of agents and n is the number of nodes, but with visibility range VR. They considered a semi-synchronous model, that is, a subset of agents execute a behavior in each round. They showed that, if $VR < \lfloor n/k \rfloor$ holds, agents cannot solve the uniform deployment problem. If $VR \geq \lfloor n/k \rfloor$ holds, they proposed an algorithm to solve the balanced uniform deployment problem without quiescence. That is, agents eventually satisfy the condition of uniform deployment and continue to move in the ring satisfying the condition. In addition, they proposed an algorithm to solve the semi-balanced uniform deployment problem with quiescence. That is, agents eventually terminate the algorithm satisfying the condition such that the distance between any two adjacent agents is between $n/k - k/2$ and $n/k + k/2$.

While [9] and [7] considered uniform deployment in ring networks, Barriere et al. [2] considered uniform deployment in grid networks and proposed an algorithm to achieve uniform deployment in $O(n/d)$ time, where d is the interval of uniform deployment.

1.3. Our contributions

In this paper, we focus on uniform deployment on asynchronous unidirectional rings. Although ring networks might seem so restricted in practice, it is known that the idea for ring networks is fundamental one and can be applied to other networks by embedding a ring in the network [6,20]. Different from [9,7,2], we consider agents that have memory but cannot observe nodes except for the currently located node. To the best of our knowledge, this is the first research considering uniform deployment for such agents. In addition to the fact that uniform deployment is useful from a practical point of view as mentioned before, it is interesting to investigate also from a theoretical point of view. The problem exhibits a striking contrast to the *rendezvous problem*. The rendezvous problem, one of the most investigated problem, requires all agents to meet at a single node [14], and by doing this agents can share information or synchronize behaviors among them [8,10,13,1,5]. While rendezvous aims to break the symmetry and requires all the agents to meet at the single node, uniform deployment aims to attain the symmetry of agent locations and requires agents to spread uniformly. It is well known that the symmetry breaking is difficult (and sometimes impossible) in distributed systems, and the rendezvous problem cannot be solved from some initial configurations. Hence, it is interesting to clarify what difference the uniform deployment problem has on the solvability and the number of agent moves compared to the rendezvous problem.

Contributions of this paper are summarized in Table 1. We assume that each agent initially has a token and can release it on a node that it is visiting. After a token is released at some node, agents cannot remove the token. In addition, we assume that agents can send a message of any size to agents staying at the same node. We consider two problem settings. First, we consider agents with knowledge of k (or n since k and n can be easily obtained if one of them is given). In this case, we propose two algorithms. The first algorithm solves the uniform deployment problem with termination detection. This algorithm requires $O(k \log n)$ memory space per agent, $O(n)$ time, and $O(kn)$ total moves. The second algorithm also solves the uniform deployment problem with termination detection. This algorithm reduces the memory space per agent to $O(\log n)$, but uses $O(n \log k)$ time, and requires $O(kn)$ total moves. Note that, from some initial configurations agents require $\Omega(kn)$ total moves to solve the problem. Hence, both algorithms are asymptotically optimal in terms of total moves.

Next, we consider agents with no knowledge of k or n. In this case, we show that, when termination detection is required, there exists no algorithm to solve the uniform deployment problem. Intuitively, it is due to impossibility of finding k or n when the initial configuration has sufficient number of repetitions of an agent location pattern: when an agent misestimates these at smaller numbers than actual ones, it prematurely terminates and uniform deployment cannot be achieved.

For this reason, we consider the relaxed uniform deployment problem that does not require termination detection, and we propose an algorithm to solve the relaxed uniform deployment problem. In this algorithm, each agent estimates k and n (possibly at smaller values than actual ones) and behaves based on the estimation. Thus, the efficiency of the algorithm depends on the estimation. To evaluate the efficiency, we introduce the following parameter l to denote by the symmetry degree of an initial configuration: we say that an initial configuration has symmetry degree l when its distance sequence can be represented as l-times repetition of some aperiodic sequence. For example, the initial configuration in Fig. 1(a) has symmetry degree 1 since its whole distance sequence (1,4,2,1,2,2) is aperiodic, and the initial configuration in Fig. 1(b) has symmetry degree 2 since its whole distance sequence (1,2,3,1,2,3) is represented as 2-times repetition of aperiodic sequence (1,2,3). Hence, the symmetry degree