数理解析研究所講究録
950巻 1996年 228-232

2次元可逆セル・オートマトンにおける一斉射撃問題

Firing Squad Synchronization Problem
in Two-Dimensional Reversible Cellular Automata

安達 太一, 古阪 真一, 今井 克暢, 森田 憲一
Taichi ADACHI,Shinichi FURUSAKA,Katsunobu IMAI and Kenichi MORITA
広島大学 工学部
〒739 東広島市鏡山 1-4-1
Faculty of Engineering, Hiroshima University,
Highashi-Hiroshima-shi, 739 Japan
{adachi,furusaka,imai,morita}@ke.sys.hiroshima-u.ac.jp

1 はじめに

セル・オートマトンにおける同期問題である一斉射撃問題は今日までに様々な研究がなされている. 1次元セル・オートマトンの一斉射撃問題 [1] は, 任意の有限の長さ n の1次元3近傍セル・オートマトンの左端のセルからの信号により, すべてのセルを同時刻に特別の状態 (射撃状態) にする問題である.

Minsky と McCarthy は, $3n$ ステップで射撃状態になる同期解を示し, その後, 後藤や Waksman [2] らによって $2n-2$ ステップで同期する最小時間解が得られた. 現在は Mazoyer[4] による6状態の最小時間解が知られている.

また2次元に拡張した問題に対しても Rosenstiehl は $4n-6$ 時間解を示し, Kobayashi [5] はいくつかの図形のクラスに対するより高速な解を示した.

われわれは, この一斉射撃問題を可逆セル・オートマトン上で実現することを試みた. ところが, 1次元可逆セル・オートマトンにおいては従来の単一の状態になるという意味での一斉射撃解は存在せず, 複数の射撃状態を許すという拡張した一斉射撃解条件を満たす解が $3n$ 時間解を元に構成可能であることを示した [8] .

本稿では, 2次元可逆セル・オートマトン上での同期解を構成したことについて報告する.

まず2次元図形のあるクラスに対する一斉射撃解を Kobayashi の方法を基に構成し, 次に任意の図形に対する解を構成するため, Rosenstiehl の解を可逆 CA に埋め込むことを試みた. その結果, 状態数がかなり増加するものの任意の図形に対しても一斉射撃解を構成できることが分かった.

解の構成にあたっては, 可逆セル・オートマトンの設計を容易にするために分割セル・オートマトン (PCA)[7] を用いた.

2 可逆セル・オートマトンに対する一斉射撃解条件

2.1 諸定義

決定性2次元セル・オートマトン (CA) は

$$A = (\mathbf{Z}^2, Q, N, \varphi_A, \#)$$

で定義される. ただし,

- $\mathbf{Z}$ は全整数の集合,
- Q は各セルの状態の空でない有限集合,
- $N = (x_1, x_2, \ldots, x_k)$ は近傍ベクトルで, $x_i \in \mathbf{Z}^2$, kは自然数,
- $\varphi_A : Q^k \to Q$ は局所写像,
- $\# \in Q$ は静止状態で, $\varphi_A(\#, \#, \cdots, \#) = \#$,

である. A の状相 c は $c : \mathbf{Z}^2 \to Q$ なる写像で, Q 上のすべての状相の集合 Conf(Q) は,

$$\mathrm{Conf(Q)} = \{\mathrm{c} | \mathrm{c} : \mathbf{Z}^2 \to \mathrm{Q}\}$$

で表される. 大域写像 $\Phi_A : \mathrm{Conf(Q)} \to \mathrm{Conf(Q)}$ は

$$\forall x \in \mathbf{Z}^2, \Phi_A(c)(x) = \varphi_A(c(x + x_1), \ldots, c(x + x_k))$$

で定義される. すなわち, ある状相 c に対して, Φ_A を適用した次の時刻の状相は $\Phi_A(c)$ となる. また, k

ステップ後の状相は $\Phi_A^k(c)$ と表すことにする．**可逆CA** とは大域写像 Φ_A が単射であるような CA である．なお，$N=((0,0),(0,1),(1,0),(0,-1),(-1,0))$ であるような CA を **2次元ノイマン近傍** CA と呼ぶ．以降，2次元ノイマン近傍 CA を 2 次元 CA とよび，$(\mathbf{Z}^2,Q,\varphi_A,\#)$ と略記する．

2 次元 CA A の 2 つのセル $p=(x,y),p'=(x',y'),p,p'\in\mathbf{Z}^2$ が

$$(x=x'\wedge|y-y'|=1)\vee(|x-x'|=1\wedge y=y')$$

なる関係を満たすとき，p,p' は **隣接** しているとよぶ．また，系列 $p_0,p_1,\cdots,p_n,(p_0,p_1,\cdots,p_n\in\mathbf{Z}^2)$ が $p_0=p,p_n=p'$ かつ，すべての $i(0\le i<n)$ について p_i,p_{i+1} が隣接しているとき，その系列を p から p' への**パス** とよぶ．$M\in\mathbf{Z}^2$ が **連結** であるとは，すべての $p,p'\in M$ について，$p\to p'$ なるパスがあることを言い，M が有限個のセルの連結で，かつ $(0,0)$ を含む場合，*Figure* とよぶ．

2.2 2次元可逆 CA に対する一斉射撃解条件

可逆 CA では単一の射撃状態の一斉射撃解を構成できないため，1 次元可逆 CA に対する一斉射撃解条件 [8] と同様にして，2 次元可逆 CA に対する一斉射撃解条件を次のように定義する．

2 次元可逆 CA に対する一斉射撃解条件
2 次元 CA $A=(\mathbf{Z}^2,Q,\varphi_A,\#)$ において，任意の Figure $M(\subset Z^2)$ について，相異なる 2 つの状態 $g,s\in Q-\{\#\}$ と状態集合 $F\subset Q-\{\#,g,s\}$ が存在し，つぎの 1,2 を満たす．

1. $\forall u_1,u_2,u_3,u_4\in\{s,\#\}$,
   $\varphi_A(s,u_1,u_2,u_3,u_4)=s$.
2. ${c_s}^{(M)}$ を次のような状相とする．

$${c_s}^{(M)}(x)=\begin{cases} g & x=(0,0)\\ s & x\in M,x\neq(0,0)\\ \# & x\notin M\end{cases}$$

このとき，$\mathbf{Z}^2$ の部分集合から自然数への関数 t_f が存在し，

$$\forall x\in\mathbf{Z}^2$$
$$((x\in M\Rightarrow\Phi_A^{t_f(M)}({c_s}^{(M)})(x)\in F)$$
$$\wedge(x\notin M\Rightarrow\Phi_A^{t_f(M)}({c_s}^{(M)})(x)\notin F)),$$

$$\forall i\in\mathbf{Z}_+(0\le i<t_f(M)$$
$$\Rightarrow\forall x\in\mathbf{Z}^2,(\Phi_A^{t_f(M)}(c_s^{(M)})(x)\notin F)).$$

が成り立つ．

すなわち，複数の状態からなる射撃状態集合 $F(\subset Q)$ を考え，$t=t_f(M)$ において，各セルの状態が F の要素になっていれば同期したと考えるものであり，同期させる M に含まれるセルだけでなく，その外側のセルの状態の変更も許すよう，従来の一斉射撃解条件を緩和したものである．

2.3 分割セル・オートマトンの定義

可逆セル・オートマトンの一斉射撃解の定義に基づいて解を構成するにあたっては，分割セル・オートマトン (PCA) を用いた．まず，PCA の定義を示す．

2 次元ノイマン近傍 PCA P は 2 次元 CA のサブクラスとみなすことができ，

$$P=(\mathbf{Z}^2,(C,U,R,D,L),\varphi_P,(\#,\#,\#,\#,\#))$$

であらわされ，

- $\mathbf{Z}$ は全整数の集合，
- C,U,R,D,L, はそれぞれ 5 分割した各セルの中央，上、右、下、左パーティションの状態の空でない有限集合，
- $\varphi_P:C\times D\times L\times U\times R\to C\times U\times R\times D\times L$ は局所関数．
- $(\#,\#,\#,\#,\#)\in C\times U\times R\times D\times L$ は静止状態で$\varphi_P(\#,\#,\#,\#,\#)=(\#,\#,\#,\#,\#)$.

である．P の状相 c は $c:\mathbf{Z}^2\to C\times U\times R\times D\times L$ なる写像で，$C\times U\times R\times D\times L$ 上のすべての状相の集合 Conf(C × U × R × D × L) は，

$$\mathrm{Conf(C\times U\times R\times D\times L)}=\{\mathrm{c|c}:\mathbf{Z}^2\to\mathrm{C\times U\times R\times D\times I}$$

で表される．大域関数

$$\Phi_P:\mathrm{Conf(C\times U\times R\times D\times L)}\to\mathrm{Conf(C\times U\times R\times D\times L}$$

は，$C\times U\times R\times D\times L$ から C,U,R,D,L をとりだす射影関数をそれぞれ，$CENTER,UP,RIGHT,DOWN,LEFT$ とすると，

$$\forall x\in\mathbf{Z}^2,\Phi_P(c)(x)=\varphi_P(CENTER(c(x)),$$
$$UP(c(x+(0,1))),$$
$$RIGHT(c(x+(1,0))),$$
$$DOWN(c(x+(0,-1))),$$
$$LEFT(c(x+(-1,0))))$$

で定義される．P が可逆であるための必要十分条件は，φ_P が 1 対 1 写像であることが示されているので [7]，容易に可逆な CA を設計することができる．

3 可逆 PCA による Figure の特定のクラスに対する一斉射撃解

3.1 Figure C1

Kobayashi は Figure のクラス Mstep に対してセル (0,0) からの距離に関して 1 次元化することで一斉射撃解を構成している．この特徴を保持し、Mstep を真に含む Figure のクラス C1 を次のように定義する．

1. セル (0,0) を G とする．
2. 任意のセルへは、G から適切に正方向 (右・上方向) に進むことで到着できる．
3. 任意のセルから正方向に進むことで，いずれかの極大のセル (正方向に隣接するセルを持たないセル) に着くことができる．
4. すべての極大のセルは G からの距離が等しい．

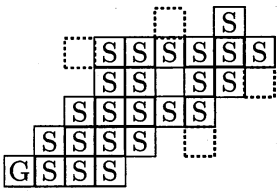

図 1: C1 の例 (点線のセルがあると C1 ではなくなる)

3.2 C1 の可逆解

解の構成にあたっては，2 次元のセルを 1 次元化し，1 次元の解を埋め込むことによって構成する．そのため，まず，すべてのセルにそれぞれの接続状況によって 9 個のタイプの状態のひとつを持たせる (図 2).

その結果，この C1 においては，それぞれのセルの正方向，負方向 (左・下方向) はそれぞれ同じ動きをさせることができるので，1 次元のように扱えるようになり 1 次元の可逆解と 9 個のタイプを掛け合わせることで，C1 の可逆解が求まることになる．

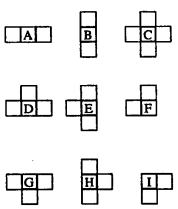

図 2: セルのタイプ

実際に構成した解 P_K を示す．だたし，各セルの内部状態を表現する際に， Aa, Ab, Ba, Bb のような状態をまとめて，$(A,B)(a,b)$ と略記する．括弧内の文字が 1 文字の時括弧は省略する．

- $P_K =$
  $(\mathbf{Z}^2, (C_K, U_K, R_K, D_K, L_K), \varphi_K, (\#,\#,\#,\#,\#))$,
  $C_K = \{\#, (A,B,C,D,E,F,G,H,I)(t, \overrightarrow{s}, \overleftarrow{s}, \overrightarrow{e}, \overleftarrow{e}, u, \overrightarrow{w}, \overleftarrow{w}, v, f, \acute{s}, \acute{e}, \acute{u}, \acute{w})\}$,
  $U_K = R_K = D_K = L_K = \{\#, +, *, 1\}$

- General を $(\overrightarrow{As}, *, *, \#, \#)$，soldier を $(\overrightarrow{Cs}, \#, \#, \#, \#)$，静止状態を $(\#,\#,\#,\#,\#)$ とする．
- 射撃状態集合はセンターセルが $(A,B,C,D,E,F,G,H,I)f$ のもの全てとする．
- 局所関数 φ_K の遷移規則は省略 (後述の Web サイトを参照).

次に解の概略を示す．

まず General からセルのタイプを分類するための信号 “*” を出す (図 3).

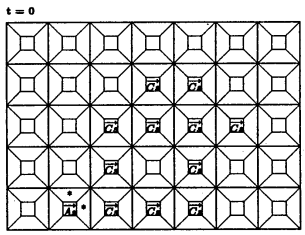

図 3: コンフィグレーション ($t = 1$)

信号 “*” を受けたセルは soldier なら “1” を空白なら “*” を左側に返し，信号 “*” を正方向に送る (図 4).

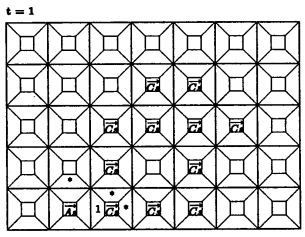

図 4: コンフィグレーション ($t = 2$)

戻ってきた信号を受けた General はセルのタイプを決定し、一斉射撃のための速度 1 の信号を正方向に出す (図 5).

引き続くセルは、正方向から戻ってくる信号と、負方向からの速度1の信号からセルのタイプを決定する(図6).

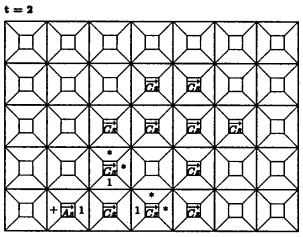

図 5: コンフィグレーション ($t=3$)

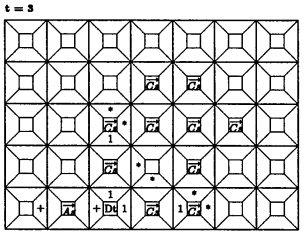

図 6: コンフィグレーション ($t=4$)

以下同様にしてセルのタイプを決定していき、極大のセルで速度1の信号を跳ね返し、速度1/3の信号と衝突させFigureを2分割させる.

その後の遷移規則は基本的には1次元の解 P_1 とセルのタイプの直積をとったものを用いればよいが、最初の2分割で生成したgeneralは可逆性を保つために違う状態を使用しているために、別に用意する必要がある.

このようにして構成した解の状態数は, Cが127状態, U,R,D,L が各4状態で, 32512状態, 射撃時間 t_fは, ここで用いた1次元の一斉射撃解の射撃時間が隊列の長さを n とすると $3n-1$ 時間であることから, セル(0,0)から極大のセルまでの長さを Kとおくと $T=3K-1$ となる.

4 任意の2次元図形に対する一斉射撃解

4.1 Rosenstiehl のアルゴリズム

任意の2次元図形に対する一斉射撃解はRosenstiehl(1966)によって示された. その射撃時間はセルの個数が Nとすると $4N-6$ であり, 以下のようなものである.

N個のセルからなる任意のFigure $M \subset Z^2$ が与えられたとする. Mのすべてのセルを通るようにセル(0,0)から次の規則にしたがってパスを引き1次元化する.

1. p をセル(0,0)とする(すなわちGeneral). 2.に行く.
2. p がまだ訪れたことのない, 隣接したセルがあるかどうかを調べる. もしなければ3.にいく. もしあればその中で最も優先順位の高いセル(p に対して, 右, 上, 左, 下のセルの順で, 右が最も優先順位が高い)を選択する. p を選択されたセルにし, 2.を繰り返す.
3. $p_1,\cdots,p_m$ を p に隣接するセルとし, その中に p からまだ進んだことのない p_i が1つ存在する. もし p_i がセル(0,0)で, セル(0,0)がまだ訪れたことのない隣接したセルがないとき, 終了する. もしそうでなければ p を p_i にして2.に行く.

このアルゴリズムによって, のべ $2N-2$ 個のセルに対するパスが引ける(図7). このパスの起点と終点とはともにGeneralセルであり, このパス上に1次元の一斉射撃問題の解をのせることで任意の2次元図形に対する一斉射撃解が得られる. 1次元の一斉射撃問題の解に最少時間解を用いるとき, この図形 Mの射撃時間は $2(2N-2)-2=4N-6$ になる.

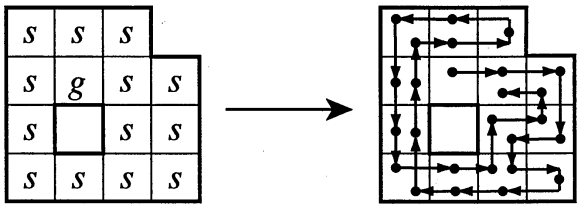

図 7: Rosenstiehl のパスの例

4.2 可逆PCAによるRosenstiehlの解の構成

4.2.1 Rosenstiehl のパスの構成

Rosenstiehl のパスを引くアルゴリズムに基づき, 任意の図形の1次元化をおこなう 可逆 PCA を構成した.

- $P_R=$
  $(\mathbf{Z}^2,(C_R,U_R,R_R,D_R,L_R),\varphi_R,(\#,\#,\#,\#,\#))$,
  $C_R=\{\#, aa, ab3, ab34, ab4, aba, aba01, aba02,$
  $aba03, abca, abca01, abd3, abda, abda01, abda02,$
  $abdca, aca, ad3, ada, ada01, adca, ba3, ba34, ba4,$

$bab, bab01, bab02, bab03, bacb, bacb01, bad3, badb,$
$badb01, badb02, badcb, bb, bcb, bd3, bdb, bdb01,$
$bdcb, ca4, cac, cac01, cadc, cb1, cb14, cb4, cba4,$
$cbac, cbac01, cbac02, cbadc, cbc, cbc01, cbc02, cbc03,$
$cbdc, cbdc01, cc, cdc, da3, dacd, dad, dad01, db1,$
$db13, db3, dba3, dbacd, dbad, dbad01, dbad02, dbcd,$
$dbcd01, dbd, dbd01, dbd02, dbd03, dcd, dd, g, g\#,$
$ga, ga01, ga02, ga3, ga34, ga4, gac, gac01, gad,$
$gad01, gad02, gad3, gadc, gb, gb01, gb02, gb03,$
$gb04, gb05, gb06, gb1, gb13, gb134, gb14, gb3,$
$gb34, gb4, gba, gba01, gba02, gba03, gba04, gba05,$
$gba06, gba07, gba08, gba3, gba301, gba34, gba4,$
$gba401, gbac, gbac01, gbac02, gbad, gbad01, gbad02,$
$gbad03, gbad3, gbadc, gbc, gbc01, gbc02, gbc03, gbd,$
$gbd01, gbd02, gbd03, gbd04, gbd05, gbd3, gbd301,$
$gbdc, gbdc01, gd01, gd3, gdc, s\},$
$U_R = R_R = D_R = L_R = \{\#, ?, +, -, =\}$

- 局所関数φ_Rの遷移規則は省略 (後述の Web サイトを参照)

次にこの解の概要を説明する.

Rosenstiehl の方法も Kobayashi の方法同様にセル間の接続関係に従って各セルを場合分けすることになる. セルの上・右・下・左の面をそれぞれ

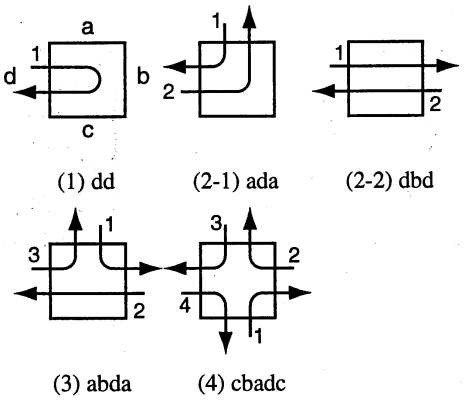

図 8: パスの符号化例 (soldier cell)

“a”,“b”,“c”,“d” (アルファベット表現の場合) または “1”,“2”,“3”,“4” (数による表現の場合) で表すことにする. セルのタイプは, セルに進入するパスの面に対応するアルファベットを順に並べた文字列で表現する. セルに進入するパスの数は高々セルの面の数 (すなわち 4) であるため, soldier セルのタイプを表す文字列は図 8のような 5 種類 32 に分類でき, 2 文字以上 5 文字以下で, 最初と最後の文字が同じアルファベットになる. General セルは, “g” で始まり, その後にパスを表すアルファベットが結合した形式で表現する. 引き続く soldier セルも同様の方法により自身の

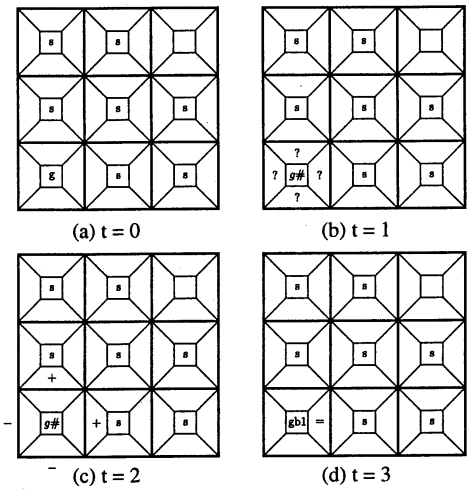

図 9: パスの生成開始

タイプを決定しパスを生成していく. まず, General

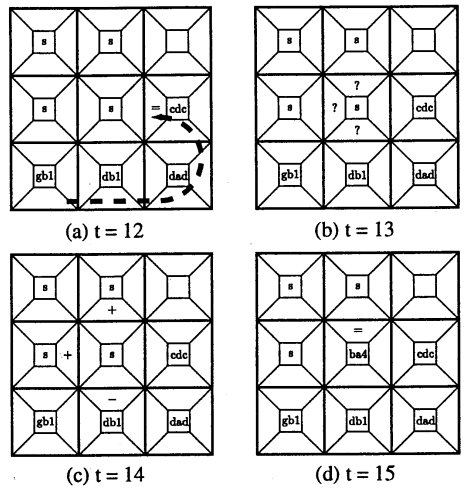

図 10: セルのタイプの決定過程

セルから隣接するセルを調べるための信号 “?” を出し (図 9(b)), これを受けた隣接セルは soldier であれば信号 “+” を, 空白セルであれば “−” を信号が来た方向に返す (c). 返ってきた信号によってセンターパーティションの状態を決定する. ここに挙げた例では, 右と上へ進むことができるが, 右の方が優先度が高いため右へ進む. そのとき, パスが右へ連結することを示す “b” を付加し, さらに, 上方向が未到達であることを “1” を付加することで記憶し, 次に進む “b” 方向のセルに対し信号 “=” を出す (d). 例示している図形の中心にある soldier セルの場合について