Proceedings of the 5th WSEAS International Conference on Applied Computer Science, Hangzhou, China, April 16-18, 2006 (pp596-600)

State-efficient Time-optimum Synchronization Protocols for Two-dimensional Arrays -A Survey-

Hiroshi Umeo
Univ. of Osaka Electro-Communication, Faculty of Information Science and Technology,
Neyagawa-shi, Hatsu-cho 18-8, Osaka, 572-8530, Japan

Abstract: The firing squad synchronization problem has been studied extensively for more than forty years, and a rich variety of synchronization algorithms have been proposed. In the present paper, we give a survey on recent developments in firing squad synchronization algorithms for synchronizing large-scale two-dimensional cellular automata. Several new algorithms and their state-efficient implementations are also given.
Key words: cellular automaton, firing squad synchronization problem

1 Introduction

We study a synchronization problem which 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 since its development, in which it was originally proposed by J. Myhill to synchronize all parts of self-reproducing cellular automata [7]. The problem has been studied extensively for more than 40 years [1-18]. In this paper, we give a survey on recent developments in designing optimum-time synchronization algorithms and their implementations for two-dimensional arrays. Several simple and state-efficient mapping schemes are proposed for embedding one-dimensional (1-D) firing squad synchronization algorithms onto two-dimensional (2-D) arrays, and some new 2-D synchronization algorithms based on the mapping scheme are presented.

2 Firing Squad Synchronization Problem on Two-dimensional Arrays

Figure 1 shows a finite two-dimensional cellular array consisting of $m \times n$ cells. Each cell is an identical (except the border cells) finite-state automaton. The array operates in lock-step mode in such a way that the next state of each cell (except border cells) is determined by both its own present state and the present states of its north, south, east and west neighbors. All cells (*soldiers*), except the north-west corner 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 north-west corner cell C_1 is in the *fire-when-ready* state, which is the initiation signal for the array. The firing squad synchronization problem 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 must be independent of n. The tricky part of the problem is that the same kind of soldier having a fixed number of states must be synchronized, regardless of the size $m \times n$ of the array. The set of states must be independent of m and n.

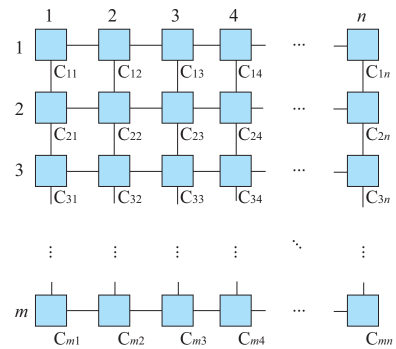

Figure 1: A two-dimensional cellular automaton.

The problem was first solved by J. McCarthy and M. Minsky who presented a 3n-step algorithm. In 1962, the first optimum-time, i.e. $(2n-2)$-step, synchronization algorithm was presented by Goto [1962], with each cell having several thousands of states. Waksman [1966] presented a 16-state optimum-time synchronization algorithm. Afterward, Balzer [1967] and Gerken [1987] developed an eight-state algorithm and a seven-state synchronization algorithm, respectively, thus decreasing the

Proceedings of the 5th WSEAS International Conference on Applied Computer Science, Hangzhou, China, April 16-18, 2006 (pp596-600)

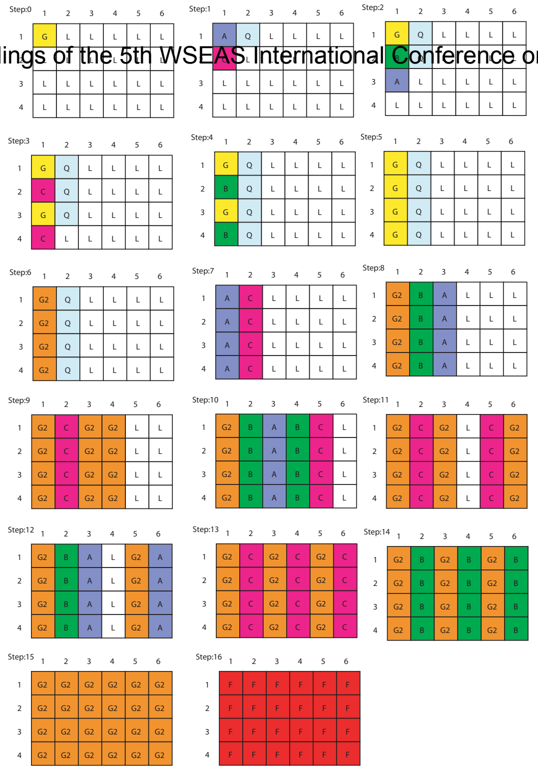

Figure 2: Snapshots of the synchronization process on 4×6 array.

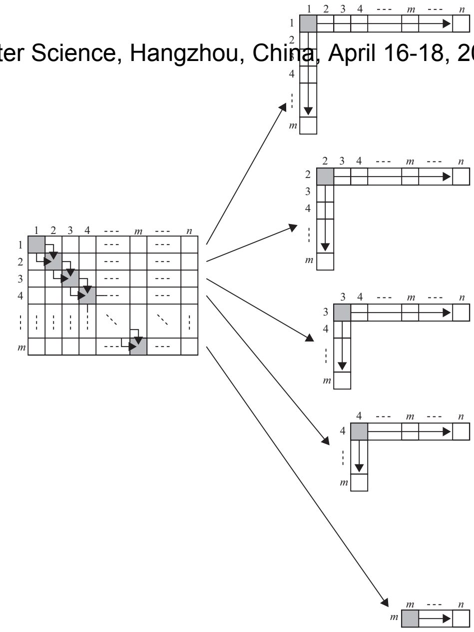

Figure 3: An optimum-time synchronization scheme for rectangular arrays.

number of states required for the synchronization. Mazoyer [1987] developed a six-state synchronization algorithm which, at present, is the algorithm having the fewest states. Several synchronization algorithms on 2-D arrays have been proposed by Beyer [1969], Grasselli [1975], Kobayashi [1977], Shinahr [1974], Szwerinski [11982] and Umeo, Maeda and Fujiwara [2002].

3 Orthogonal Mapping: A Simple Linear-time Algorithm $\mathcal{A}_1$

In this section, we give a very simple synchronization algorithm for 2-D arrays. The overall of the algorithm is as follows: First, *synchronize* the first column cells using optimum-step 1-D algorithm with a general at one end, thus requiring $2m - 2$ steps. Then, *start the row synchronization operation* on each row simultaneously. Additional $2n - 2$ steps are required for the row synchronization. Totally, its time complexity is $2(m + n) - 4$ steps. We refer the implementation as *orthogonal mapping*. It is shown that $s + 2$ states are enough for the implementation of the algorithm above, where s is the number of internal states of the 1-D base algorithm. In Fig. 2, we show snapshots of our 8-state synchronization algorithm running on a rectangular array of size 4×6.

[Theorem 1] There exists an $(s + 2)$-state protocol for synchronizing any $m \times n$ rectangular arrays in $2(m+n)-4$ steps, where s is number of states of any optimum-time one-dimensional synchronization protocol.

4 L-shaped Mapping: Shinar's Optimum-time Algorithm $\mathcal{A}_2$

The first optimum-time synchronization algorithm was developed by Shinar [1974] and Beyer [1969]. The rectangular array of size $m \times n$ is regarded as $\min(m, n)$ L-shaped 1-D arrays, where they are synchronized independently using the generalized firing squad synchronization algorithm. The configuration of the generalized firing on 1-D arrays can be mapped on 2-D array. Thus, an $m \times n$ array synchronization problem is reduced to $\min(m, n)$ independent 1-D generalized synchronization problems: $\mathcal{P}(m, m + n - 1)$, $\mathcal{P}(m - 1, m + n - 3)$,..., $\mathcal{P}(1, n - m)$. Beyer [1969] and Shinahr [1974] presented an optimum-time synchronization scheme in order to synchronize any $m \times n$ arrays in steps.

[Theorem 2] There exists an optimum-time $(2s \pm O(1))$-state protocol for synchronizing any $m \times n$ rectangular arrays in $m + n + max(m, n) - 3$ steps, where s is number of states of any optimum-time one-dimensional synchronization protocol.

Shinahr [1974] has given a 28-state implementation.
[Theorem 3]$^{\text{Shinahr[1974]}}$ There exists an optimum-time 28-state protocol for synchronizing any $m \times n$ rectangular arrays in $m + n + max(m, n) - 3$ steps.

5 Diagonal Mapping I: Six-state Linear-time Algorithm $\mathcal{A}_3$

The proposal is a simple and efficient mapping scheme that enables us to embed any one-dimensional firing squad synchronization algorithm onto two-dimensional arrays without introducing additional states. We consider a 2-D array of size $m \times n$, where $m, n \geq 2$. We divide mn cells into $m+n-1$ groups g_k, $1 \leq k \leq m+n-1$, defined as follows.

$$g_k = \{\mathrm{C}_{i,j} | (i-1)+(j-1) = k-1\}, \text{ i.e.,}$$

$g_1 = \{\mathrm{C}_{1,1}\}$, $g_2 = \{\mathrm{C}_{1,2}, \mathrm{C}_{2,1}\}$, $g_3 = \{\mathrm{C}_{1,3}, \mathrm{C}_{2,2}, \mathrm{C}_{3,1}\}$, . . . , $g_{m+n-1} = \{\mathrm{C}_{m,n}\}$.

Figure 4 shows the division of the two-dimensional array of size $m \times n$ into $m+n-1$ groups. For convenience, we define $g_0 = \{\mathrm{C}_{0,0}\}$ and $g_{m+n} = \{\mathrm{C}_{m+1,n+1}\}$.

Let $M = (Q, \delta_1, w)$ be any one-dimensional CA that fires ℓ cells in $T(\ell)$ steps, where Q is the finite state set of M, $\delta_1 : Q^3 \rightarrow Q$ is the transition function, and $w \in Q$ is the state of the right and left ends. We assume that M has $m+n-1$ cells, denoted by C_i, where $1 \leq i \leq m+n-1$. For convenience, we assume that M has a left and right end cells, denoted by C_0 and C_{m+n}, respectively. Both end cells C_0 and C_{m+n} always take the end state $w (\in Q)$. We consider the one-to-one correspondence between the ith group g_i and the ith cell C_i on M such that $g_i \leftrightarrow \mathrm{C}_i$, where $1 \leq i \leq m+n-1$ (see Fig. 4). We can construct a 2-D CA $N = (Q, \delta_2, w)$ such that all cells in g_i simulate the ith cell C_i in real-time and N can fire any $m \times n$ arrays at time $t = T(m+n-1)$ if and only if M fires 1-D arrays of length $m+n-1$ at time $t = T(m+n-1)$, where $\delta_2 : Q^5 \rightarrow Q$ is the transition function, and $w \in Q$ is the border state of the array. Note that the set of internal states of N is the same as M. For the details of the construction of the transition rule set, see Umeo, Maeda and Fujiwara [13].

Now let M have $m+n-1$ cells. Here we show that the construction of 2-D CA N can generate the configuration of M in real-time. Specifically, for any i, $1 \leq i \leq m+n-1$, the state of any cell in g_i at any step is the same and is identical to the state of C_i at the corresponding step. Let S_i^t, $S_{i,j}^t$ and $S_{g_i}^t$ denote the state of C_i, $\mathrm{C}_{i,j}$ at step t and the set of states of the cells in g_i at step t, respectively. Then, we can establish the following lemma.

[Lemma 4] Let i and t be any integers such that $1 \leq i \leq m+n-1$, $0 \leq t \leq T(m+n-1)$. Then, $S_{g_i}^t = \{S_i^t\}$.

Umeo, Maeda and Fujiwara [2002] presented a 6-state two-dimensional synchronization algorithm that fires any $m \times n$ arrays in $2(m+n)-4$ steps. An $m \times n$ 2-D array synchronization problem is reduced to one 1-D synchronization problem with the general at the left end. The algorithm is slightly slower than the optimum ones, but the number of internal states is considerably smaller.

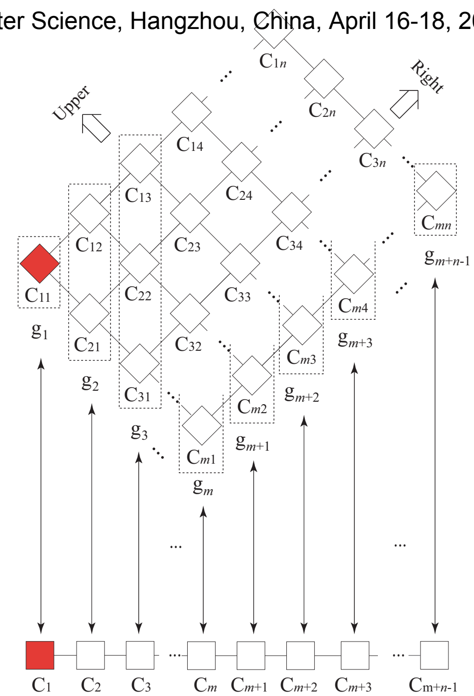

Figure 4: A correspondence between 1-D and 2-D arrays.

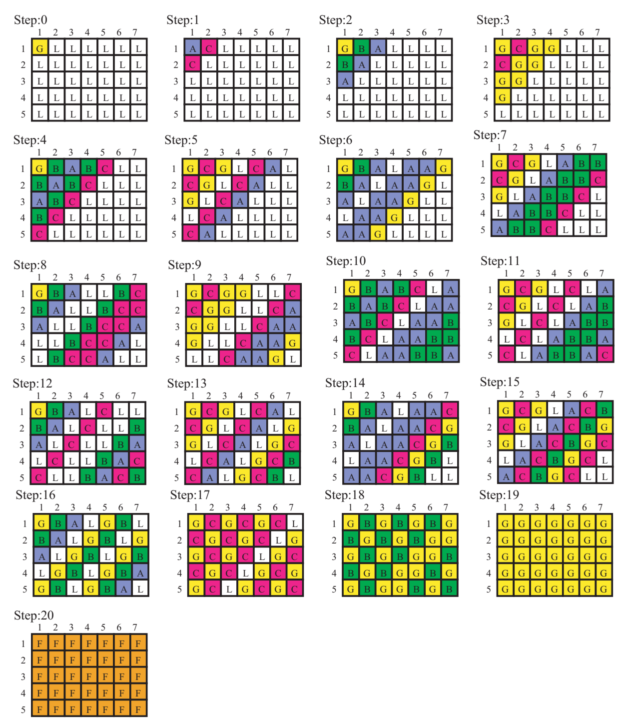

Figure 5: Snapshots of the proposed 6-state linear-time firing squad synchronization algorithm on rectangular arrays.

[Theorem 5][Umeoetal.[2006]] Let A be any s-state firing synchronization algorithm operating in $T(\ell)$ steps on 1-D ℓ cells. Then, there exists a 2-D s-state cellular automaton that can synchronize any $m \times n$ rectangular array in $T(m+n-1)$ steps.

[Theorem 6][Umeoetal.[2006]] There exists a 6-state 2-D CA that can synchronize any $m \times n$ rectangular array in $2(m+n)-4$ steps.

[Theorem 7][Umeoetal.[2006]] There exists a 6-state 2-D CA that can synchronize any $m \times n$ rectangular array containing isolated rectangular holes in $2(m+n)-4$ steps.

[Theorem 8][Umeoetal.[2006]] There exists a 6-state firing squad synchronization algorithm that can synchronize any 3-D $m \times n \times \ell$ solid arrays in $2(m+n+\ell)-6$ steps.

[Theorem 9] There exists a 14-state 2-D CA that can synchronize any $m \times n$ rectangular array in $m+n+\max(r+s, m+n-r-s+2)-4$ steps with the general at an arbitrary initial position (r,s).

Szwerinski [12] also proposed an optimum-time generalized 2-D firing algorithm with 25,600 internal states that fires any $m \times n$ array in $m+n+\max(m,n)-\min(r, m-r+1)-\min(s, n-s+1)-1$ steps, where (r,s) is the general's initial position. Our 2-D generalized synchronization algorithm is $\max(r+s, m+n-r-s+2)-\max(m,n)+\min(r, m-r+1)+\min(s, n-s+1)-3$ steps larger than the optimum algorithm proposed by Szwerinski [12]. However, the number of internal states required to yield the firing condition is the smallest known at present. Snapshots of our 14-state generalized synchronization algorithm running on a rectangular array of size 6×8 with the general at $C_{2,3}$ are shown in Fig. 6.

6 Diagonal Mapping II: Twelve-state Time-optimum Algorithm $\mathcal{A}_4$

An $m \times n$ 2-D array synchronization problem is reduced to one 1-D generalized synchronization problem: $\mathcal{P}(m, m+n-1)$. The proposal is a simple and efficient mapping scheme that enables us to embed a special class of one-dimensional generalized synchronization algorithm onto two-dimensional arrays without introducing additional states.

We divide mn cells into $m+n-1$ groups g_k defined as follows, where k is any integer such that $-(m-1) \leq k \leq n-1$.

$$g_k = \{C_{i,j} | j-i=k\},\ -(m-1) \leq k \leq n-1$$

Figure 7 shows the correspondence between 1-D and 2-D arrays.

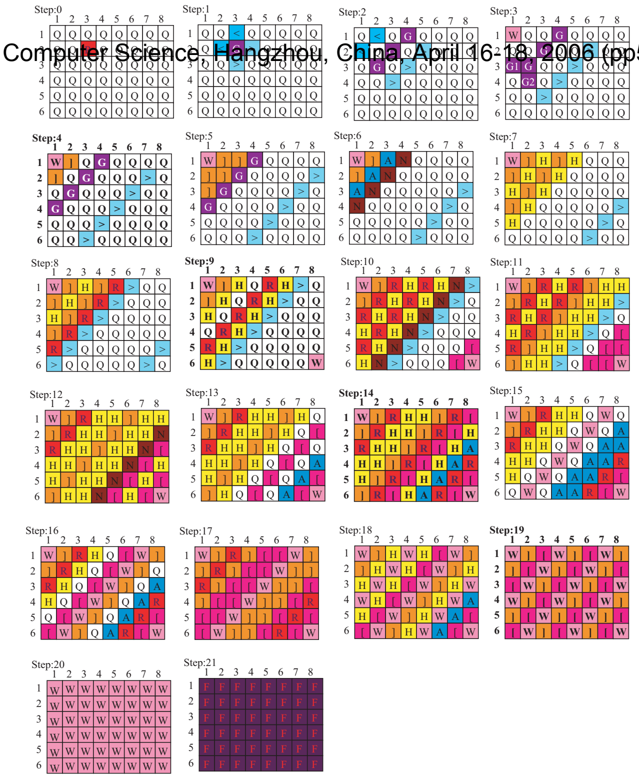

Figure 6: Snapshots of our 14-state linear-time generalized firing squad synchronization algorithm on rectangular arrays.

Property $\mathcal{A}$: Let S_i^t denote the state of C_i at step t. We say that a generalized firing algorithm has a *property $\mathcal{A}$*, where any state S_i^t appearing *in the area* A(See Fig. 8) can be computed from its left and right neighbor states S_{i-1}^{t-1} and S_{i+1}^{t-1} but it never depends on its own previous state S_i^{t-1}.

The one-dimensional generalized firing squad synchronization algorithm with the property $\mathcal{A}$ can be easily embedded onto two-dimensional arrays without introducing any additional states.

Let S_i^t and $S_{k,\ell}^t$ denote the state of C_i and $C_{k,\ell}$ at step t, respectively, where $-(m-1) \leq i \leq n-1$, $1 \leq k \leq m$, $1 \leq \ell \leq n$. We define the following set of cells. Let $\mathcal{S}$, $\mathcal{S}_t$, $\overline{\mathcal{S}_t}$ be set of cells such that:

$\mathcal{S} = \{C_{k,\ell} | 1 \leq k \leq m, 1 \leq \ell \leq n\}$,

$\mathcal{S}_t = \{C_{k,\ell} | 2 \leq k+\ell \leq t+2, 1 \leq k \leq m, 1 \leq \ell \leq n\}$,

$\overline{\mathcal{S}_t} = \mathcal{S} - \mathcal{S}_t$, where $0 \leq t \leq T(m, m+n-1)$.

Let $S_{g_i}^t$ denote a set of states of the cells in $g_i \cap \mathcal{S}_t$ at step t and $\overline{S}_{g_i}^t$ be a set of states of the cells in $g_i \cap \overline{\mathcal{S}_t}$ at step t, where $0 \leq t \leq T(m, m+n-1)$ and $-(m-1) \leq i \leq n-1$.

[Lemma 10] Let i and t be any integers such that $-(m-1) \leq i \leq n-1$, $0 \leq t \leq T(m, m+n-1)$.

1. For any t such that $0 \leq t \leq m+n-1$, $\| S_{g_i}^t \| = \| \overline{S}_{g_i}^t \| = 1$. That is, the set $S_{g_i}^t$ and $\overline{S}_{g_i}^t$ are singletons

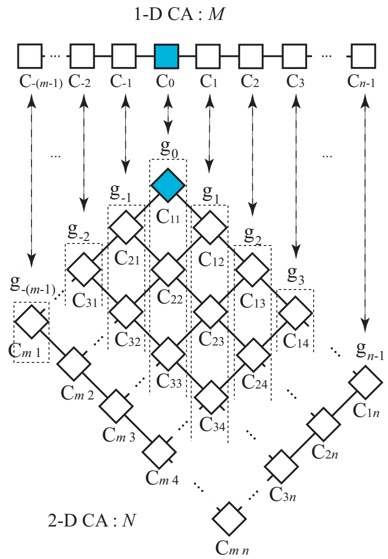

Figure 7:
Correspondence between 1-D and 2-D cellular arrays.

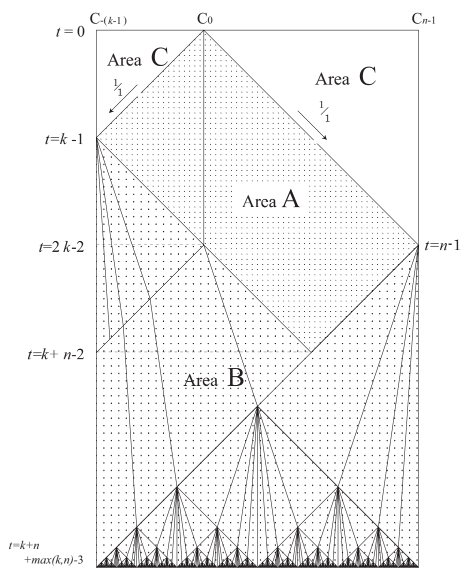

Figure 8: Time-space diagram for generalized optimum-step firing squad synchronization algorithm.

and all cells in $g_i \cap \mathcal{S}_t$ and $g_i \cap \overline{\mathcal{S}_t}$ at step t, respectively are in the same state. We denote these states as $S^t_{g_i}$ and $\overline{S}^t_{g_i}$, respectively.

2. Then, $S^t_{g_i} = S^t_i$ and $\overline{S}^t_{g_i}$ =q, where q is the quiescent state of N.

3. For any t such that $m + n \le t \le T(m, m + n - 1)$, $\| S^t_{g_i} \| = 1, \| \overline{S}^t_{g_i} \| = 0$ and $S^t_{g_i} = S^t_i$.

[Theorem 11] Let M be any s-state generalized synchronization algorithm with the property $\mathcal{A}$ operating in $T(k, \ell)$ steps on 1-D ℓ cells with a general on the k-th cell from the left end. Then, based on M, we can construct a 2-D s-state cellular automaton that can synchronize any $m \times n$ rectangular array in $T(m, m + n - 1)$ steps.

Moore and Langdon [1968], Szwerinski [1982] and Varshavsky, Marakhovsky and Peschansky [1970] developed a generalized optimum-time firing algorithm with 17, 10 and 10 internal states, respectively, that fires 1-D n cells in $n - 2+\max(k, n - k + 1)$ steps, where the general is located on C_k. Recently, Settle and Simon [2002] and Umeo, Hisaoka, Michisaka, Nishioka and Maeda [2002] have proposed a new 9-state generalized synchronization algorithm operating in optimum-step. The next theorem is a 12-state implementation of the generalized optimum-time synchronization algorithms having the property $\mathcal{A}$. Figure 9 shows the Snapshots for a 12-state implementation of generalized firing squad synchronization algorithm with the property $\mathcal{A}$ on 15 cells with a general on C_7.

[Theorem 12] There exists a 12-state 1-D cellular automaton with the property $\mathcal{A}$ that can synchronize ℓ cells with a general on the k-th cell from the left end in optimum $\ell - 2 + max(k, \ell - k + 1)$ steps.

Based on [Theorems 11, 12] we can get a 12-state optimum-time synchronization algorithm for rectangular arrays.

[Theorem 13] There exists a 12-state firing squad synchronization algorithm that can synchronize any $m \times n$ rectangular array in optimum $m + n + max(m, n) - 3$ steps.

7 Conclusions

We have given a survey on recent developments of optimum-time algorithms that can synchronize any $m \times n$ two-dimensional rectangular arrays in $m + n + max(m, n) - 3$ steps. Those algorithms are based on an efficient mapping schemes for embedding a special class of generalized one-dimensional optimum-time firing squad synchronization algorithms onto 2-D rectangular arrays. We progressively reduce the number of internal states of each cellular automaton operating in

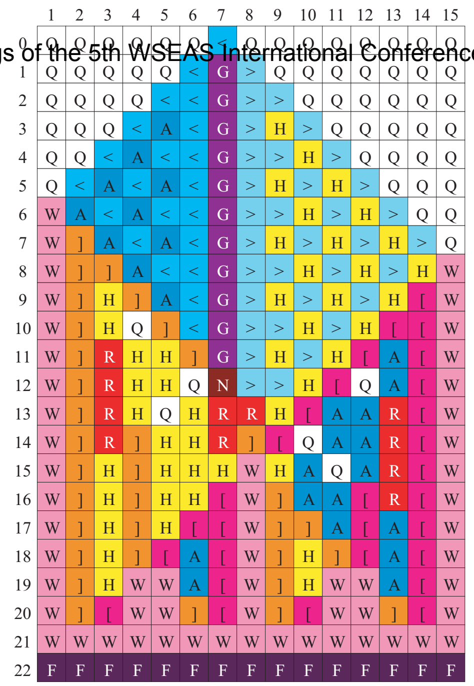

Figure 9: Snapshots for a 12-state implementation of generalized firing squad synchronization algorithm with the property $\mathcal{A}$ on 15 cells with a general on C_7.

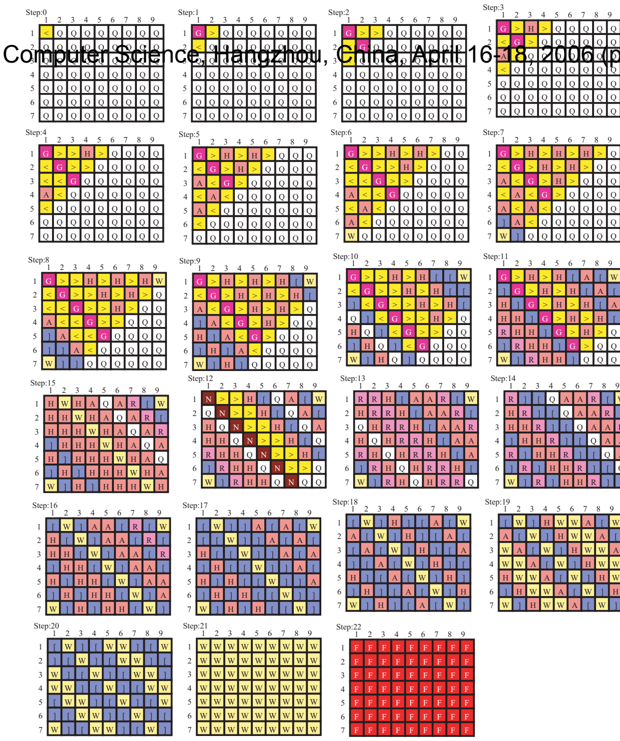

Figure 10: Snapshots of the proposed 12-state optimum-time firing squad synchronization algorithm on rectangular arrays.

optimum-step on rectangular arrays, achieving six and twelve states. These are the smallest number of states reported to date for synchronizing rectangular arrays in linear- and optimum-step, respectively.

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] W. T. Beyer: Recognition of topological invariants by iterative arrays. Ph.D. Thesis, MIT, (1969), pp. 144.

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

[4] A. Grasselli: Synchronization of cellular arrays: The firing squad problem in two dimensions. *Information and Control*, vol. 28(1975), pp. 113-124.

[5] K. Kobayashi: The firing squad synchronization problem for two-dimensional arrays. *Information and Control*, vol. 34(1977), pp. 177-197.

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

[7] 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.

[8] F. R. Moore and G. G. Langdon: A generalized firing squad problem. *Information and Control*, vol. 12(1968), pp. 212-220.

[9] H. B. Nguyen and V. C. Hamacher: Pattern synchronization in two-dimensional cellular space. *Information and Control*, vol. 26(1974), pp. 12-23.

[10] I. Shinahr: Two- and three-dimensional firing squad synchronization problems. *Information and Control*, vol. 24(1974), pp. 163-180.

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

[12] H. Szwerinski: Time-optimum solution of the firing-squad-synchronization-problem for n-dimensional rectangles with the general at an arbitrary position. *Theoretical Computer Science*, vol. 19(1982), pp. 305-320.

[13] H. Umeo, M. Maeda and N. Fujiwara: An efficient mapping scheme for embedding any one-dimensional firing squad synchronization algorithm onto two-dimensional arrays. *Proc. of the 5th International Conference on Cellular Automata for Research and Industry*, LNCS 2493, Springer-Verlag, pp.69-81(2002).

[14] H. Umeo: A simple design of time-efficient firing squad synchronization algorithms with fault-tolerance. *IEICE Trans. on Information and Systems*, Vol. E87-D, No.3, 2004, pp.733-739.

[15] H. Umeo, T. Sogabe and Y. Nomura: Correction, optimization and verification of transition rule set for Waksman's firing squad synchronization algorithms. *Proc. of the 4th International Conference on Cellular Automata for Research and Industry*, 2000, pp. 152-160.

[16] H. Umeo, M. Hisaoka and T. Sogabe: A survey on optimum-time firing squad synchronization algorithms for one-dimensional cellular automata. *Intern. J. of Unconventional Computing*, 2005, vol. 1, pp.403-426.

[17] H. Umeo, M. Hisaoka, M. Teraoka and M. Maeda: Several new generalized linear- and optimum-time synchronization algorithms for two-dimensional rectangular arrays. *Proc. of 4th International Conference on Machines, Computations and Universality : MCU 2004, LNCS 3354 (M. Margenstern(Ed.)*, (2005), pp.223-232.

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