Applied Mathematics and Computation 000 (2017) 1–17

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

# A new reconstruction and the first implementation of Goto's FSSP algorithm

Hiroshi Umeo*, Mitsuki Hirota, Youhei Nozaki, Keisuke Imai, Takashi Sogabe

*University of Osaka Electro-Communication, Neyagawa-shi, Hatsu-cho, 18-8, 572-8530 Osaka, Japan*

ARTICLE INFO

*Article history:*
Available online xxx

*Keywords:*
Cellular automaton
Firing squad synchronization problem
FSSP

ABSTRACT

The firing squad synchronization problem (FSSP) on cellular automata has been studied extensively for more than fifty years, and a rich variety of synchronization algorithms has been proposed. Goto's FSSP algorithm (Goto 1962) has been known as the first minimum-time FSSP algorithm, however the paper itself had been a completely unknown one in the research community of cellular automata for a long time due to its hard accessibility. In the present paper, we reconstruct the Goto's FSSP algorithm and present the first small-state implementation. The implementation is realized on a cellular automaton having 165-state and 4378 state-transition rules and the realization is far smaller than Goto (1962) imagined, where he thought that it would require many thousands of thousands states. It is shown that the reconstructed algorithm uses a quite different synchronization mechanism in comparison with the designs employed in Waksman (1966), Balzer (1967), Gerken (1987) and Mazoyer (1987). We show that the algorithm has $\Theta(n \log n)$ minimum-state-change complexity for synchronizing $n$ cells. The algorithm is optimum not only in time but also in state-change complexities. We show that the reconstructed algorithm can be generalized as to the initial general's position and its implementation on a cellular automaton with 434 internal states and 13,328 state-transition rules is also given. The general purpose of this investigation is to achieve more insights into the structure of the classical minimum-time FSSP solutions and such insights would be helpful in the design of new FSSP algorithms.

## 1. Introduction

We study a synchronization problem that gives a finite-state communication protocol for synchronizing large-scale cellular automata. The synchronization in cellular automata has been known as the firing squad synchronization problem (FSSP) since its development, in which it was originally proposed by Myhill in Moore [8] to synchronize all/some parts of self-reproducing cellular automata. The problem has been studied extensively for more than fifty years, and a rich variety of synchronization algorithms has been proposed not only for one-dimensional (1D) arrays but also for multi-dimensional arrays.

Historically, the FSSP problem was first studied by McCarthy and M. Minsky [7] who presented a $3n$-step non-minimum-time algorithm for 1D cellular array of length $n$. In 1962, the first minimum-time, i.e. $(2n-2)$-step synchronization algorithm was presented by Goto [3], with each cell having many thousands of thousands states (according to Goto's statements

* Corresponding author.
*E-mail address:* umeo@cyt.osakac.ac.jp (H. Umeo).

http://dx.doi.org/10.1016/j.amc.2017.05.015
0096-3003/© 2017 Elsevier Inc. All rights reserved.

in Goto [4]). Afterward, Waksman [18], Balzer [1], Gerken [2], and Mazoyer [5] presented a 16-state, 8-state, 7-state, and 6-state minimum-time synchronization algorithm, respectively, thus decreasing the number of states required for the synchronization.

In the present paper, we reconstruct Goto's FSSP algorithm and present the first small-state implementation on a finite-state automaton. The Goto's algorithm has been known as the first FSSP algorithm to synchronize 1D array in minimum-time. The first minimum-time algorithm was presented as a technical lecture note in Goto [3]. The paper has been usually cited as the first minimum-time FSSP algorithm, but the original paper had been a completely unknown one for a long time in the community of cellular automata due to its hard accessibility, until Umeo [10] gave a talk on the reconstruction of the Goto's algorithm at the first IFIP cellular automata workshop in 1996.

In this paper, specifically, we attempt to answer the following questions:

- What is the first Goto's FSSP algorithm?
- What is the difference among many minimum-time FSSP algorithms, such as Goto [3], Waksman [18], Balzer [1], Gerken [2], and Mazoyer [5]?
  - Waksman [18], Balzer [1] and Gerken [2] algorithms are based on a recursive halving marking and they are two-sided recursive algorithms.
  - Mazoyer [5] is a one-sided FSSP algorithm which is based on a recursive division with a 1/3 ratio.
  - The reconstructed algorithm we show here is quite different from the above designs. It is based on an exponential marking and it is a non-recursive FSSP algorithm.
- How many states are required for its implementation?
  - It has been said by Goto himself [4] that many thousands of thousands states would be required for its realization.
  - It is shown that the reconstructed algorithm we show here is realized on a finite-state automaton only with 165 states and 4378 state-transition rules, where these amounts are far smaller than being thought.
- How can we synchronize 1D arrays without calling itself recursively?
  - It is shown that the reconstructed algorithm consists of two operation phases, the first one is an exponential marking and the second one is a synchronization phase which calls a simple $3n$-step FSSP algorithm.
- What is the state-change complexity of the reconstructed algorithm for synchronizing $n$ cells?
  - A state-change complexity is an important complexity measure to evaluate the efficiency of cellular automata, motivated by energy consumption in certain SRAM-type memory systems.
  - We show that the reconstructed algorithm has $\Theta(n \log n)$ minimum-state-change complexity for synchronizing $n$ cells.
  - The algorithm is optimum not only in time but also in state-change complexity.
- Can we generalize the reconstructed algorithm to the case where the initiator can be located at any position of the array, while keeping the minimum-time optimality?
  - Yes, it can be generalized so that the synchronization operation can be started from any position of the array. It is shown that the generalized FSSP (GFSSP) algorithm constructed here can synchronize any 1D array of length $n$ in minimum $n - 2 + \max(k, n - k + 1)$ steps, where the general is located on the $k$th cell from the left end.
  - The GFSSP algorithm is optimum not only in time but also in state-change complexity.
  - The realized minimum-time GFSSP algorithm can be implemented on a cellular automaton with 434 internal states and 13328 state-transition rules and has a minimum-state-change complexity.

Section 2 gives a definition of the FSSP. In Section 3 we present a reconstruction of the Goto's FSSP algorithm and give the first small-state implementation of the reconstructed algorithm. We also give a generalization of the reconstructed algorithm as to the initial position of the general and consider the algorithms from a point of state-change complexity. Section 4 gives a summary of the paper and several discussions on the reconstruction.

## 2. Firing squad synchronization problem

### 2.1. FSSP on one-dimensional array

The FSSP is formalized in terms of the model of cellular automata. Consider a 1D array of finite state automata. All cells (except the end cells) are identical finite state automata. The array operates in lock-step mode such that the next state of each cell (except the end cells) is determined by both its own present state and the present states of its right and left nearest neighbors. All cells (*soldiers*), except one *general* cell at the left end of the array, are initially in the *quiescent* state at time $t = 0$ and have the property that the next state of a quiescent cell having quiescent neighbors is the quiescent state. At time $t = 0$ the *general* cell is in the *fire-when-ready* state, which is an initiation signal to the array.

The FSSP is stated as follows: given a 1D array of $n$ identical cellular automata, including a *general* at one end that is activated at time $t = 0$, we want to design the automaton $M = (\mathcal{Q}, \delta)$ such that *at some future time* all the cells will *simultaneously* and *for the first time* enter a special *firing* state, where $\mathcal{Q}$ is a finite state set and $\delta : \mathcal{Q}^3 \rightarrow \mathcal{Q}$ is a next-state function. The tricky part of the problem is that the same kind of soldier having a fixed number of states must be synchronized, regardless of length $n$ of the array. The set of states and the next-state function must be independent of $n$.

Fig. 1 shows a finite 1D cellular array consisting of $n$ cells, denoted by $\mathrm{C}_i$, where $1 \leq i \leq n$. Without loss of generality, we assume $n \geq 2$.