J. Parallel Distrib. Comput. 83 (2015) 159–167

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

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

Tradeoffs between cost and information for rendezvous and treasure hunt☆

Avery Miller*, Andrzej Pelc

Université du Québec en Outaouais, Canada

HIGHLIGHTS

- In rendezvous, two agents traverse edges in rounds and have to meet at some node.
- In treasure hunt, an agent must find a fixed target at some node of the network.
- Objective: tradeoffs between the advice available to the agents and the cost.
- Results: bounds on the size of advice to achieve a given cost.

ARTICLE INFO

Article history:
Received 8 January 2015
Received in revised form
25 May 2015
Accepted 13 June 2015
Available online 18 June 2015

Keywords:
Rendezvous
Treasure hunt
Advice
Deterministic algorithms
Mobile agents
Cost

ABSTRACT

In rendezvous, two agents traverse network edges in synchronous rounds and have to meet at some node. In treasure hunt, a single agent has to find a stationary target situated at an unknown node of the network. We study tradeoffs between the amount of information (*advice*) available *a priori* to the agents and the cost (number of edge traversals) of rendezvous and treasure hunt. Our goal is to find the smallest size of advice which enables the agents to solve these tasks at some cost C in a network with e edges. This size turns out to depend on the initial distance D and on the ratio $\frac{e}{C}$, which is the *relative cost gain* due to advice. For arbitrary graphs, we give upper and lower bounds of $O(D\log(D\cdot\frac{e}{C})+\log\log e)$ and $\Omega(D\log\frac{e}{C})$, respectively, on the optimal size of advice. For the class of trees, we give nearly tight upper and lower bounds of $O(D\log\frac{e}{C}+\log\log e)$ and $\Omega(D\log\frac{e}{C})$, respectively.

1. Introduction

1.1. Model and problems

Rendezvous and treasure hunt are two basic tasks performed by mobile agents in networks. In rendezvous, two agents, initially located at distinct nodes of the network, traverse network edges in synchronous rounds and have to meet at some node. In treasure hunt, a single agent has to find a stationary target (called treasure) situated at an unknown node of the network. The network might model a labyrinth or a system of corridors in a cave, in which case the agents might be mobile robots. The meeting of such robots might be motivated by the need to exchange previously collected samples, or to agree how to share a future cleaning or decontamination task. Treasure hunt might mean searching a cave for a resource or for a missing person after an accident. In other applications we can consider a computer network, in which the mobile entities are software agents. The meeting of such agents might be necessary to exchange data or share a future task of checking the functionality of network components. Treasure hunt in this case might mean looking for valuable data residing at some node of the network, or for a virus implanted at some site.

The network is modeled as a simple undirected connected graph whose nodes have distinct identities. Ports at a node of degree d are numbered $0,\ldots,d-1$. The agents are anonymous, i.e., do not have identifiers. Agents execute a deterministic algorithm, such that, at each step, they choose a port at the current node. When an agent enters a node, it learns the entry port number, the label of the node and its degree. The cost of a rendezvous algorithm is the total worst-case number of edge traversals performed by both agents until meeting. The cost of a treasure hunt algorithm is the worst-case number of edge traversals performed by the agent

☆ A preliminary version of this paper appeared in the Proceedings of the 18th International Conference on Principles of Distributed Systems (OPODIS 2014).

* Correspondence to:812-265 Poulin Ave., Ottawa, Ontario, Canada, K2B 7Y8.
E-mail addresses: avery@averymiller.ca (A. Miller), andrzej.pelc@uqo.ca (A. Pelc).

http://dx.doi.org/10.1016/j.jpdc.2015.06.004
0743-7315/

until the treasure is found. If the agents have no information about the network, the cost of both rendezvous and treasure hunt can be as large as $\Theta(e)$ for networks with e edges. This is clear for treasure hunt, as all edges (except one) need to be traversed by the agent to find the treasure in the worst case. The same lower bound for rendezvous follows from Proposition 2.1 in the present paper. On the other hand, if D is the distance between the initial positions of the agents, or from the initial position of the agent to the treasure, a lower bound on the cost of rendezvous and of treasure hunt is D.

In this paper, we study tradeoffs between the amount of information available *a priori* to the agents and the cost of rendezvous and treasure hunt. Following the paradigm of algorithms with advice [1,14,16,21,26,28–34,36–38,44,49], this information is provided to the agents at the start of their navigation by an oracle that knows the network, the starting positions of the agents and, in the case of treasure hunt, the node where the treasure is hidden. The oracle assists the agents by providing them with a binary string called *advice*, which can be used by the agent during the algorithm execution. In the case of rendezvous, the advice given to each agent can be different. The length of the string given to the agent in treasure hunt and the sum of the lengths of strings given to both agents in rendezvous are called the *size of advice*.

1.2. Our results

Using the framework of advice permits us to quantify the amount of information needed for an efficient solution of a given network problem (in our case, rendezvous and treasure hunt) regardless of the type of information that is provided. Our goal is to find the smallest size of advice which enables the agents to solve rendezvous and treasure hunt at a given cost C in a network with e edges. This size turns out to depend on the initial distance D (between the agents in rendezvous, and between the agent and the treasure in treasure hunt) and on the ratio $\frac{e}{C}$, which is the *relative cost gain* due to advice. For arbitrary graphs, we give upper and lower bounds of $O(D\log(D\cdot\frac{e}{C})+\log\log e)$ and $\Omega(D\log\frac{e}{C})$, respectively, on the optimal size of advice. Hence our bounds leave only a logarithmic gap in the general case. For the class of trees, we give nearly tight upper and lower bounds of $O(D\log\frac{e}{C}+\log\log e)$ and $\Omega(D\log\frac{e}{C})$, respectively. Our upper bounds are obtained by constructing an algorithm for all graphs (respectively, for all trees) that works at the given cost and with advice of the given size, while the lower bounds are proved by exhibiting networks for which it is impossible to achieve the given cost with smaller advice.

1.3. Related work

Treasure hunt, network exploration and rendezvous in networks are interrelated problems that have received much attention in recent literature. Treasure hunt has been investigated in the line [13,35], in the plane [9] and in other terrains [41]. Treasure hunt in anonymous networks (without any information about the network) has been studied in [48,50] with the goal of minimizing cost.

The related problem of graph exploration by mobile agents (often called robots) has been intensely studied as well. The goal of this task is to visit all of the nodes and/or traverse all of the edges of a graph. A lot of research considered the case of a single agent exploring a labeled graph. In [2,20] the agent explores strongly-connected directed graphs. In a directed graph, an agent can move only in the direction from tail to head of an edge, not vice-versa. In particular, [20] investigated the minimum time of exploration of directed graphs, and [2] gave improved algorithms for this problem in terms of the deficiency of the graph (i.e., the minimum number of edges that must be added to make the graph Eulerian). Many papers, e.g., [22,24,45] studied the scenario where the graph to be explored is labeled and undirected, and the agent can traverse edges in both directions. In [45], it was shown that a graph with n nodes and e edges can be explored in time $e+O(n)$. In some papers, additional restrictions on the moves of the agent were imposed, e.g., it was assumed that the agent is tethered, i.e., attached to the base by a rope or cable of restricted length [24]. In [47], a log-space construction of a deterministic exploration for all graphs with a given bound on size was shown.

The problem of rendezvous has been studied both under randomized and deterministic scenarios. In the framework of networks, it is usually assumed that the nodes do not have distinct identities. An extensive survey of randomized rendezvous in various models can be found in [5], cf. also [3,4,8,11]. Deterministic rendezvous in networks has been surveyed in [46]. Several authors considered geometric scenarios (rendezvous in an interval of the real line, e.g., [11,12], or in the plane, e.g., [6,7]). Gathering more than two agents was studied, e.g., in [27].

For the deterministic setting, many authors studied the feasibility and time complexity of rendezvous of synchronous agents, i.e., agents that move in rounds. In [42] the authors studied tradeoffs between the time of rendezvous and the number of edge traversals by both agents. In [22], the authors presented a rendezvous algorithm whose running time is polynomial in the size of the graph, the length of the shorter label and the delay between the starting times of the agents. In [39,48], rendezvous time is polynomial in the first two of these parameters and independent of the delay. The amount of memory required by the agents to achieve deterministic rendezvous was studied in [17] for general graphs. The amount of memory needed for randomized rendezvous in the ring was discussed, e.g., in [40]. Several authors investigated asynchronous rendezvous in the plane [15,27] and in network environments [10,18,19,23].

Providing nodes or agents with information of arbitrary type that can be used to perform network tasks more efficiently has been proposed in [1,14,16,21,26,28–34,36–38,43,44,49]. This approach was referred to as algorithms with *advice*. The advice is given either to nodes of the network or to mobile agents performing some network task. Several of the authors cited above studied the minimum size of advice required to solve the respective network problem in an efficient way.

In [38], given a distributed representation of a solution for a problem, the authors investigated the number of bits of communication needed to verify the legality of the represented solution. In [29], the authors compared the minimum size of advice required to solve two information dissemination problems using a linear number of messages. In [31], it was shown that a constant amount of advice enables the nodes to carry out the distributed construction of a minimum spanning tree in logarithmic time. In [26], the advice paradigm was used for online problems. In [28], the authors established lower bounds on the size of advice needed to beat time $\Theta(\log^* n)$ for 3-coloring a cycle and to achieve time $\Theta(\log^* n)$ for 3-coloring unoriented trees. In the case of [44], the issue was not efficiency but feasibility: it was shown that $\Theta(n\log n)$ is the minimum size of advice required to perform monotone connected graph clearing. In [36], the authors studied radio networks for which it is possible to perform centralized broadcasting with advice in constant time. They proved that $O(n)$ bits of advice allow to obtain constant time in such networks, while $o(n)$ bits are not enough. In [33], the authors studied the problem of topology recognition with advice given to nodes. In [21], the authors considered the task of drawing an isomorphic map by an agent in a graph, and their goal was to determine the minimum amount of advice that has to be given to the agent for the task to be feasible.

Among the papers using the paradigm of advice, [16,30,43] are closest to the present work. Both [16,30] concerned the task of graph exploration by an agent. In [16], the authors investigated the