A State-Efficient Implementation of Synchronization Algorithms for Two-Dimensional Cellular Automata
-Extended Abstract-

Masato Teraoka†, Masaya Hisaoka†, Masashi Maeda† and Hiroshi Umeo†

†Univ. of Osaka Electro-Communication
Neyagawa-shi, Hatsu-cho 18-8
Osaka, 572-8530 Japan

1 Introduction

The synchronization in cellular automata has been known as 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 [5]. The firing squad synchronization problem has been studied extensively for more than 40 years [1-10]. The present authors are involved in research on firing squad synchronization algorithms on two-dimensional (2-D) cellular arrays.

In this paper, we first propose a new linear-time generalized synchronization algorithm 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) of the array. The algorithm is based on a state-efficient mapping scheme for embedding a restricted class of generalized one-dimensional optimum-time synchronization algorithms onto 2-D rectangular arrays. The embedding can be implemented with providing two additional states. We show that the linear-time 14-state solution developed yields an optimum-time synchronization algorithm in the case where the general is located at the north-east corner. Due to the space available, we omit proofs of the theorems.

2 Firing Squad Synchronization Problem on Two-Dimensional Cellular Automata

Figure 1 shows a finite two-dimensional (2-D) 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 one *general* cell, 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 $t = 0$, any one cell on the array is in the *fire-when-ready* state, which is the initiation signal for synchronizing 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 m and n. We call the synchronization problem *normal*, when the initial position of the general is restricted to north-west corner of the array. We consider a *generalized* firing squad synchronization problem, in which the general can be initially located at any position on the array. As for the normal synchronization problem, several algorithms have been proposed, including Beyer [1], Grasselli [2], Kobayashi [3], Shinahr [7], Szwerinski [8] and Umeo, Maeda and Fujiwara [9]. Umeo, Maeda and Fujiwara [9] presented a 6-state two-dimensional synchronization algorithm that fires

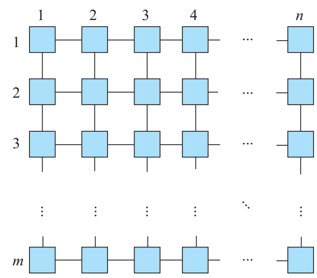


Figure 1: A two-dimensional cellular automaton.

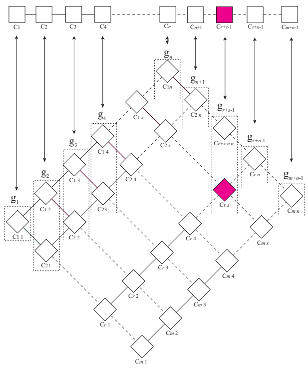


Figure 2: Correspondence between 1-D and 2-D cellular arrays.

any $m \times n$ arrays in $2(m+n)-4$ steps. The algorithm is slightly slower than the optimum ones, but the number of internal states is considerably smaller. Beyer [1] and Shinahr [7] presented an optimum-time synchronization scheme in order to synchronize any $m \times n$ arrays in $m+n+max(m,n)-3$ steps. To date, the smallest number of cell states for which an optimum-time synchronization algorithm has been developed is 28 for rectangular array, achieved by Shinahr [7]. On the other hand, Szwerinski [8] proposed an optimum-time generalized 2-D firing algorithm with 25,600 internal states.

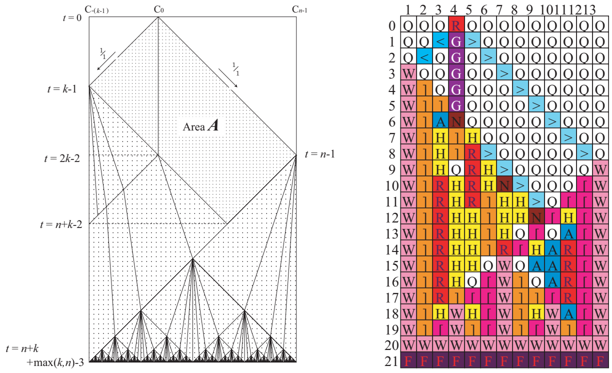
3 Linear- and Optimum-Time Firing Squad Synchronization Algorithms

Now we consider a generalized firing squad synchronization problem, in which the general can be initially located at any position on the array. Before presenting the algorithm, we propose a simple mapping scheme for embedding one-dimensional generalized synchronization algorithms onto two-dimensional arrays.

Consider a correspondence between 1-D array of length $m+n-1$ and 2-D array of size $m \times n$, shown in Fig. 2. Each black square corresponds to initial *general* cell on the array. The 2-D array of size $m \times n$ is divided into $m+n-1$ groups g_k, $1 \leq k \leq m+n-1$, that is defined as follows:

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

$g_1 = \{\mathrm{C}_{1,1}\}$,
$g_2 = \{\mathrm{C}_{1,2}, \mathrm{C}_{2,1}\}$,
$g_3 = \{\mathrm{C}_{1,3}, \mathrm{C}_{2,2}, \mathrm{C}_{3,1}\}$,
.



(a) (b)

Figure 3: Time-space diagram for optimum-time generalized firing squad synchronization algorithm and snapshots for a 12-state implementation of the generalized firing squad synchronization algorithm with the property $\mathcal{Q}$ on 13 cells with a general on C_4.

.
.
,
$g_{m+n-1} = \{\mathrm{C}_{m,n}\}$.

The objective of our correspondence is to embed configurations of 1-D generalized synchronization algorithms onto 2-D arrays.

Property $\mathcal{Q}$: We say that a generalized firing algorithm has a *property* $\mathcal{Q}$, where any cell, except the general cell C_k, keeps a quiescent state *in the area* A of the time-space diagram shown in Fig. 3(a).

For any 2-D array M of size $m \times n$ with the general at $\mathrm{C}_{r,s}$, where $1 \leq r \leq m$, $1 \leq s \leq n$, there exists a corresponding 1-D cellular array N of length $m+n-1$ with the general at C_{r+s-1} such that the configuration of N can be mapped on M, and M fires if and only if N fires. The transition table for N consists of four parts, one is a transformation rule set (Type (I)) that is for the inner cells of 2-D array and the other two sets (Type (II) and Type (III)) are for the state transition of cells $\mathrm{C}_{1,1}$ and $\mathrm{C}_{m,n}$. The fourth part is a new set of transition rules (omitted) for the transmission of *general* state in the diagonal direction. Let $\delta_1(a,b,c) = d$ be any transition rule of M, where $a,b,c,d \in \{Q - \{w\}\}$. Then, N has seven Type (I) transition rules, as shown in Fig. 4. The first rule (1) in Type (I) is used by an inner cell that does not include border cells amongst its four neighbors. Rules

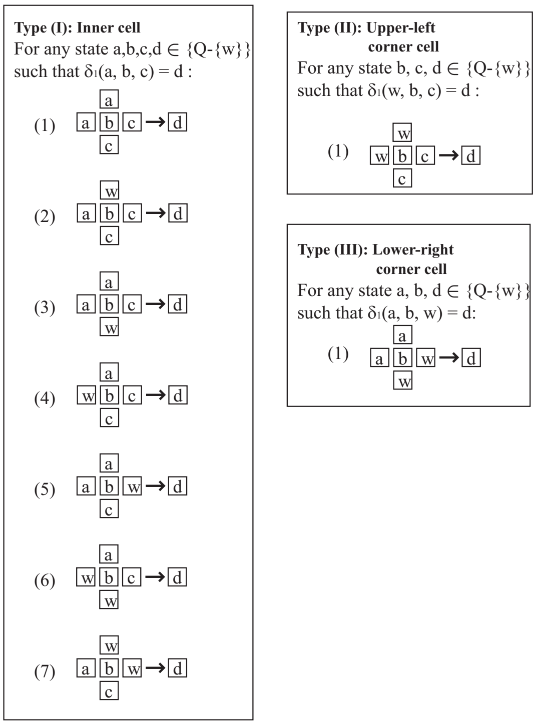


Figure 4: Construction of transition rules for 2-D linear-time firing squad synchronization algorithm.

(2)-(5) are used by an inner cell that has a border cell as its upper, lower, left, right, lower left, or upper right neighbor, respectively. Here the terms *upper*, *right* etc. on the rectangular array are interpreted in a usual way, shown in Fig. 2, although the array is rotated by 45° in the counterclockwise direction. Rules (6)-(7) in Type (I) are used by an inner cell that has border cells in its left-lower and right-upper neighbors, respectively.

Let S_i^t, $S_{i,j}^t$ and $S_{g_i}^t$ denote the state of C_i, $C_{i,j}$ 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 1] The following two statements hold:

1. For any integer i and t such that $1 \leq i \leq m+n-r-s+1$, $r+s+i-3 \leq t \leq T(m+n-1, r+s-1)$, $\parallel S_{g_i}^t \parallel = 1$ and $S_{g_i}^t = S_i^t$. That is, all cells in g_i at step t are in the same state and it is equal to S_i^t, where the state in $S_{g_i}^t$ is simply denoted by $S_{g_i}^t$.

2. For any integer i and t such that $m+n-r-s+2 \leq i \leq m+n-1$, $2m+2n-r-s-i-1 \leq t \leq T(m+n-1, r+s-1)$, $\parallel S_{g_i}^t \parallel = 1$ and $S_{g_i}^t = S_i^t$.

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

The one-dimensional generalized firing squad synchronization algorithm with the *property* $\mathcal{Q}$ can be easily embedded onto two-dimensional arrays with a small overhead. Fig. 3(b) shows snapshots of our 12-state optimum-time generalized firing squad synchronization algorithm with the *property* $\mathcal{Q}$.

[Theorem 3] There exists a 12-state one-dimensional cellular automaton with the property $\mathcal{Q}$ 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 the 12-state generalized 1-D algorithm given above, we obtain the following 2-D generalized synchronization algorithm that synchronizes any 2-D array of size $m \times n$ in $m+n-3+\max(r+s-1, m+n-r-s+1) = m+n+\max(r+s, m+n-r-s+2)-4$ steps.

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

Two additional states are required in our construction (details omitted). Szwerinski [8] 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 [8]. 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 7×9 with the general at $C_{4,5}$ are shown in Fig. 5.

Our linear-time synchronization algorithm is interesting in that it includes an optimum-step synchronization algorithm as a special case where the general is located at the north-east corner. By letting $r = 1$, $s = n$, we get $m+n+\max(r+s, m+n-r-s+2)-4 = m+n+\max(n+1, m+1)-4 = m+n+\max(m,n)-3$. Thus the algorithm is a time-optimum one. We have:

[Theorem 5] There exists a 14-state 2-D CA that can synchronize any $m \times n$ rectangular array in $m + n + \max(m, n) - 3$ steps.

4 Conclusions

We have proposed a state-efficient mapping scheme for embedding a restricted class of generalized one-dimensional synchronization algorithms onto 2-D rectangular arrays, then based on the scheme, a new linear-time generalized synchronization algorithm with fourteen states has been presented 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) of the array. It is shown that the linear-time 14-state solution developed yields an optimum-time synchronization algorithm in the case where the general is located at the north-east corner.

References

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

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

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

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

[5] 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.

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

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

[8] 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.

[9] 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).

[10] H. Umeo, M. Hisaoka, K. Michisaka, K. Nishioka and M. Maeda: Some generalized synchronization algorithms and their implementations for a large scale cellular automata. *Proc. of the Third International Conference on Unconventional Models of Computation, UMC 2002*, LNCS 2509, Springer-Verlag, pp.276-286(2002).

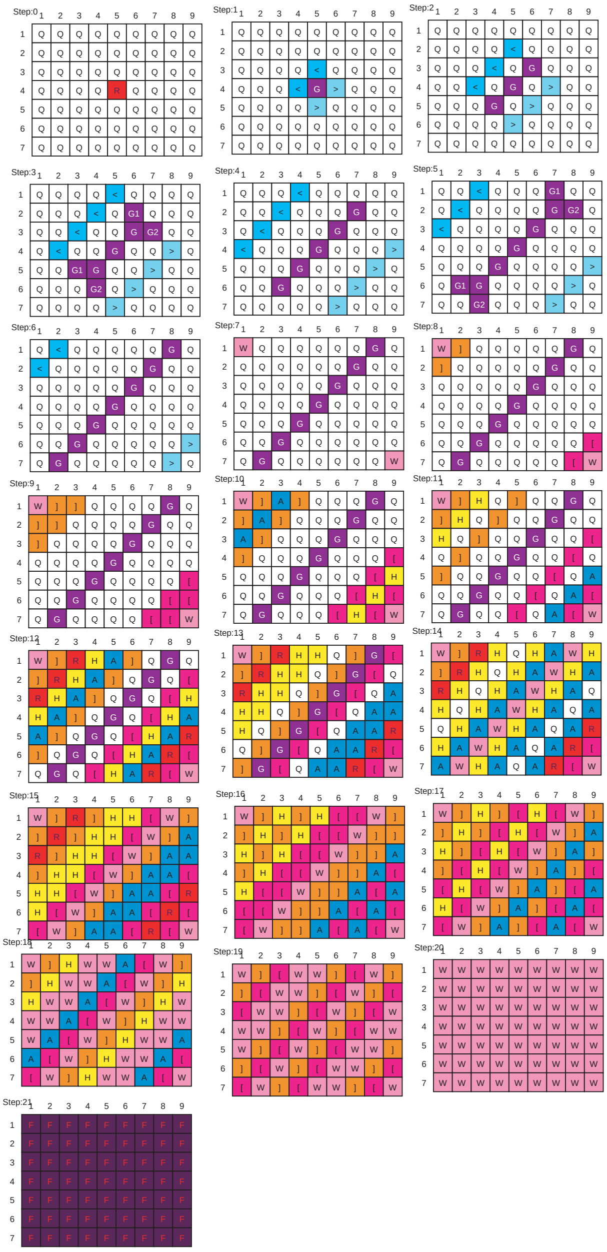


Figure 5: Snapshots of the 14-state linear-time generalized firing squad synchronization algorithm on rectangular arrays.