Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Fast rendezvous on a cycle by agents with different speeds

Ofer Feinerman [a], Amos Korman [b], Shay Kutten [c], Yoav Rodeh [d]

[a] Weizmann Institute of Science, Rehovot, Israel
[b] CNRS and University Paris Diderot, Paris, France
[c] The Technion, Haifa, Israel
[d] Jerusalem College of Engineering, Jerusalem, Israel

ARTICLE INFO

Article history:
Received 1 April 2014
Received in revised form 28 September 2015
Accepted 24 December 2015
Available online xxxx

Keywords:
Rendezvous
Asynchrony
Heterogeneity
Speed
Cycle
Pebble
White board
Mobile agents

ABSTRACT

The difference between the speed of the actions of different processes is typically considered as an obstacle that makes the achievement of cooperative goals more difficult. In this work, we aim to highlight potential *benefits* of such asynchrony phenomena to tasks involving symmetry breaking. Specifically, in this paper, identical (except for their speeds) mobile agents are placed at arbitrary locations on a (continuous) cycle of length n and use their speed difference in order to rendezvous fast. We normalize the speed of the slower agent to be 1, and fix the speed of the faster agent to be some $c > 1$. (An agent does not know whether it is the slower agent or the faster one.) The straightforward *distributed-race (*DR*)* algorithm is the one in which both agents simply start walking until rendezvous is achieved. It is easy to show that, in the worst case, the rendezvous time of DR is $n/(c-1)$. Note that in the interesting case, where c is very close to 1 (e.g., $c = 1 + 1/n^k$), this bound becomes huge. Our first result is a lower bound showing that, up to a multiplicative factor of 2, this bound is unavoidable, even in a model that allows agents to leave arbitrary marks (the *white board* model), even assuming sense of direction, and even assuming n and c are known to agents. That is, we show that under such assumptions, the rendezvous time of any algorithm is at least $\frac{n}{2(c-1)}$ if $c \leq 3$ and slightly larger (specifically, $\frac{n}{c+1}$) if $c > 3$. We then manage to construct an algorithm that precisely matches the lower bound for the case $c \leq 2$, and almost matches it when $c > 2$. Moreover, our algorithm performs under weaker assumptions than those stated above, as it does not assume sense of direction, and it allows agents to leave only a single mark (a pebble) and only at the place where they start the execution. Finally, we investigate the setting in which no marks can be used at all, and show tight bounds for $c \leq 2$, and almost tight bounds for $c > 2$.

1. Introduction

1.1. Background and motivation

The difference between the speed of the actions of different entities is typically considered disruptive in real computing systems. In this paper, we illustrate some advantages of such phenomena in cases where the difference remains *fixed* throughout the execution.[1] We demonstrate the *usefulness* of this manifestation of asynchrony to tasks involving symmetry

E-mail address: kutten@ie.technion.ac.il (S. Kutten).

[1] Advantages can also be exploited in cases where the difference in speed follows some stochastic distribution, however, in this initial study, we focus on the simpler fully deterministic case. That is, we assume a speed heterogeneity that is arbitrary but fixed throughout the execution.

http://dx.doi.org/10.1016/j.tcs.2015.12.035
0304-3975/

breaking. More specifically, we show how two mobile agents, identical in every aspect save their speed, can lever their speed difference in order to achieve fast rendezvous.

Symmetry breaking is a major issue in distributed computing that is completely absent from traditional sequential computing. Symmetry can often prevent different processes from reaching a common goal. Well known examples include leader election [1], mutual exclusion [2], agreement [3,4] and renaming [5]. To address this issue, various differences between processes are exploited. For example, solutions for leader election often rely on unique identifiers assumed to be associated with each entity (e.g., a process) [1]. Another example of a difference is the location of the entities in a network graph. Entities located in different parts of a non-symmetric graph can use this knowledge in order to behave differently; in such a case, a leader can be elected even without using unique identifiers [6]. If no differences exist, breaking symmetry deterministically becomes impossible (see, e.g., [1,7]) and one must resort to randomized algorithms, assuming that different entities can draw different random bits [8].

We consider mobile agents aiming to rendezvous. See, e.g., [9–14,7]. As is the case with other symmetry breaking problems, it is well known that if the agents are completely identical then rendezvous is, in some cases, impossible. In fact, a large portion of the research about rendezvous dealt with identifying the conditions under which rendezvous was possible, as a result of some asymmetries. Here, the fact that agents have different speeds implies that the mere feasibility of rendezvous is trivial, and our main concern is therefore the time complexity, that is, the time to reach a rendezvous. More specifically, we study the case where the agents are identical except for the fact that they have different speeds of motion. Moreover, to isolate the issue of the speed difference, we remove other possible differences between the agents. That is, the agents are assumed to be anonymous. To avoid solutions of the kind of [6], that are based on the underlying graph being asymmetric, we consider a symmetric topological object, that is, specifically, a cycle topology. We denote by n the length of the cycle.

1.2. The model and the problem

The problem of rendezvous on a cycle Consider two identical deterministic *agents* placed on a cycle of length n (in some distance units). To ease the description, we name these agents A and B but these names are not known to the agents. Each agent is initially placed in some location on the cycle by an adversary and both agents start the execution of the algorithm simultaneously. An agent can move on the cycle at any direction. Specifically, at any given point in time, an agent can decide to either start moving, continue in the same direction, stop, or change its direction. The agents' goal is to *rendezvous*, namely, to get to be co-located somewhere on the cycle.[2] We consider continuous movement, so this rendezvous can occur at any location along the cycle. An agent can detect the presence of another agent at its location and hence detect a rendezvous. When agents detect a rendezvous, the rendezvous task is considered completed.

Orientation issues We distinguish between two models based on orientation. The first assumes that agents have the *sense of direction* [16], that is, we assume that the agents can distinguish clockwise from the anti-clockwise. In the second model, we do not assume this orientation assumption. Instead, each agent has its own perception of which direction is clockwise and which is anti-clockwise, but there is no guarantee that these perceptions are globally consistent. (Hence, e.g., in this model, if both agents start walking in their own clockwise direction, they may happen to walk in opposite directions, i.e., towards each other).

The pebble *and the* white board *models* Although the agents do not hold any direct means of communication, in some cases, we do assume that an agent can leave marks in its current location on the cycle, to be read later by itself and by the other agent. In the *pebble* model, an agent can mark its location by dropping a pebble [17,18]. Both dropping and detecting a pebble are local acts taking place only on the location occupied by the agent. We note that in the case where pebbles can be dropped, our upper bound employs agents that drop a pebble only once and only at their initial location [19,20,14]. On the other hand, our corresponding lower bound holds for any mechanism of (local) pebble dropping. Moreover, this lower bound holds also for the seemingly stronger '*white board* model, in which an agent can change a memory associated with its current location such that it could later be read and further manipulated by itself or the other agent [9,21,22].

Speed Each agent moves at the same fixed speed at all times; the *speed* of an agent A, denoted $s(A)$, is the inverse of the time t_α it takes agent A to traverse one unit of length. For agent B, the time t_β and speed $s(B)$ are defined analogously. Without loss of generality, we assume that agent A is faster, i.e., $s(A) > s(B)$ but emphasize that this is unknown to the agents themselves. Furthermore, for simplicity of presentation, we normalize the speed of the slower agent B to one, that is, $s(B) = 1$ and denote $s(A) = c$ where $c > 1$. We stress that the more interesting cases are when c is a function of n and arbitrarily close to 1 (e.g., $c = 1 + 1/n^k$, for some constant k). We assume that each agent has a pedometer that enables it to measure the distance it travels.

[2] In a sense, this rendezvous problem is also similar to the *cow-path* problem, see, e.g., [15]. Here, the agents (the cow and the treasure she seeks to find) are both mobile (in the cow-path problem only one agent, namely, the cow, is mobile). It was shown in [15] that if the cow is initially located at distance D from the treasure on the infinite line then the time to find the treasure can be $9D$, and that 9 is the best multiplicative constant (up to lower order terms in D).