2009 International Symposium on Nonlinear Theory and its Applications
NOLTA'09, Sapporo, Japan, October 18-21, 2009

Small Non-Optimum-Time Firing Squad Synchronization Protocols for One-Dimensional Rings

Hiroshi UMEO†, Jean-Baptiste YUNÈS††, Naoki KAMIKAWA†, and Juntarou KURASHIKI†

† Univ. of Osaka Electro-Communication,
Faculty of Information Science and Technology,
Neyagawa-shi, Hatsu-cho, 18-8, Osaka, 572-8530, Japan
E-mail: {umeo, kamikawa, kurashiki}@cyt.osakac.ac.jp
†† LIAFA - Universite Paris 7 Denis Diderot,
175, rue du chevaleret, 75013 Paris - France,
E-mail: Jean-Baptiste.Yunes@liafa.jussieu.fr

Abstract—The synchronization in cellular automata has been known as a firing squad synchronization problem since its development, in which it was originally proposed by J. Myhill in Moore [1964] to synchronize all or some parts of self-reproducing cellular automata. The firing squad synchronization problem has been studied extensively for more than 40 years [1-22]. In the present article, we propose two six-state firing squad synchronization *full* protocols for rings, which are the smallest ones known at present for rings. In addition, we present a family of 4-state *partial* protocols that can synchronize any one-dimensional rings of length $n = 2^k$ for any positive integer k. The number $four$ is the smallest one in the class of synchronization protocols proposed so far. We also study state change complexities for those protocols.

1. Introduction

Cellular automata are considered to be a nice model of complex systems in which an infinite one-dimensional array of finite state machines (cells) updates itself in synchronous manner according to a uniform local rule. We study a synchronization problem that gives a finite-state protocol for synchronizing a large scale of cellular automata. The synchronization in cellular automata has been known as a firing squad synchronization problem (FSSP) since its development, in which it was originally proposed by J. Myhill in Moore [1964] to synchronize all or some parts of self-reproducing cellular automata. The FSSP has been studied extensively for more than 40 years [1-22]. The optimum-time (i.e., $(2n-2)$-step) synchronization algorithm for one-dimensional array of length n was devised first by Goto [1962]. The algorithm needed many thousands of internal states for its realization. Afterwards, Waksman [1966], Balzer [1967], Gerken [1987] and Mazoyer [1987] developed an optimum-time algorithm and reduced the number of states realizing the algorithm, each with 16, 8, 7 and 6 states. On the other hand, Balzer [1967], Sanders [1994] and Berthiaume et al. [2004] studied the state lower bounds for realizing synchronization and have shown that there exists no four-state synchronization algorithm. Thus, an existence or non-existence of five-state firing squad synchronization protocol has been a longstanding open problem. One has to note that any solution in the original problem is to synchronize any array of length greater than two. We call it **full** solution. Umeo and Yanagihara [2007] initiated an investigation on the FSSP solutions that can synchronize an infinite set of arrays, but not all, and presented a five-state $3n + O(1)$ step algorithm that can synchronize any one-dimensional cellular array of length $n = 2^k$ for any positive integer k in $3n - 3$ steps. Recently, Yunès [2008] and Umeo, Yunès, and Kamikawa [2008] developed 4-state protocols based on Wolfram's rule 60 and 150. We call such protocol as **partial** solution. Umeo, Kamikawa, and Yunès [2008] also have given an answer partially to the problem by proposing a family of smallest four-state firing squad synchronization protocols that can synchronize any one-dimensional ring cellular array of length $n = 2^k$ for any positive integer k. The number $four$ is the smallest one in states required in the class of synchronization protocols proposed so far. In the present article, we propose several six-state firing squad synchronization *full* protocols for rings, which are the smallest ones known at present for rings. In addition, we present a family of 4-state *partial* protocols that can synchronize any one-dimensional ring cellular arrays of length $n = 2^k$ for any positive integer k. We also study state change complexities for those protocols.

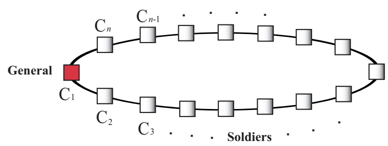

Figure 1: A one-dimensional ring cellular automaton.

2. Firing Squad Synchronization Problem on Rings

2.1. Firing Squad Synchronization Problem

Figure 1 shows a finite one-dimensional ring cellular array consisting of n cells. Each cell is an identical finite-state automaton. The array operates in lock-step mode in such a way that the next state of each cell is determined by both its own present state and the present states of its left and right neighbors. All cells (*soldiers*), except one cell (*general*), are initially in the quiescent state at time $t = 0$ with the property that the next state of a quiescent cell with quiescent neighbors is the quiescent state again. At time $t = 0$, the one cell, for example C_1, is in the *fire-when-ready* state, which is the initiation signal for the array. The firing squad synchronization problem (FSSP) is to determine a description (state set and next-state function) for cells that ensures all cells enter the *fire* state at exactly the same time and for the first time. The set of states and the next-state function must be independent of n.

2.2. Complexity Measures in FSSP

2.2.1. Time Complexity

Any solution to the firing squad synchronization problem for rings can be shown to require n-steps for synchronizing n cells, since signals on the array can propagate no faster than one cell per step, and the time from the general's instruction until the synchronization must be at least n. It has been shown by Berthiaume et al. [2004] that there exists no algorithm that can synchronize any ring of length n in less than n steps.

Theorem 1[Berthiaume, Bittner, Perkovic, Settle, and Simon [2004]] The minimum time in which the firing squad synchronization could occur is no earlier than n steps for any ring of length n.

2.2.2. Number of States

The following three distinct states: the *quiescent* state, the *general* state, and the *firing* state, are required in order to define any cellular automaton that can solve the FSSP. Balzer [1967] implemented a search strategy in order to prove that there exists no four-state *full* solution. He showed that no four-state optimum-time *full* solution exists. Sanders [1994] studied a similar problem on a parallel computer and gave a proof based on a computer simulation for the non-existence of four-state *full* solution. The question that remains is: "What is the minimum number of states for an optimum-time solution of the problem?"At present, that number is *five* or *six.* Umeo and Yanagihara [2007], Yunès [2008], and Umeo, Yunès, and Kamikawa [2008] gives some 4- and 5-state *partial* solutions that can synchronize infinite cells, but not all.

Theorem 2[Balzer[1967], Sanders[1994]] There is no four-state *full* solution that can synchronize n cells.

Berthiaume, Bittner, Perković, Settle and Simon [2004] considered the state lower bound on ring-connected cellular automata. It is shown that there exists no three-state solution and no four-state symmetric solution for rings.

Theorem 3[Berthiaume et al.[2004]] There is no four-state symmetric optimum-time *full* solution for ring cellular automata.

Theorem 4[Umeo, Yunes, and Kamikawa [2008], Yunes [2008]] There exist 4-state *partial* solutions to the firing squad synchronization problem for the rings.

2.2.3. State-Change Complexity

Vollmar [1982] introduced a state-change complexity in order to measure the efficiency of cellular algorithms and showed that $\Omega(n \log n)$ state-changes are required for the synchronization of n cells in $(2n - 2)$ steps.

Theorem 5[Vollmar [1982]] $\Omega(n \log n)$ state-change is necessary for synchronizing n cells in n steps.

3. Ring Solutions

3.1. Optimum-Time 8-State Full Solution

Berthiaume, Bittner, Perković, Settle and Simon [2004] proposed an 8-state *full* solution operating exactly in optimum-step.

Theorem 6[Berthiaume, Bittner, Perkovic, Settle, and Simon [2004]] There exists an 8-state solution that can synchronize any ring of length n exactly in optimum n-steps.

3.2. Non-Optimum-Time 6-State Full Solutions

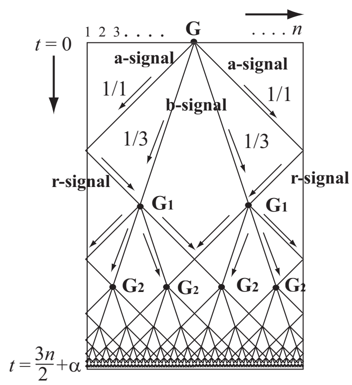

Figure 2: A time-space diagram for finite-width thread-like non-optimum-step firing squad synchronization algorithm.

Figure 2 shows a time-space diagram for the well-known non-optimum-time firing squad synchronization algorithm for rings. The synchronization process can be viewed as a typical divide-and-conquer strategy that operates in parallel in the cellular space. An initial "*General*" G, located at an arbitrary cell of the array of size n, generates two special signals, referred to as *a-signal* and *b-signal*, which propagate in the right and left directions at speed of 1/1 (i.e., 1 cell per unit step) and 1/3 (1 cell per three steps), respectively. The a-signal collides with each other at time $t = n/2$, reflects there immediately, and then continues to move at the same speed in the left and right directions. The reflected signal is referred to as *r-signal.* The b- and r-signals meet at a quarter cell(s), depending on the parity of n. In the case that n is odd, the cells $C_{\lceil n/4 \rceil}$ and $C_{\lceil 3n/4 \rceil}$ become a *General* at time $t = 3\lceil n/4 \rceil - 2$. The new *Generals* work for synchronizing both its left and right halves of

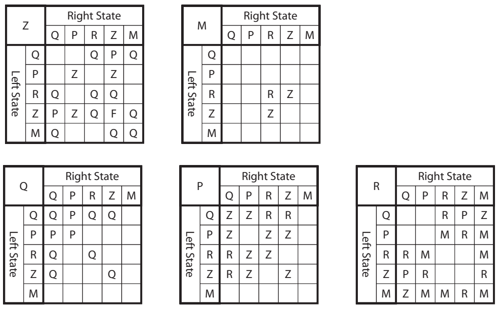

Figure 3: A state transition table for the six-state symmetrical full protocol.

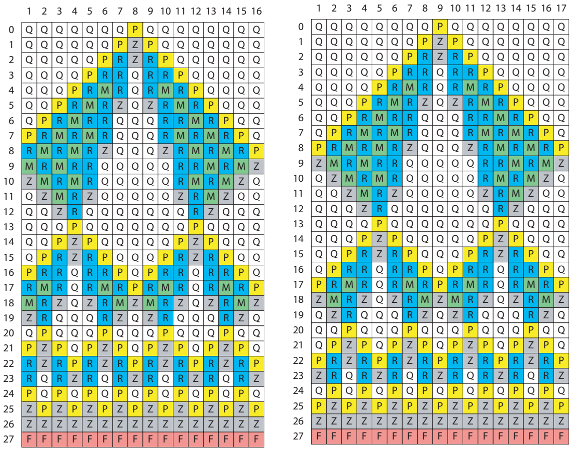

Figure 4: Snapshots for the 6-state symmetrical firing squad synchronization algorithm on 16 and 17 cells.

the cellular space. Note that the *General* is shared by the two halves. In the case that n is even, two cells $C_{\lceil n/4 \rceil}$ and $C_{\lceil n/4 \rceil+1}$ become the next *Generals* at time $t = 3\lceil n/4 \rceil$. Each *General* works for synchronizing its left and right quarters of the cellular space, respectively. Thus at time $t = t_{G_1}$

$$t_{G_1} = \begin{cases} 3\lceil n/4 \rceil - 2 & n\text{: odd} \\ 3\lceil n/4 \rceil & n\text{: even,} \end{cases} \tag{1}$$

the array knows its quarter point(s) and generates one or two new *General*(*s*) G_1 at the quarter cell(s). The new *General*(*s*) G_1 generates the same 1/1- and 1/3-speed signals in both left and right directions and repeats the same procedures as above. Thus, the original synchronization problem of size n is divided into four sub-problems of size $\lceil n/4 \rceil$. In this way, the original array is split into equal two, four, eight, ..., subspaces synchronously. In the last, the original problem of size n can be split into small sub-problems of size 2. Based on the time-space diagram shown in Fig. 2, we provide two 6-state full protocols. Figure 3 illustrates the symmetrical transition table and snapshots of the algorithm are given in Fig. 4. The 6-state algorithm can synchronize any ring of length n in $3n/2 + \mathrm{O}(\log n)$ steps. The state-change-complexity is $\mathrm{O}(n^2)$. The other 6-state algorithm has $O(n \log n)$-state-change complexity. It can synchronize any ring of length n in $3n/2 + \mathrm{O}(\log n)$ steps. Figure 5 illustrates the transition table and snapshots of the synchronization processes are given in Fig. 6. A proof for the correctness of those algorithms is omitted. We have:

Theorem 7 There exists six-state full solutions that can synchronize any ring of length n in $3n/2 + \mathrm{O}(\log n)$ steps.

3.3. Optimum-Time 4-State Partial Solution

We present four optimum-time *partial* solutions 1, 2, 3, and 4 each operating in exactly n steps for any ring of length $n = 2^k, k \geq 1$, where k is any positive integer. See Umeo, Kamikawa and Yunès [2009] for details. Figure 7 shows the transition rules and snapshots on 16 cells for Solution 1. It is noted that both of the states G and A can be an initial general state in each solution without introducing any additional transition rules. Let $T_{i,G}(n)$, $T_{i,A}(n)$ be time complexity of Solution i for synchronizing

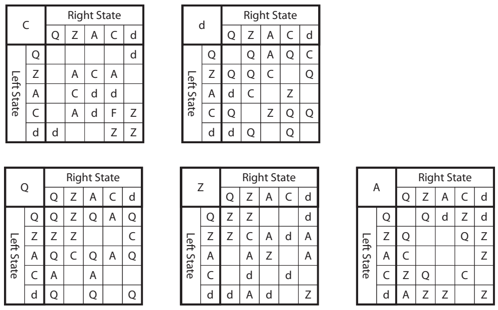

Figure 5: A state transition table for the six-state thread-like algorithm.

a ring CA of length n with an initial general in state G, A, respectively. We get the following theorem.

Theorem 8 For any i such that $1 \leq i \leq 4$, $T_{i,G}(n) = T_{i,A}(n) = n$, where $n = 2^k, k \geq 1$.

4. Conclusions

We have presented two six-state *full* FSSP protocols for rings, which are the smallest non-optimum-time ones known at present for rings. One solution has $O(n^2)$ and the other has optimum $\Omega(n \log n)$ state-change complexity, respectively.

Acknowledgment A part of this work is supported by Grant-in-Aid for Scientific Research (C) 21500023.

References

[1] R. Balzer: An 8-state minimal time solution to the firing squad synchronization problem. *Information and Control*, vol. 10(1967), pp. 22-42.

[2] A. Berthiaume, T. Bittner, L. Perkovic, A. Settle and J. Simin: Bounding the firing squad synchronization

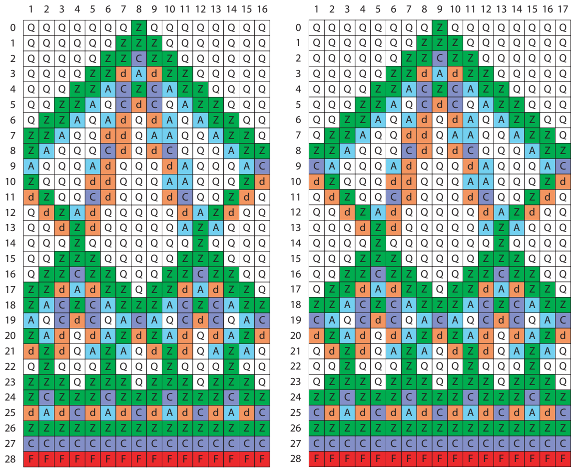

Figure 6: Snapshots for the 6-state thread-like firing squad synchronization algorithm on 16 and 17 cells.

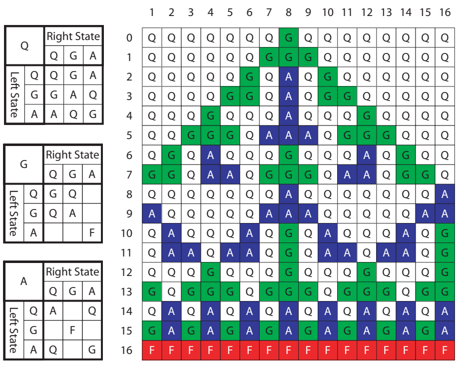

Figure 7: Transition table and snapshots with an initial general in state G on 16 cells for the Solution 1.

problem on a ring. *Theoretical Computer Science*, 320 (2004), 213-228.

[3] K. Culik: Variations of the firing squad problem and applications. *Information Processing Letters*, 30, (1989), 153-157.

[4] H. D. Gerken: Über Synchronisations - Probleme bei Zellularautomaten. *Diplomarbeit*, Institut für Theoretische Informatik, Technische Universität Braunschweig, (1987), pp. 50.

[5] E. Goto: A minimal time solution of the firing squad problem. *Dittoed course notes for Applied Mathematics* 298, Harvard University, (1962), pp. 52-59.

[6] Y. Kobuchi: A note on symmetrical cellular spaces. *Information Processing Letters*, 25 (1987), 413-415.

[7] J. Mazoyer: A six-state minimal time solution to the firing squad synchronization problem. *Theoretical Computer Science*, vol. 50 (1987), pp. 183-238.

[8] M. L. Minsky: *Computation: Finite and infinite machines*. Prentice Hall, (1967), pp. 28-29.

[9] E. F. Moore: The firing squad synchronization problem. in *Sequential Machines, Selected Papers* (E. F. Moore, ed.), Addison-Wesley, Reading MA., (1964), pp. 213-214.

[10] A. Settle and J. Simon: Smaller solutions for the firing squad. *Theoretical Computer Science*, 276 (2002), 83-109.

[11] H. Szwerinski: Symmetrical one-dimensional cellular spaces. *Information and Control*, vol. 67(1982), 163-172.

[12] H. Umeo: Firing Squad Synchronization Problem in Cellular Automata. In *Encyclopedia of Complexity and System Science*, R. A. Meyers (Ed.), Springer, Vol.4 (2009), pp.3537-3574.

[13] H. Umeo, M. Hisaoka and T. Sogabe: A survey on firing squad synchronization algorithms for one-dimensional cellular automata. International Journal of Unconventional Computing, Vol.1(2005), pp.403-426.

[14] H. Umeo, M. Maeda and K. Hongyo: A design of symmetrical six-state $3n$-step firing squad synchronization algorithms and their implementations. *Proc. of 7th International Conference on Cellular Automata for Research and Industry, ACRI2006*, LNCS 4173(2006), pp.157-168.

[15] H. Umeo and T. Yanagihara: A small five-state non-optimum-time solution to the firing squad synchronization problem - a geometrical approach. Fundamenta Informaticae, 91 (2009), 161-178.

[16] H. Umeo, J. B. Yunès, and N. Kamikawa: About 4-state solutions to the firing squad synchronization problem. *Proc. of 8th International Conference on Cellular Automata for Research and Industry: ACRI 2008, LNCS 5191*, (2008), pp.108-113.

[17] H. Umeo, N. Kamikawa, and J. B. Yunès: A family of smallest symmetrical four-state firing squad synchronization protocols for ring arrays. Parallel Processing Letters, Vol.19, No.2 (2009), pp.299-313.

[18] R. Vollmar: On cellular automata with a finite number of state changes. *Computing, Supplementum*, vol. 3(1981), pp.181-191.

[19] A. Waksman: An optimum solution to the firing squad synchronization problem. *Information and Control*, vol. 9 (1966), pp. 66-78.

[20] J. B. Yunès: Seven-state solution to the firing squad synchronization problem. *Theoretical Computer Science*, 127(1994), pp.313-332.

[21] J. B. Yunès: An intrinsically non minimal-time Minsky-like 6 states solution to the firing squad synchronization problem. *Theoretical Informatics and Applications*, Vol.42, No.1(2008), pp.55-68.

[22] J. B. Yunès: A 4-states algebraic solution to linear cellular automata synchronization. *Information Processing Letters*, Vol. 19, Issue 2(2008), pp.71-75.