

## B105 2次元セルラオートマトン上での同期アルゴリズム

## A Design of Two-Dimensional Firing Squad Synchronization Algorithm and Its Implementation

前田 雅史<sup>†</sup>      藤原 法生<sup>†</sup>      梅尾 博司<sup>\*\*</sup>  
 Masashi Maeda      Norio Fujiwara      Hiroshi Umeo

<sup>†</sup>大阪電気通信大学大学院 工学研究科 情報工学専攻  
 〒572-0833 大阪府寝屋川市初町 18 番 8 号  
 Osaka Electro-Communication Univ., Graduate School of Engineering

<sup>\*\*</sup>大阪電気通信大学大学院 工学研究科 情報工学専攻,  
 大阪電気通信大学 総合情報学部 情報工学科  
 〒572-0833 大阪府寝屋川市初町 18 番 8 号  
 Osaka Electro-Communication Univ., Graduate School of Engineering,  
 Osaka Electro-Communication Univ., Faculty of Information Science and Technology

**Abstract :** The famous firing squad synchronization problem is stated as follows: Given a one-dimensional array of  $n$  identical cellular automata, including a "general" at the left end which is activated at time  $t = 0$ , we want to design the automata so that, at some future time, all the cells will *simultaneously* and, *for the first time*, enter a special "firing" state. The problem was originally proposed by J. Myhill in 1957, presented in Moore[5], to synchronize all parts of a self-reproducing machine. A lot of literatures have been published on this topic[1-17].

In this paper we propose a simple and efficient embedding of one-dimensional firing squad synchronization algorithms onto multi-dimensional arrays and give some implementations on two-dimensional arrays. Several 2-D synchronization algorithms have been presented in Shinar[6], Grasselli[7], and Szwerinski[9], however, practical implementations are rarely found. We give the following implementations together with firing configurations on computer simulation. Most of them have relatively small number of internal states.

**[Theorem 1]** There exists a 9-state 2-D CA which can synchronize  $n \times n$  square arrays in optimum  $2n - 2$  steps.

**[Theorem 2]** There exists a 6-state 2-D CA which can synchronize  $m \times n$  rectangular arrays in  $2(m + n) - 4$  steps.

**[Theorem 3]** There exists a 6-state 2-D CA that can synchronize  $m \times n$  cellular arrays with some isolated rectangular holes in  $2(m + n) - 4$  steps.

**[Theorem 4]** There exists a 21-state 2-D CA which can synchronize  $m \times n$  rectangular arrays in  $2(m + n) - \min((k + l - 2), (m + n - k - l)) - 4$  steps. The general is located on  $C_{kl}$ .

## 1. はじめに

1957 年 Myhill により一斉射撃問題 (Firing Squad Synchronization Problem, FSSP と略す) が提唱され, Minsky [1] により世界で最初の  $3n + O(\log n)$  ステップ ( $n$  はセル数を意味する) 一斉射撃アルゴリズムが提案されて以来, FSSP に関する研究はこれまでに数多くなされている。代表的な一斉射撃アルゴリズムとして, 2次元セルラオートマトン(2-Dimensional Cellular Automata, 2次元 CA と略す) 上では, Shinahr [6], Grasselli [7], Szwerinski [9] のアルゴリズムが知られている。しかし, これらのアルゴリズムは状態数が非常に多いという欠点がある。また, 故障セルの存在する 2次元 CA に対応した耐故障性一斉射撃アルゴリズムの存在も知られていない。

本稿では,  $n \times n$  個のセルからなる正方形アレイ上で  $2n - 2$  ステップで動作する 9 状態最適時間一斉射撃アルゴリズム,  $m \times n$  個のセルからなる長方形アレイ上で  $2(m + n) - 4$  ステップで動作する 6 状態の耐故障性一斉射撃アルゴリズム, さらに,  $m \times n$  個のセルからなる長方形アレイ上で初期将軍を任意とした場合,  $2(m + n) - 4 - \min(k + l - 2, m - k + n - l)$  ステップで動作する 21 状態の一般化一斉射撃アルゴリズムを提案する。

## 2. 2次元 CA

有限個の有限オートマトンを 2次元整数格子点上に配置したアレイを考える。Fig. 1 参照。オートマトンは, 境界を除き全て同

一構造を持つ。各オートマトンはセルと呼ばれる。全てのセルは一斉射撃して動作し, 各セルの離散時刻  $t+1$  における状態は, 時刻  $t(t \geq 0)$  における自身の状態と上下左右に隣接する 4 つのセルの状態により決定される。この動作の行えないセルを故障セルという。境界上のセルは自身が端のセルであることを認識している。Fig. 1 において  $m$  と  $n$  が等しい場合に限定した 2次元 CA を正方形 CA,  $m$  と  $n$  が等しくない場合も含む 2次元 CA を長方形 CA と呼ぶ。  $i$  行  $j$  列に位置するセルを  $C_{ij}$  であらわす。

次に説明する一斉射撃問題を考える時は, 時刻  $t=0$  における初期計算状況において, 左上端のセルを将軍, それ以外のセルを兵士と呼ぶ。

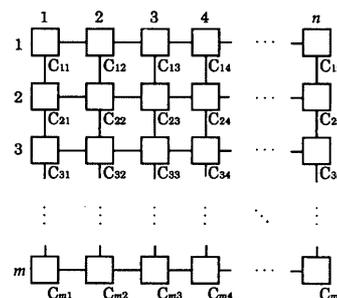


Fig. 1 A two-dimensional cellular automaton

### 3. 一斉射撃問題

一斉射撃問題とは時刻  $t=0$  に将軍から発せられた、「準備ができたら一斉に射撃せよ」という命令により、未来の時刻  $t=\alpha$  において一斉に特殊な状態(射撃状態)に遷移するように遷移規則を決定する問題である。局所的な通信しか持たないモデルにおいてグローバルな制御を行うことが多くの研究者の興味を引き、広く研究されている。

### 4. 正方形 CA 上での最適時間一斉射撃アルゴリズム

1次元 CA 上での最適時間一斉射撃アルゴリズムをベースに、 $n \times n$  のセルからなる正方形 CA 上で最適時間一斉射撃アルゴリズムを設計するメカニズムを Fig. 2 に示す。時刻  $t=0$  において  $C_{11}$  の将軍状態のセルは、右方向と下方向に隣接するセルに新しい将軍を生成するための情報を伝達する。この情報により  $t=2$  に  $C_{22}$  に新しい将軍を生成する。同様に生成された将軍は右方向と下方向に隣接するセルに新しい将軍を生成するための情報を伝達し再帰的に、次々と将軍を生成する。 $C_{ii}$  ( $1 \leq i \leq n$ ) が将軍となる時刻は  $t=2i-2$  である。

$C_{ii}$  は  $t=2i-2$  より  $C_{ii}$  から  $C_{in}$  までを縦方向のグループ、 $C_{ii}$  から  $C_{in}$  までを横方向のグループとし、各グループが1次元 CA 上での一斉射撃アルゴリズムの動作を実時間で模倣するように遷移規則を決定する。それぞれのグループが一斉射撃に要する時間は  $2(n-i+1)-2$  ステップである。 $C_{ii}$  に将軍が生成される時刻は  $t=2i-2$  なので、 $C_{ii}$  を含む縦方向、横方向のグループが一斉射撃する時刻は  $t=2i-2+2(n-i+1)=2n-2$  である。よって全てのセルが  $t=2n-2$  において一斉射撃を実現できる。

この設計方法では各セルは自信が縦方向のグループのセルか横方向のグループのセルかの判断が必要となる。上記の判断を行うために縦方向のグループのセルと横方向のセルとで別の状態を用いると、状態数はベースとする1次元 CA 上での一斉射撃アルゴリズムの2倍となる。また、 $C_{ii}$  が縦方向と横方向両方のグループで共有されるので、そのことを認識できるよう遷移規則を決定する。

上記の方法に基づき、ベースとなる1次元 CA 上での最適時間一斉射撃アルゴリズムとして Mazoyer [4]を用い、正方形 CA 上での一斉射撃アルゴリズムの設計を行った。今回、縦方向のグループと横方向のグループの状態を横方向は状態名の後ろに1、縦方向のグループに用いる状態に状態名の後ろに2をつけた状態を用いることとした。Mazoyer [4]の6状態を全て縦方向のグループのセルと横方向のグループのセルとで別の状態を用いると、状態数は  $6 \times 2 = 12$  となるが、縦方向のグループのセルと横方向のグループのセルとで別の状態を用いる必要がない状態が3種類存在する。

射撃状態である  $F$  が1つの状態でありことは自明である。次に、静止状態である  $L$  を1つの状態とする。 $L$  は隣接するセルの状態から自身の動作を判断できる。これら2つを1つの状態として作成した2次元 CA 上での10状態一斉射撃アルゴリズムを  $H=(Q, \delta, w)$  とする。ここで、 $Q$  は内部状態集合、 $w(\in Q)$  を左右の端をあらわす境界記号とする。

縦方向のグループと横方向のグループで分けた状態を  $A_1, A_2, B_1, B_2, \dots$  とする。

$$\delta(A_1, A_1, L, w, w) = F \dots (1)$$

$$\delta(A_2, A_2, L, w, w) = F \dots (2)$$

この遷移規則(1)と(2)は  $A$  を縦方向のグループと横方向のグループで異なる状態を用いなくとも次の遷移規則で同じ動作が可能である。

$$\delta(A, A, L, w, w) = F \dots (3)$$

よって、この遷移規則は状態を分ける必要がない。

しかし、次の遷移規則は状態を分けなければ判断できない。

$$\delta(B_1, L, B_1, B_2, L) = B_1 \dots (4)$$

$$\delta(B_2, L, B_2, B_1, L) = B_2 \dots (5)$$

この遷移規則(4)と(5)は  $B_1, B_2$  を  $B$  としてしまうと次に遷移すべき状態の判断が不可能なので、 $B$  は縦方向のグループと横方向のグループで状態を必要がある。このような縦方向と横方向で分ける必要のない状態、分ける必要のある状態を、 $H$  の全ての遷移規則についてコンピュータにより分類した。そして、全ての遷移規則において状態  $A$  は縦方向と横方向で異なる状態を用いる必要がないと結果を得た。これより状態  $A$  を1つの状態とする9状態のアルゴリズムを設計した。

設計するにあたり、 $C_{ii}$  は縦方向と横方向両方のグループで共有されていることを認識するように状態を用いなければならないが、本稿では横方向のグループの状態を用いた。この場合  $C_{ii}$  は上と右に隣接するセルが横方向のグループの状態、左と下に隣接するセルが縦方向のグループの状態となり、 $C_{ii}$  は縦方向と横方向両方のグループで共有されていることを認識できる。よって  $L, A, F$  を1つの状態とし、状態数が  $6 \times 2 - 3 = 9$  の一斉射撃アルゴリズムを設計した。このアルゴリズムのシミュレータへの実装結果を Fig. 3 に示す。以上の結果を次の定理にまとめる。

[定理 1]  $n \times n$  個のセルからなる2次元セル空間を  $2n-2$  ステップで一斉に同じ状態にする9状態一斉射撃アルゴリズムが存在する。

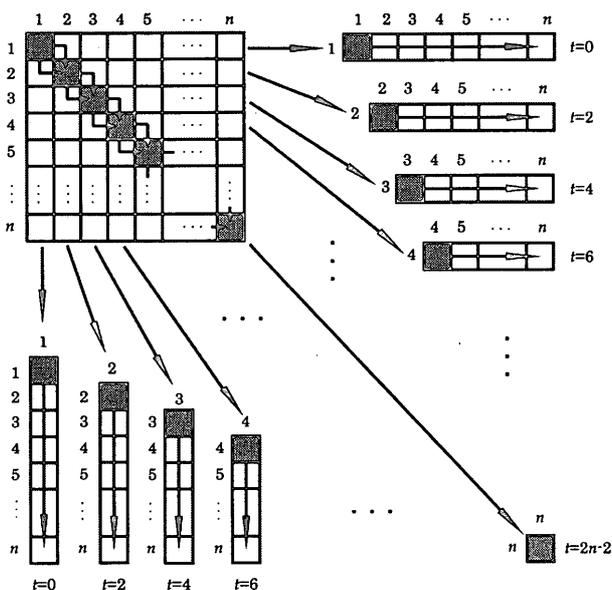


Fig. 2 A decomposition of 2-dimensional cellular space into vertical and horizontal 1-dimensional arrays, where Mazoyer's 6-state optimum-time firing squad synchronization algorithm operates on each array

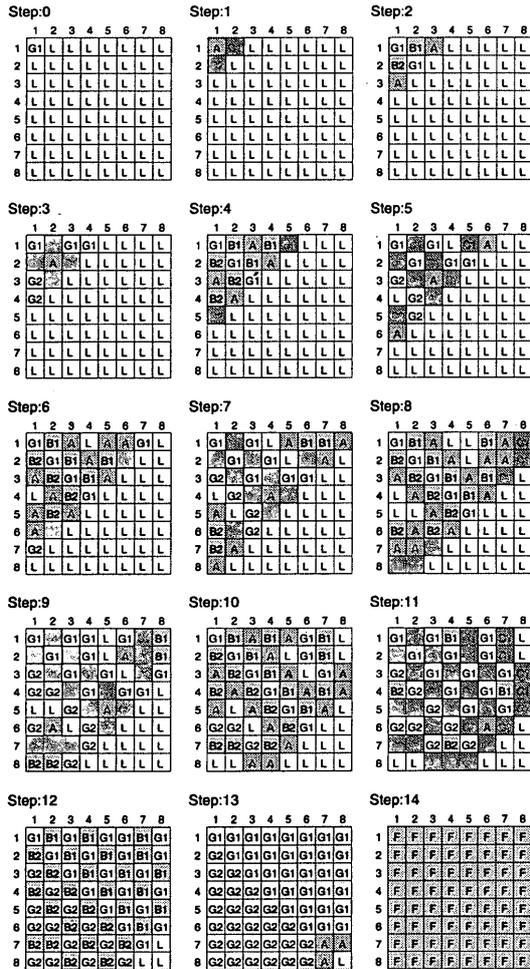


Fig. 3 Snapshots of our 9-state optimum-time firing squad synchronization algorithm

5. 長方形CA上での $2(m+n)-4$ ステップ同期アルゴリズム

長さ $n$ の1次元セル空間を $T(n)$ ステップで一斉射撃する1次元CAを $M=(Q, \delta, w)$ とする。ここで、 $Q$ は $M$ の内部状態集合、 $\delta$ は $Q^3 \rightarrow Q$ なる遷移関数、 $w \in Q$ は1次元セル空間における左端および右端の境界記号である。 $M$ と同じ内部状態集合を持ち、サイズ $m \times n$ のセル空間を $T(m+n-1)$ ステップで一斉射撃する2次元CA、 $N=(Q, \hat{\delta}, w)$ を以下のように構成する。ここで、 $Q$ は $N$ の内部状態集合、 $\hat{\delta}$ は $Q^5 \rightarrow Q$ なる遷移関数、 $w \in Q$ は2次元セル空間における上側、下側、左端、および右端の境界記号である。 $N$ の遷移関数は次のように決定される。

$\hat{\delta}(a, b, c) = d$ と仮定する。但し、 $a, b, c, d \in Q - \{w\}$ 。この時、 $N$ はFig. 4 Type(I)に示した(1)~(9)の遷移関数を持つ。(1)はどの境界とも隣接していないセル上で使用される規則である。(2)~(7)はそれぞれ上下左右の境界、左下隅、右上隅に隣接しているセル上で使用される規則である。(8)及び(9)は2次元アレイの特殊なケースとして $m=1, n \geq 2$ あるいは $m \geq 2, n=1$ の場合に対応した $N$ の規則である。  
 $a=w$ の時、すなわち、 $\hat{\delta}(w, b, c) = d, b, c, d \in Q - \{w\}$ の時、 $N$ はFig. 4 Type(II)に示した(1)~(3)の遷移関数を持つ。これらの規則はアレイの左上隅に隣接しているセル上で使用される。(2)および(3)は先の(8)、(9)に対応した特殊ケースであ

る。

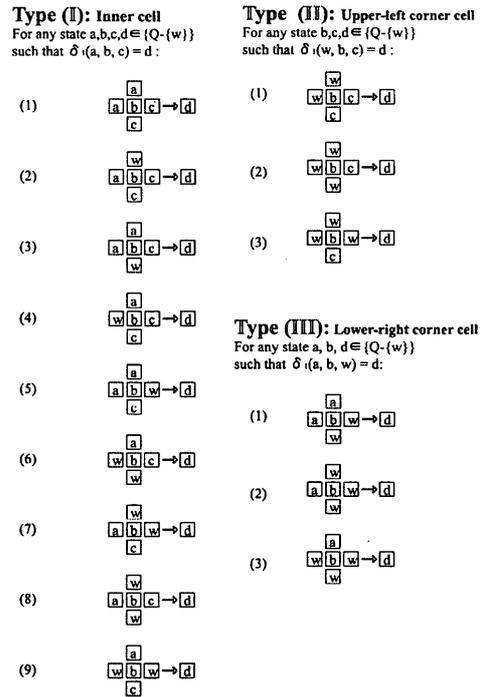


Fig. 4 Construction of transition rules for two-dimensional firing squad synchronization algorithm

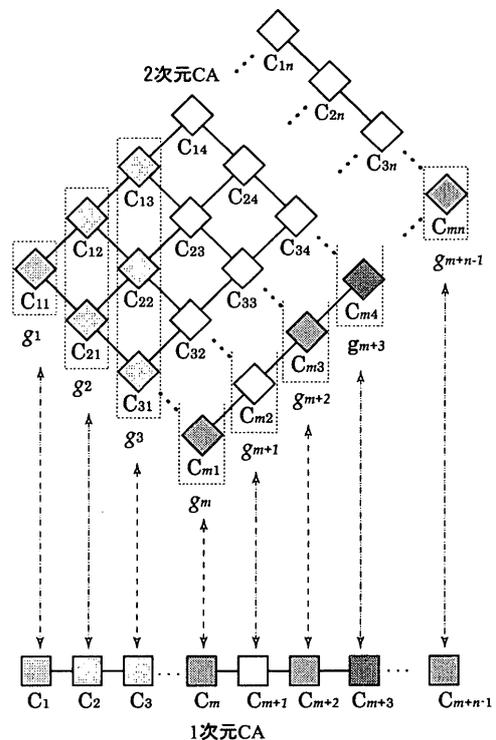


Fig. 5 A mapping between 1-D and 2-D cellular arrays

$c=w$ の時、すなわち、 $\hat{\delta}(a, b, w) = d, a, b, d \in Q - \{w\}$ の時、 $N$ はFig. 4 Type(III)に示した(1)~(3)の遷移関数を持つ。これらの規則はアレイの右下隅に隣接しているセル上で使用される。(2)および(3)は先の(8)、(9)に対応した特殊ケースである。

サイズ $m \times n$ のセル空間を考える。 $N$ 上のセルで $C_{11}$ からの距

離が  $k$  であるセル  $C_{ij}$  の集合を  $g_{k+1}$  とする。但し、 $k$  は  $0 \leq k \leq m+n-1$  なる任意の整数である。  $g_{k+1}$  は次式で定義される。  $g_{k+1} = \{C_{ij} | (i-1) + (j-1) = k\}$  Fig. 5 参照。 すなわち、

- $g_1 = \{C_{11}\}$
- $g_2 = \{C_{12}, C_{21}\}$
- $g_3 = \{C_{13}, C_{22}, C_{31}\}$
- $\vdots$
- $g_{m+n-1} = \{C_{mn}\}$ .

さらに便宜上  $g_0, g_{m+n}$  をアレイの左上隅, 右下隅の境界セルと定義する。 すなわち、

- $g_0 = \{C_{00}\}$
- $g_{m+n} = \{C_{m+1, n+1}\}$ .

$m+n-1$  個のセル  $\{C_i | 1 \leq i \leq m+n-1\}$  より構成される 1 次元 CA を考える。 任意の  $i(i \geq 1)$  に対して  $g_i$  は  $C_i$  の動作を実時間で模倣する。 すなわち  $g_i$  上のすべてのセルは各時刻毎に  $C_i$  の状態をとるように動作する。 時刻  $t(t \geq 0)$  における  $C_i, C_j$  の状態をそれぞれ  $S_i^t, S_j^t$  とする。 この時、次の補題が成り立つ。

**[補題 1]**  $i, t$  を  $1 \leq i \leq m+n-1, 0 \leq t \leq T(m+n-1)$  なる任意の整数とする。  $g_i$  に属するすべてのセルの時刻  $t$  における状態は同一で、それらはすべて  $S_i^t$  に等しい。

(略証) 時刻  $t$  に関する数学的帰納法により証明する。  $t=0$  の時、補題が成立することは次の事実より明らかである。  $t=0$  時  $C_1$  は将軍状態、それ以外のセル  $C_j (2 \leq j \leq m+n-1)$  は静止状態である。 一方  $N$  において、 $t=0$  時  $C_{11}$  は将軍状態、 $C_{11}$  を除くすべてのセルは静止状態にある。 従って、補題が成立する。

次に  $k$  を  $0 \leq k \leq T(m+n-1)-1$  なる任意の整数とし、 $t=k$  の時上記補題が成立すると仮定する。 仮定より、時刻  $k$  における  $g_{i-1}, g_i, g_{i+1} (1 \leq i \leq m+n-1)$  上のすべてのセルは同一の状態をとり、それらの状態をそれぞれ  $a, b, c (a, b, c \in Q)$  とする。 この時、1 次元 CA  $M$  上のセル  $C_{i-1}, C_i, C_{i+1}$  は時刻  $t=k$  時にそれぞれ  $a, b, c (a, b, c \in Q)$  の状態をとっている。  $M$  の遷移関数において、 $\delta(a, b, c) = d$  とする。 2 次元 CA  $N$  は Fig. 4 に示す遷移関数を持つ。 以下の証明は  $m < n$  の時も同様に成立するので  $m \geq n$  と仮定する。 任意の  $i (1 \leq i \leq m+n-1)$  について、 $g_i$  上のすべてのセルは時刻  $t=k+1$  時に  $C_i$  と同じ状態をとることを示す。

(I)  $i=1$  の時、 $a=w$  で、Fig. 4 Type(II) に示す規則(1)~(3)より  $C_{11}^{k+1} = d$  となり、また  $S_1^{k+1} = d$  より、 $S_{11}^{k+1} = S_1^{k+1}$  が成立する。

(II)  $1 < i < n-1$  の時、 $g_i$  上の任意のセル  $C_{xy}$  を考える。 但し  $x+y=i+1$ 。

(1)  $2 \leq x \leq i-1, 2 \leq y \leq i-1$  の時、 $C_{x-1, y}, C_{x, y-1} \in g_{i-1}, C_{x+1, y}, C_{x, y+1} \in g_{i+1}$ 。 従って仮定より、 $S_{x-1, y}^k = S_{x, y-1}^k = a, S_{x+1, y}^k = S_{x, y+1}^k = c$ 、Fig. 4 Type(I) に示す規則(1)より  $S_{xy}^{k+1} = d$  が成り立つ。 また仮定  $S_{i-1}^k = a, S_i^k = b, S_{i+1}^k = c, \delta(a, b, c) = d$  より、 $S_i^{k+1} = d$  が成立する。 従って、 $S_{xy}^{k+1} = S_i^{k+1}$  が成立する。

(2)  $x=1, y=i$  のとき、セル  $C_{1i}$  は上側の境界と隣接している。 この時、 $C_{1, i-1} \in g_{i-1}, C_{2, i} \in g_{i+1}$ 。 従って仮定より、 $S_{1, i-1}^k = a, S_{2, i}^k = b, S_{1i}^k = S_{1, i-1}^k = c$ 。 従って、Fig. 4 Type(I) 規則(2)より  $S_{1i}^{k+1} = d$  が成立する。 また  $S_{i-1}^k = a, S_i^k = b, S_{i+1}^k = c, \delta(a, b, c) = d$  より、 $S_i^{k+1} = d$  が成立する。 従って、 $S_{1i}^{k+1} = S_i^{k+1}$  が成立する。

(3)  $x=i, y=1$  の時、セル  $C_{i1}$  は左側の境界と隣接している。 この時、 $C_{i-1, 1} \in g_{i-1}, C_{i, 2} \in g_{i+1}$ 。 従って仮定より、 $S_{i-1, 1}^k = a, S_{i, 2}^k = b, S_{i1}^k = S_{i-1, 1}^k = c$ 。 従って、Fig. 4 Type(I) 規則(4)より  $S_{i1}^{k+1} = d$  が成立する。 また  $S_{i-1}^k = a, S_i^k = b, S_{i+1}^k = c, \delta(a, b, c) = d$  より、 $S_i^{k+1} = d$  が成立する。 従って、 $S_{i1}^{k+1} = S_i^{k+1}$  が成立する。

(1), (2), (3)より、 $1 < i < n-1$  の時、 $g_i$  上のすべてのセルは  $t=k+1$  時に  $C_i$  と同一の状態をとる。

(III)  $i=n$  の時、 $g_i$  上のセルは Fig. 4 Type(I) 規則(1), (4), (7)を使って状態遷移を行い、時刻  $t=k+1$  時には  $C_i$  と同一の状態をとる。

さらに(IV)  $n+1 \leq i \leq m-1$  の時、(V)  $i=m$  の時、(VI)  $m+1 \leq i \leq m+n-2$  の時、(VII)  $i=m+n-1$  の時も同様な手法により、上記の事実を示すことができる。

以上より、補題が成立することは明らかである。

補題 1 より、長さ  $n$  の 1 次元 CA  $M$  が  $T(n)$  ステップで一斉に

同じ状態になれば、それに対応する 2 次元 CA  $N$  においても同じ時刻に  $N$  上のすべてのセルが同じ状態になる。 以上を次の定理にまとめる。

**[定理 2]**  $M$  を長さ  $n$  のセル空間に対し  $T(n)$  ステップで一斉射撃する 1 次元 CA とする。 サイズ  $m \times n$  の 2 次元セル空間に対し、 $M$  と同じ内部状態数を持ち、 $T(m+n-1)$  ステップで一斉射撃する 2 次元 CA  $N$  が存在する。

よって、この長方形 CA 上での同期アルゴリズムの射撃に要する時間はセル数  $n+m-1$  の 1 次元 CA が射撃に要する時間と同じであり、ベースとなる 1 次元 CA 上での同期アルゴリズムに最適時間アルゴリズムを用いると、同期する時刻は  $t=2(m+n-1)-2=2(m+n)-4$  となる。

また、この設計方法を用いて同期アルゴリズムを設計した場合、ベースに用いる 1 次元 CA 上での同期アルゴリズムの状態数を変えずに、2 次元 CA 上での同期アルゴリズムを設計可能である。 以上より次の定理を得る。

上記の設計方法に基づき、ベースとなる 1 次元 CA 上での同期アルゴリズムに Mazoyer の 6 状態同期アルゴリズムを用いて 6 状態の同期アルゴリズムを設計した。 このアルゴリズムは 2 次元 CA 上での同期アルゴリズムとして最少の状態数である。 このアルゴリズムのシミュレーション結果を Fig. 6 に示す。 以上の結果を次の定理にまとめる。

**[定理 3]**  $m \times n$  個のセルからなる 2 次元セル空間を  $2(m+n)-4$  ステップで一斉に同じ状態にする 6 状態同期アルゴリズムが存在する。

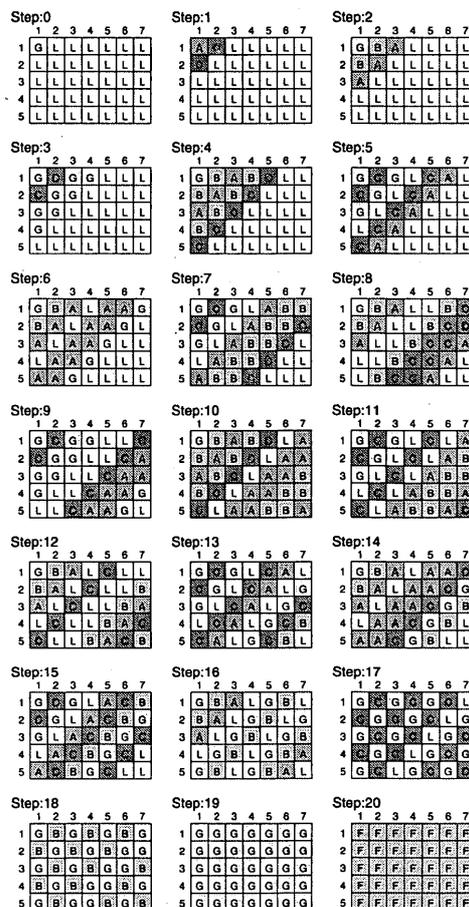


Fig. 6 Snapshots of our 6-state linear-time firing squad synchronization algorithm on 2-D arrays

上記の設計方法は 3 次元直方体 CA にも応用でき、次の定理を得る。

**[定理 4]**  $l \times m \times n$  個のセルからなる 3 次元セル空間を  $2(l + m + n) - 6$  ステップで一斉に同じ状態にする 6 状態同期アルゴリズムが存在する。

さらに上記の設計方法は、複数の長方形故障領域を含む 2 次元セル空間にも応用でき、次の定理を得る。

**[定理 5]** 複数の長方形故障領域を含むサイズ  $m \times n$  の 2 次元セル空間を  $2(m + n) - 4$  ステップで一斉に同じ状態にする 6 状態同期アルゴリズムが存在する。

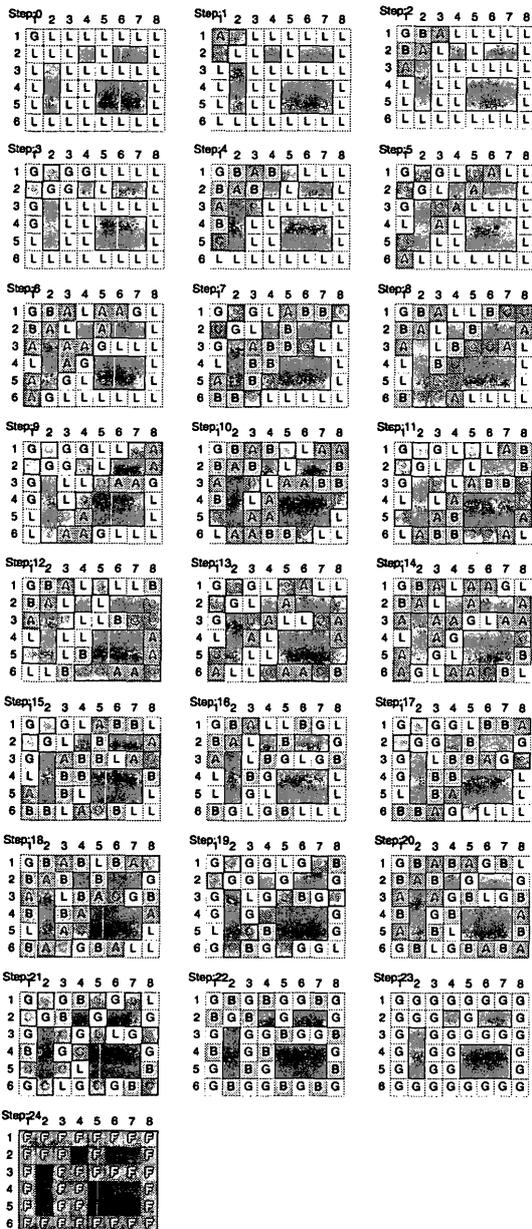


Fig. 7 Snapshots of our 6-state linear-time firing squad synchronization algorithm on 2-D arrays with some isolated rectangular obstacles

## 6. 2 次元 CA 上での一般化一斉射撃アルゴリズム

2 次元上での一般化一斉射撃問題では、 $t=0$  時において将軍セルの位置は任意である。前節で述べたように  $g_x$  内の全てのセルは同一の状態となるように遷移規則を設計しなければならない。よって  $t=0$  時の将軍セルを含むセル集合をどのように遷移させるかが問題となった。

F.R.Moore[5]のアルゴリズムにおける将軍 G は次のステップで将軍位置記憶波状態 D に遷移し、 $t=0$  時に生成され、セル端で反射して返ってきた 1/1 信号と D が衝突することで新たな状態に遷移する。つまり  $t=0$  時に生成された 1/1 信号がセル端で反射して返ってくるまでに将軍セルを含むセル集合のセル全てを将軍位置記憶波状態 D に遷移させることでこの問題を解決した。

### 将軍位置記憶波状態 D の生成方法

将軍位置記憶波状態 D は 1/1 信号 L,S から情報を得ることができるためステップ 2 までは必ず生成される。しかし 1/1 信号 L,S だけではステップ 3 以降 D を生成することができない。そこで新たな状態 {D1, D2, D3, D4} という 4 つの新しい状態を作ることにより、前節の問題点を解決することができる。

追加した状態 { D1, D2, D3, D4 } はステップ数  $t=2i+1$  ( $i \geq 1$ ) 時に生成され、将軍位置記憶波状態 D は  $t=2^j$  ( $j \geq 0$ ) 時に生成される。生成された D は  $t=0$  時に将軍セルから生成され、セル端で反射して返ってきた 1/1 信号と衝突するまでその状態を保持し、設計方法で解説した  $m+n-1$  個のセルからなる全てのセル集合  $g_x$  は Moore[5]の一般化一斉射撃アルゴリズムの動作を模倣して遷移していく。

本稿では状態数 17 である Moore[5]の一般化一斉射撃アルゴリズムをベースとして用いたため、追加した状態 {D1, D2, D3, D4} を合わせて状態数 21 の一般化一斉射撃アルゴリズムを設計した。以上の結果を次の定理にまとめる。

**[定理 6]**  $m \times n$  個のセルからなる 2 次元セル空間を  $2(m + n) - \min((k+1) - 2), (m+n - k - 1)) - 4$  ステップで一斉に同じ状態にする 21 状態同期アルゴリズムが存在する。

## 7. まとめ

正方形 CA 上で最適時間で動作する 9 状態  $2n - 2$  ステップ一斉射撃アルゴリズムの設計と実装を行い、セル数  $2 \times 2$  から  $1000 \times 1000$  までの全ての正方形 CA において上記のアルゴリズムが正しく動作することを確認した。また長方形 CA 上での 6 状態  $2(m + n) - 4$  ステップ一斉射撃アルゴリズムについても同様にシミュレーションを行い、セル数  $2 \times 2$  から  $500 \times 500$  までの全ての長方形 CA 上で正しく動作することを確認した。

また、長方形 CA 上で動作する状態数 21 の  $2(m+n) - \min((k+1) - 2), (m+n - k - 1)) - 4$  ステップ一般化一斉射撃アルゴリズムの設計と実装を行い、セル数  $2 \times 2$  から  $500 \times 500$  までの全ての長方形 CA 上において上記のアルゴリズムが正しく動作することを確認した。従来から知られている Szwerinski[9]のアルゴリズムに比べると内部状態数は 25600 から 21 に大幅に削減した。

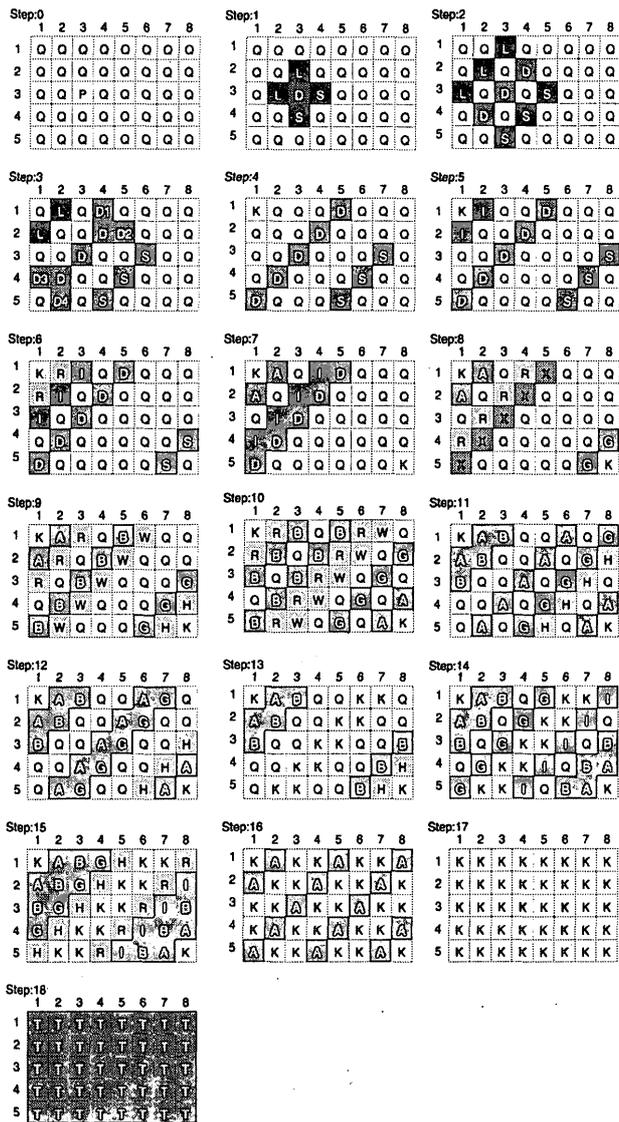


Fig. 8 Snapshots of our 21-state linear-time generalized firing squad synchronization algorithm on 2-D arrays, where the general locates at any position on the array

参考文献

[1] M. Minsky and J. McCarthy, "Computation: Finite and infinite machines", Prentice Hall, pp. 28-29 (1967).  
 [2] A. Waksman, "An optimum solution to the firing squad synchronization problem", Information and Control, 9, pp. 66-78 (1966).  
 [3] R. Balzer, "An 8-state minimal time solution to the firing squad synchronization problem", Information and Control, 10, pp. 22-42 (1967).  
 [4] J. Mazoyer, "A six-state minimal time solution to the firing squad synchronization problem", Theoretical Computer Science, 50, pp. 183-238 (1987).  
 [5] F. R. Moore & G. G. Langdon, "A generalized firing squad problem", Information and Control, 12, pp. 210-220 (1968).  
 [6] I. Shinahr, "Two- and three-dimensional firing-squad synchronization problems", Information and Control, 24, pp.

163-180 (1974).

[7] A. Grasselli, "Synchronization of cellular arrays: The firing squad problem in two dimension", Information and Control, 28, pp. 113-124 (1975).  
 [8] V. I. Varshavsky, V. B. Marakhovskiy and V. A. Peschansky, "Synchronization of Interacting Automata", Mathematical System Theory, Vol. 4, No. 3, pp. 212-230 (1969).  
 [9] H. Szwerinski, "Time-optimal solution of the firing-squad-synchronization-problem for n-dimensional rectangles with the general at an arbitrary position", Theoretical Computer Science, 19, pp. 305-320 (1982).  
 [10] J. Mazoyer, "On optimal solution to the firing squad synchronization problem", Theoretical Computer Science, 168, pp. 367-404 (1996).  
 [11] H. Umeo, J. Nishimura and T. Sogabe, "1-Bit Inter-Cell Communication Cellular Algorithms", Proc. of the 8th International Colloquium on Numerical Analysis and Computer Science with Applications (1999).  
 [12] M. Maeda and H. Umeo, "A Design of 6-State 3n-Step Firing Squad Synchronization Algorithm and Its Implementation", Proc. The 14th Annual of Conference of Japanese Society for Artificial Intelligence, pp. 79-80 (2000).  
 [13] H. Umeo, J. Nishimura, T. Sogabe and M. Maeda, "A Design of Synchronization Algorithm for a Large Scale of Cellular Automata", Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th'01) Tokyo, pp. 381-386 (2001).  
 [14] M. Maeda and H. Umeo, "A Design of Two-Dimensional Firing Squad Synchronization Algorithms and Their Implementations", Proc. The 15th Annual Conference of Japanese Society for Artificial Intelligence, 2C3-05, pp. 1- 4, (2001).  
 [15] 藤原, 前田, 梅尾, "2次元セルラオートマトン上での一斉射撃アルゴリズムの設計と実装", 2001年電子情報通信学会情報・システムソサイエティ大会, p. 1, (2001).  
 [16] 前田, 梅尾, "2次元セルラオートマトン上での一斉射撃アルゴリズム", 情報処理学会アルゴリズム研究会, 第81回アルゴリズム研究会, (2001年11月, 27日, 於新潟, 発表予定).