Theoretical Computer Science 547 (2014) 46–69

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

The minimum firing time of the generalized firing squad synchronization problem for squares

Kojiro Kobayashi

ARTICLE INFO

Article history:
Received 29 January 2013
Received in revised form 27 March 2014
Accepted 6 June 2014
Available online 12 June 2014
Communicated by D. Peleg

Keywords:
Firing squad synchronization problem
Cellular automata
Automata theory
Distributed computing

ABSTRACT

For almost all variations of the firing squad synchronization problem for elementary geometric figures such as lines, rings, squares, rectangles, and cubes, minimal-time solutions are known. However, in 2012 Umeo and Kubo introduced a very simple variation of this type and pointed out that its minimal-time solutions are unknown. In that variation, a problem instance is a square array of n columns and n rows and the position of the general is arbitrary. For this variation they constructed a solution that fires at time $2n-2$ for any position of the general and wrote that it is not known whether this solution is minimal-time or not. We determine the exact value of the minimum firing time of this variation. For some problem instances this value is smaller than $2n-2$ and hence the $2n-2$ time solution is not minimal-time. Our result does not solve the problem of existence or non-existence of minimal-time solutions of the variation. However the result gives one necessary condition for solutions to be minimal-time.

1. Introduction

1.1. The firing squad synchronization problem

The firing squad synchronization problem (FSSP) is a famous problem in automata theory [16]. It was posed almost half a century ago, but it still continues to attract interest of researchers. It is an interesting problem as a puzzle. However it is also a mathematical formulation of one of the core problems of distributed computing: how to realize global synchronization only with local information exchange. Solutions of the problem are basic components in the design of protocols in distributed computing.

The problem consists in designing a finite automaton A that fulfills several requirements. The automaton A has two inputs, one from the left and another from the right, and two outputs, one to the left and another to the right. The value of each output at a time is the state of the automaton at the time. We connect n copies of A as shown in Fig. 1. We call such a linear array of n copies of A a *configuration* of size n and denote it by C_n. We call each copy in C_n a *node* of C_n. We call the leftmost node the "general" of C_n. The values of the input from the left of the general and the input from the right of the rightmost node are a special symbol # denoting that there are no nodes there. The set of states of A must contain at least three different states G (the *general state*), Q (the *quiescent state*) and F (the *firing state*). When the state of a node is Q and both of its inputs are either Q or # at a time t, the state of the node at time $t+1$ must be Q again. At time 0, the state of a node is G if it is the general and Q if not. Then the state of any node v of C_n at any time t is uniquely determined by the state transition function of the automaton A. We denote this state by $\mathrm{st}(v, t, C_n, A)$. The finite automaton A must satisfy the following condition: for any size n there exists a time t_n (that may depend on n) such that, for any node v of C_n,

E-mail address: kojiro@gol.com.

http://dx.doi.org/10.1016/j.tcs.2014.06.016
0304-3975/

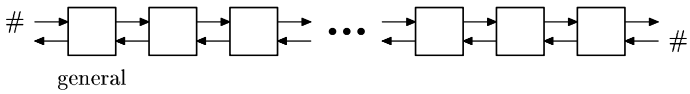

Fig. 1. A configuration C_n of FSSP.

$\mathrm{st}(v, t, C_n, A) \neq \mathrm{F}$ for any $t < t_n$ and $\mathrm{st}(v, t_n, C_n, A) = \mathrm{F}$. We call a finite automaton A that fulfills all of these requirements a *solution* of FSSP. We call the time t_n the *firing time* of the configuration C_n of the solution A and denote it by $\mathrm{ft}(C_n, A)$. We also say that A *fires* C_n at time t_n. According to [16], this problem was first solved by John McCarthy and Marvin Minsky.

It is not difficult to construct a solution and usually we construct a solution A such that $\mathrm{ft}(C_n, A) = 3n + O(\log n)$. We can easily show a lower bound $\mathrm{ft}(C_n, A) \geq 2n - 2$ for any solution A and n. Intuitively, the general needs at least time $2n - 2$ to know the position of the rightmost node (the minimum time for a signal to go from the general to the rightmost node and then come back to the general). Formalizing this intuitive reasoning we obtain a rigorous mathematical proof of this lower bound.

In 1962 Goto constructed a solution $\tilde{A}$ such that $\mathrm{ft}(C_n, \tilde{A}) = 2n - 2$ for any n [6]. For this solution $\tilde{A}$ we have $\mathrm{ft}(C_n, \tilde{A}) \leq \mathrm{ft}(C_n, A)$ for any solution A and n, that is,

$$(\forall A)(\forall n) \quad \big[\mathrm{ft}(C_n, \tilde{A}) \leq \mathrm{ft}(C_n, A)\big]. \tag{1}$$

Any solution $\tilde{A}$ that satisfies (1) has been called a *minimal-time solution*. Later we give another definition of minimal-time solutions. Hence, to distinguish the above-mentioned definition from the later definition, we call a solution $\tilde{A}$ that satisfies (1) a *first type minimal-time solution*. Later Waksman and Balzer constructed simpler first type minimal-time solutions [1,27]. Thus, the problem to find best solutions with respect to firing time was essentially solved. However, the problem to find simpler solutions (easy to understand or having small number of states) continues to be investigated.

1.2. Variations of the firing squad synchronization problem

After the original FSSP was introduced, many variations have been introduced and studied. The following is a list of some of the variations for which first type minimal-time solutions were obtained:

- the generalized FSSP, where configurations are linear arrays shown in Fig. 1 but the general may be any node [17,19,21,26],
- FSSP for squares such that the general is at one of the corners [20],
- FSSP for rectangles such that the general is at one of the corners [20],
- FSSP for rectangles such that the general may be any node [21,24],
- FSSP for rings [2,3],
- FSSP for one-way rings (that is, with one-way information flow) [3,11,13].

For surveys on such variations we refer the reader to [15,23].

For each of these variations first type minimal-time solutions were found in the following steps: (1) the variation was formalized and stated; (2) a solution was shown; (3) a solution $\tilde{A}$ with small firing time was shown; (4) the firing time of $\tilde{A}$ was shown to be a lower bound of the firing time. Usually (3) is difficult and (4) is easy. We know that the FSSP for directed networks has a solution [8] and hence any variation such that its configurations are directed graphs has a solution. Hence, for almost all cases the step (2) is not necessary.

The original FSSP itself continues to be studied extensively and for these results see [23,28–31].

1.3. Variations with no known minimal-time solutions

There remain some variations for which we know no first type minimal-time solutions. For surveys on such variations we refer the reader to [4,5,10]. We list four of such variations.

- FSSP for paths in the two-dimensional grid space (2PATH). A configuration of this variation is a path in the two-dimensional grid space and the general is one of the two end nodes.
- The three-dimensional analogue of the above variation (3PATH).
- FSSP for undirected networks (UN).
- FSSP for directed networks (DN).

At present we know no first type minimal-time solutions for these variations and we know no proofs of their non-existence. However we could show that existence of first type minimal-time solutions of these variations implies existence of efficient algorithms for some (seemingly) difficult problems.

For 2PATH, we showed that if it has a minimal-time solution then a problem called the *two-dimensional path extension problem* (2PEP) has a polynomial time algorithm [10]. In 2PEP a path in the two-dimensional grid space is given and we