

## 16 セグメントディスプレイ上の英字パターンの一斉射撃問題

山下 和也<sup>†</sup>, 酒井 充<sup>†</sup>, 広瀬 貞樹<sup>†</sup>, 大川 知<sup>††</sup>, 大里 延康<sup>†††</sup>, 西谷 泰昭<sup>††††</sup>

<sup>†</sup> 富山大学工学部

<sup>††</sup> 会津大学コンピュータ理工学部

<sup>†††</sup> 大阪工業大学情報科学部

<sup>††††</sup> 岩手大学工学部

あらまし 一斉射撃問題は、セルオートマトンに関する有名な問題の1つで、1957年頃に J. Myhill によって提起され、E. F. Moore [1] によって広く知らされた問題である。この一斉射撃問題に対して、M. Minsky と J. McCarthy [2] によって最初の一斉射撃アルゴリズムが示され、E. Goto [3] によって最小時間のアルゴリズムが示されている。その後、最小時間アルゴリズムの状態数を減らす研究 [4-7] や、この問題を拡張した問題に関する研究 [8-16] 等、多くの研究がなされている。最近、山下等 [17,18] は7セグメントディスプレイ上の数字パターンの一斉射撃問題を定義し、それらの問題に対する一斉射撃アルゴリズムを示した。本稿では、この問題を拡張し、16セグメントディスプレイ上の英字パターンの一斉射撃問題を新しく提案する。また、それら26種の英字パターンの一斉射撃問題に対する一斉射撃アルゴリズムを与える。

### The Firing Squad Synchronization Problems for English Character Patterns on Sixteen-Segment Display

Kazuya YAMASHITA<sup>†</sup>, Mitsuru SAKAI<sup>†</sup>, Sadaki HIROSE<sup>†</sup>,  
Satoshi OKAWA<sup>††</sup>, Nobuyasu OSATO<sup>†††</sup>, Yasuaki NISHITANI<sup>††††</sup>

<sup>†</sup> Faculty of Engineering, University of Toyama

<sup>††</sup> Software Department, Aizu University

<sup>†††</sup> Faculty of Information Science and Technology, Osaka Institute of Technology

<sup>††††</sup> Faculty of Engineering, Iwate University

**Abstract** The firing squad synchronization problem, one of the most famous problems on cellular automata, was originally proposed by J. Myhill in 1957 and became famous through E. F. Moore [1]. The first solution to this problem was given by M. Minsky and J. McCarthy [2] and a minimal time solution was given by E. Goto [3]. Researches to reduce the number of states of the minimal time solution [4-7] and some variants of this problem [8-16] have been studied. Recently, K. Yamashita and others [17,18] defined the firing squad synchronization problems for number patterns on seven-segment display and gave solutions to these problems. In this paper, we extend these problems to English character patterns on sixteen-segment display and give solutions to these problems for twenty-six English character patterns.

### 1. ま え が き

一斉射撃問題は1957年頃に J. Myhill によって提起され、E. F. Moore [1] によって広く知らされた問題である。最初にこの問題の解を示したのは M. Minsky と J. McCarthy [2] で、状態数  $13, 3n + O(\log n)$  ステップで射撃状態に移させるアルゴリズムを示した。この問題の解としては少なくとも  $2(n-1)$  ステップ必要であることは容易に示されるが、実際に E. Goto [3] によって、 $2(n-1)$  ステップで射撃状態に移させる最小時間のアルゴリズム

が示された。Goto のアルゴリズムは状態数が数千にも及ぶ怪物的なものであったので、その後のより簡単な解を求める研究を促進させた。1966年に A. Waksman [4] によって状態数16、1967年に R. Balzer [5] によって状態数8、1987年に H. D. Gerken [6] によって状態数7、J. Mazoyer [7] によって状態数6のアルゴリズムがそれぞれ示されている。また、文献 [5] で一斉射撃問題の解として少なくとも5状態必要であることが示されている。

一斉射撃問題を拡張した研究も多くなされている。セル列を輪にした問題 [8]、2次元や3次元配列へ拡張した問

題 [9,10], ケイリーグラフへ拡張した問題 [11], 特定のグラフへ拡張した問題 [12] 等が研究されている。また, 可逆型セルオートマトンに対する問題 [13], 保存的セルオートマトン (状態は合計が一定となる非負の整数の組である) に対する問題 [14], 隣接するセル間の情報量を制限した問題 [15,16] 等がある。

最近, 山下等 [17,18] は 7 セグメントディスプレイ上の数字パターンの一斉射撃問題を定義し, それらの問題に対する一斉射撃アルゴリズムを示した。

本稿では, この問題を拡張し, 16 セグメントディスプレイ上の英字パターンの一斉射撃問題を新しく提案する。また, それら 26 種の英字パターンの一斉射撃問題に対する一斉射撃アルゴリズムを与える。

## 2. 一斉射撃問題

一斉射撃問題は, セルオートマトンに関する有名な問題の 1 つである。

### 2.1 セルオートマトン

セルオートマトンとは, 規則的な空間にセルと呼ばれる同一の有限オートマトンが配置されたものである。各セルは近傍 (自分自身及び隣接するセル) の状態から遷移規則にしたがって次の状態へ遷移する。この状態遷移は, 全てのセルが同期して同時に行う。各セルが 1 回状態遷移する時間を 1 ステップと呼ぶ。

一般的に良く用いられるセルオートマトンは 1 次元と 2 次元のものである。1 次元の場合には, 各セルは直線上に配置され, その近傍は自分自身とその左右のセルの 3 つである。よってこのようなセルオートマトンは 3 近傍 1 次元セルオートマトンと呼ばれる。2 次元の場合には, 各セルは正方形格子上に配置され, 近傍としては次の 2 つのタイプが一般的である。1 つは Neumann 近傍と呼ばれるもので, 自分自身とその上下左右の 5 つのセルが近傍である。もう 1 つは Moore 近傍と呼ばれるもので, Neumann 近傍にさらに斜め (対角線上) 上下にある 4 つのセルを加えた 9 つのセルが近傍である。前者は 5 近傍 2 次元セルオートマトン, 後者は 9 近傍 2 次元セルオートマトンと呼ばれる。

### 2.2 一斉射撃問題の定義

一斉射撃問題は, 3 近傍 1 次元セルオートマトン上で定義される以下のような問題である。

『 $n$  個のセル  $C_1, C_2, \dots, C_n$  からなる 1 次元セル列を考える。セルの個数  $n$  は任意であるが, 有限とする。左端のセル  $C_1$  は将軍と呼ばれ, 残りのセル  $C_2, C_3, \dots, C_n$  は兵士と呼ばれる。時刻  $t = 0$  で将軍だけは将軍状態と呼ばれる状態で, 兵士は静止状態と呼ばれる状態である。各セルの動作は同期していて, 時刻  $t + 1$  での状態は, 時刻

$t$  での自分の状態と隣接するセルの状態から, 遷移規則にしたがって決められる。ただし, 自分自身も隣接するセルも静止状態である場合には, 次の状態も静止状態である。このとき, ある時刻に全てのセルを一斉に射撃状態と呼ばれる状態に遷移させることができるように, セルオートマトンの状態と遷移規則を定めよ。』<sup>(注1)</sup>

### 2.3 最小時間アルゴリズム

一斉射撃問題に対しては多くのアルゴリズムが知られており, 最小時間アルゴリズムも既にいくつか知られている。ここでは, わかり易い Waksman [4] の最小時間アルゴリズムの考え方を簡単に説明する。

時刻  $t = 0$  に将軍  $C_1$  は速度  $\frac{1}{1}, \frac{1}{3}, \frac{1}{7}, \dots, \frac{1}{2^k-1}, \dots$  ( $k$  は  $1 \leq k \leq \lfloor \log_2(2n-2) \rfloor$  を満たす自然数) の信号<sup>(注2)</sup>を右方向に送出する。これら一連の信号を射撃信号と呼ぶ。時刻  $t = n-1$  に速度  $\frac{1}{1}$  の信号がセル  $C_n$  に到達し,  $C_n$  は将軍状態となる。将軍状態になったセル  $C_n$  は  $C_1$  と同様の射撃信号を左方向に送出する。時刻  $t = 0$  に  $C_1$  が送出した速度  $\frac{1}{3}$  の信号と時刻  $t = n-1$  に  $C_n$  が送出した速度  $\frac{1}{1}$  の信号が  $C_1$  と  $C_n$  の間で衝突する。信号が衝突した場所にあるセルは将軍状態となる。将軍状態になったこのセルは射撃信号を左方向と右方向に送出する。このように信号が将軍状態の中間で衝突し, その場所のセルが将軍状態になることを繰り返すことで, セル空間を再帰的に均等に分割することができる。均等に分割していくといずれ全てのセルが将軍状態になる。各セルは, 全てのセルが将軍状態になったかどうか, 自分自身と隣接するセルが将軍状態になったことでわかるので, その次のステップに一斉に射撃状態に遷移することができる。

このような一連の処理にかかる時間は, 速度  $\frac{1}{1}$  の信号が 1 次元セル列を往復する時間に等しく,  $2(n-1)$  ステップであることがわかる。

図 1 は最小時間アルゴリズムの時間空間図式で, セル列  $C_1, C_2, \dots, C_n$  が状態遷移する過程を上から下へと時系列順に並べたものである。横軸はセル空間, 縦軸は時間を表し, 図中の分数は射撃信号の速度を表している。黒い点はそのセルが将軍状態になったことを示している。

## 3. 16 セグメントディスプレイ上の英字パターンの一斉射撃問題

本章では, 山下等 [17,18] の定義した 7 セグメントディ

(注1): 簡単にいえば『 $C_1$  を将軍,  $C_2, C_3, \dots, C_n$  を兵士とし, 情報はそれぞれ隣同士でしかやりとりできない状況とする。将軍が時刻  $t = 0$  に「一斉に射撃せよ!」という命令を出したとき, どのようにすれば一斉に射撃できるか。』という問題である。

(注2): 信号とは, ある特定の状態 (情報) がセル列を移動することを指している。分数で信号の速度を表し,  $t$  ステップに 1 セル伝播する信号を速度  $\frac{1}{t}$  の信号という。

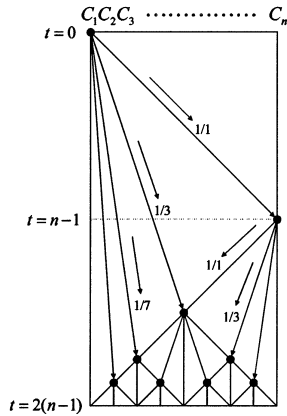


図1 最小時間アルゴリズムの時間空間図式

スプレイ上の数字パターンの一斉射撃問題を拡張し、16セグメントディスプレイ上の英字パターンの一斉射撃問題を新しく提案する。

### 3.1 16セグメントディスプレイ上の英字パターン

16セグメントディスプレイとは図2に示すようなディスプレイである。名前が示すとおり16個のセグメントと呼ばれる素子で構成されていて、それぞれをオン、オフすることによって英字や数字を表示することができる。



図2 16セグメントディスプレイ

AからZまでの英字は、16セグメントディスプレイ上にそれぞれ図3に示すようなパターンとして表示される。

### 3.2 モデル化

図2に示した16セグメントディスプレイを、図4に示すように2次元セル空間上でモデル化する。

16のセグメントはそれぞれn個のセルからなるものとする。ただし、nは任意であるが、有限とする。これらセグメントを構成するセルをセグメントセルと呼ぶ。また、各セグメントをつなぐセルとして9個のセルを置く。図4では網かけして示してある。これら9個のセルを特にジョイントセルと呼び、左の列の上からそれぞれ $C_A$ 、 $C_B$ 、 $C_C$ 、中央の列の上から $C_D$ 、 $C_E$ 、 $C_F$ 、右の列の上から $C_G$ 、 $C_H$ 、 $C_I$ とする。

図3に示した16セグメントディスプレイ上の英字パターンを、それぞれ図4に示したモデル化されたディスプレイ

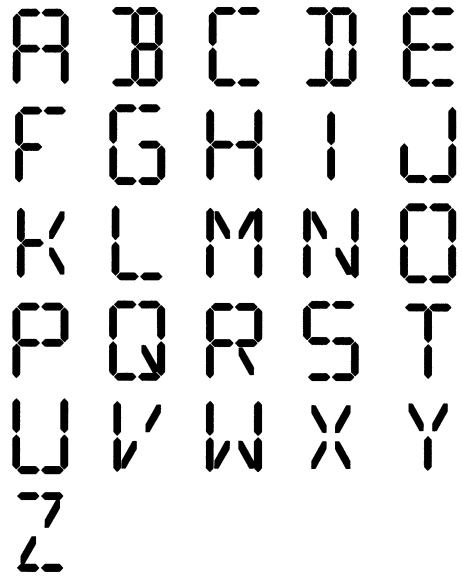


図3 16セグメントディスプレイ上の英字パターン

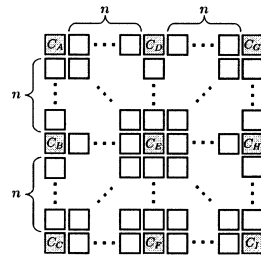


図4 16セグメントディスプレイのモデル化

ディスプレイ上に配置し、筆順の1番初めの位置にあるジョイントセルを将軍として、16セグメントディスプレイ上の英字パターンの一斉射撃問題を9近傍2次元セルオートマトン上の問題として定義する。

図5に各英字パターンの初期様相を示した。将軍は、英字パターンA, B, D, I, O, Q, Tについてはジョイントセル $C_D$ 、英字パターンC, G, J, Sについてはジョイントセル $C_G$ 、英字パターンE, F, H, K, L, M, N, P, R, U, V, W, X, Y, Zについてはジョイントセル $C_A$ で、図5では黒く塗りつぶして示してある<sup>(注3)</sup>。

図には示していないが、パターン以外の2次元セル空間上のセルは全て特定の状態 $S$ にある。すなわち、パターンは状態 $S$ のセルで境界を囲われていることに注意されたい。

## 4. 一斉射撃アルゴリズム

本章では、前章で定義した英字パターンの一斉射撃問題

(注3)：中学校で一般的に教えられている筆順を採用した。

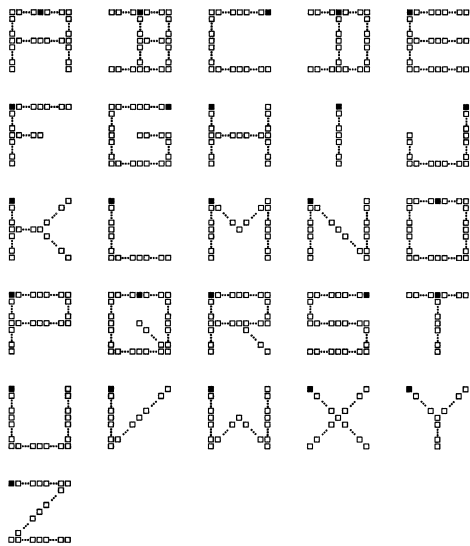


図5 英字パターンの初期様相

に対する一斉射撃アルゴリズムを与える。

初めに、図5に示した英字パターンの初期様相のジョイントセルに着目する。

ジョイントセルには、近傍の状態から自分がジョイントセルであることがわかるものと、セグメントセルと同じ近傍の状態を持つため自分がジョイントセルであることがわからないものがある。

例えば、英字パターンAを考える。ジョイントセル $C_A$ は、その上と左にあるセルは境界を示す状態 $S$ である。またセグメントセルは、上下もしくは左右のセルが境界を示す状態 $S$ である。よって、 $C_A$ は近傍の状態から自分がジョイントセルであることがわかる。しかし、ジョイントセル $C_D$ 、 $C_E$ は、その上下のセルが境界を示す状態 $S$ であり、セグメントセルと同じ近傍の状態を持つので、近傍の状態のみでは自分がジョイントセルであることがわからない。

この違いは、アルゴリズムを構成する上で大きな違いとなる。

英字パターン26個のうち、C, E, F, G, H, I, J, K, L, M, N, S, T, U, V, W, X, Y, Zの19パターンはいわゆる木である。

まず、英字パターンK, X, Yの将軍をそれぞれ手で持って、ぶらぶらと振ったときのイメージで表示すると、図6に示すようになる。レベル $i$  ( $0 \leq i \leq 3$ )にある節点(図では丸で描いている)はジョイントセルで特にレベル0にある節点は将軍である。枝を直線で描いているが、それぞれセグメントに対応し、 $n$ 個のセグメントセルからなっているものとする。

また、英字パターンC, I, L, N, U, V, Zについて同

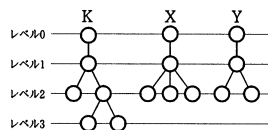


図6 英字パターンK, X, Yに対する木

様のことをすると、図7に示すようになる。偶数レベル $2i$  ( $0 \leq i \leq 3$ )にある節点は、近傍の状態から自分がジョイントセルであることがわかるジョイントセルである。また、近傍の状態から自分がジョイントセルであることがわからないジョイントセルは枝に含まれるものとする。よって、全ての枝は $2n+1$ 個のセルからなっている。

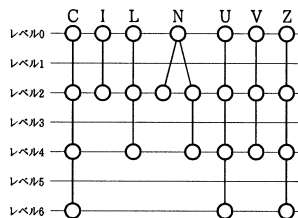


図7 英字パターンC, I, L, N, U, V, Zに対する木

同様に、英字パターンE, F, G, H, J, M, S, T, Wは図8に示すようになる。節点は、近傍の状態から自分がジョイントセルであることがわかるジョイントセルである。また、近傍の状態から自分がジョイントセルであることがわからないジョイントセルは枝に含まれるものとする。よって、全ての枝は $n$ 個もしくは $2n+1$ 個のセルからなっている。

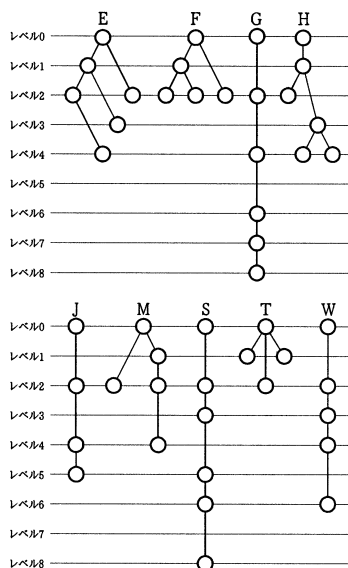


図8 英字パターンE, F, G, H, J, M, S, T, Wに対する木

一方、残り7個の英字パターン A, B, D, O, P, Q, R は木ではなくグラフである。

まず、英字パターン A, B は図9のようなグラフになる。ただし、英字パターン A のジョイントセル  $C_E$  は、近傍の状態から自分がジョイントセルであることがわからない。しかし、将軍  $C_D$  から送出された速度  $\frac{1}{3}$  の信号が  $C_A, C_B$  を経由する径路と  $C_G, C_H$  を経由する径路の2つの径路から、 $3(n+1)$  ステップ後にジョイントセル  $C_E$  に同時に到着する。このことによって、 $C_E$  は自分がジョイントセルであることがわかる。この時間は、将軍から送出された速度  $\frac{1}{3}$  の信号がレベル3の節点に到達するのに必要なステップ数と同じであるので、 $C_E$  をレベル3にある節点として考えることにする。

図9のジョイントセルのうち3つのセル（直線で半分にしたように描いてある節点）を半分に割ったと考えて、それぞれ2つにすると、図10のように木となる。全ての枝は  $n$  個のセルからなっている。半分に割ったジョイントセルには、将軍との信号（情報）のやりとりをする同じ長さの径路が複数存在するので、1つでも十分であるという意味で分割する。

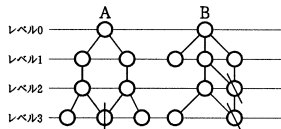


図9 英字パターン A, B に対するグラフ

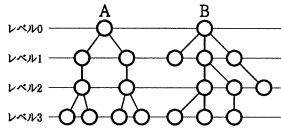


図10 英字パターン A, B に対する木

英字パターン D, O, P, Q, R も木ではなく、図11に示すようなグラフである。ただし、英字パターン O, Q のジョイントセル  $C_F$  は近傍の状態から自分がジョイントセルであることがわからない。しかし、将軍  $C_D$  から送出された速度  $\frac{1}{4}$  の信号が  $C_A, C_G$  を経由する径路と、 $C_G, C_I$  を経由する径路の2つの径路から、 $4(n+1)$  ステップ後にジョイントセル  $C_F$  に同時に到着する。このことによって、 $C_F$  は自分がジョイントセルであることがわかる。この時間は、将軍から送出された速度  $\frac{1}{4}$  の信号がレベル4の節点に到達するのに必要なステップ数と同じであるので、 $C_F$  をレベル4にある節点として考えることにする。

英字パターン A, B と同様に、図11のジョイントセルのうちの5つのセル（直線で半分にしたように描いてある節点）を、半分に割ったと考えてそれぞれ2つにする

と、図12に示すような木となる。全ての枝は  $n$  個もしくは  $2n+1$  個のセルからなっている。

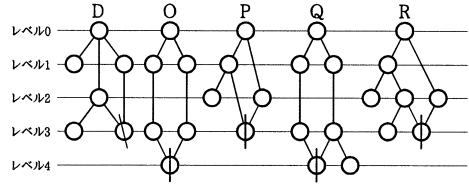


図11 英字パターン D, O, P, Q, R に対するグラフ

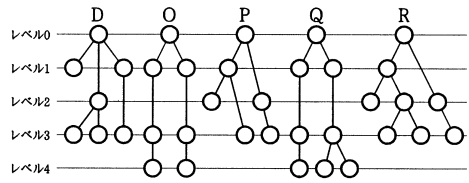


図12 英字パターン D, O, P, Q, R に対する木

以上により、全ての英字パターンを木として表すことができることがわかった。

図6と図10に示すような木で表される英字パターン A, B, K, X, Y の一斉射撃問題は、文献 [17,18] で定義されるセグメント木の斉射撃問題として一般化することができる。

セグメント木とは、以下の条件を満たす木である。

- 枝は1つのセグメント ( $n$  個のセル) からなる。
- 節点も1つのセルであるが、セグメントセルと自分を区別できる。

根から節点に至る径路にあるセグメント（枝）の数をその節点のレベルという。節点のレベルの中で一番大きいレベルをセグメント木の高さという。

セグメント木の根を将軍とし、全てのセルを一斉に射撃状態に遷移させる問題を、セグメント木の斉射撃問題とする<sup>(注4)</sup>。

文献 [17,18] には、高さ  $h$  のセグメント木を  $(h+1)(n+1)$  ステップで一斉射撃させるアルゴリズムが示してある。

このアルゴリズムを用いれば、英字パターン A, B, K, X, Y の斉射撃問題を解くことができる。

また、図7に示すような木は、枝が  $2n+1$  個のセルからなるセグメント木であると考えられるので、英字パターン C, I, L, N, U, V, Z の斉射撃問題も、同様にセグメント木の斉射撃アルゴリズムを用いて解くことができる。

表1に、セグメント木の斉射撃アルゴリズムを用いて解くことができる英字パターンの射撃時間をまとめた。

(注4)：もはや2次元セルオートマトン上で定義される斉射撃問題ではないことに注意されたい。

表 1 セグメント木の斉射撃アルゴリズムを用いて解ける英字パターンとその射撃時間

英字パターン	セグメント木		射撃時間
	高さ $h$	枝の長さ	
A	3	$n$	$4(n+1)$
B	3	$n$	$4(n+1)$
C	3	$2n+1$	$8(n+1)$
I	1	$2n+1$	$4(n+1)$
K	3	$n$	$4(n+1)$
L	2	$2n+1$	$6(n+1)$
N	2	$2n+1$	$6(n+1)$
U	3	$2n+1$	$8(n+1)$
V	2	$2n+1$	$6(n+1)$
X	2	$n$	$3(n+1)$
Y	2	$n$	$3(n+1)$
Z	3	$2n+1$	$8(n+1)$

英字パターン I は長さ  $2n+3$  の 1 次元セル列, 英字パターン L, V は長さ  $4n+5$  の 1 次元セル列, 英字パターン C, U, Z は長さ  $6n+7$  の 1 次元セル列であるから, 元々の (オリジナルの) 斉射撃問題と考えると最小時間アルゴリズムを適用すると, それぞれ  $4(n+1)$ ,  $8(n+1)$ ,  $12(n+1)$  ステップで解けることになる。

英字パターン I についてはどちらも同じ時間であるが, 英字パターン C, L, U, V, Z については, セグメント木として文献 [17, 18] のアルゴリズムを適用した方が元々の斉射撃問題として最小時間アルゴリズムを適用したより高速である。当然のことながら, これは自分がジョイントセルであることがわかるセルの働きによるものである。

一方, 図 8 と図 12 に示すような木で表される英字パターン D, E, F, G, H, J, M, O, P, Q, R, S, T, W の斉射撃問題は, 以下に示す 2-セグメント木の斉射撃問題として一般化することができる。

2-セグメント木とは, 以下の条件を満たす木である。

- 枝は 1 つのセグメント ( $n$  個のセル) または 2 つのセグメント ( $2n+1$  個のセル) からなる。
- 1 つのセグメントからなる枝と 2 つのセグメントからなる枝がそれぞれ少なくとも 1 つは含まれる。
- 節点も 1 つのセルであるが, セグメントセルと自分を区別できる。

根から節点に至る径路にあるセグメントの数をその節点のレベルという。節点のレベルの中で一番大きいレベルを 2-セグメント木の高さという。ただし,  $n$  個のセルからなる枝は 1,  $2n+1$  個のセルからなる枝は 2 と数えるものとする。

2-セグメント木の根を将軍とし, 全てのセルを斉に射撃状態に遷移させる問題を, 2-セグメント木の斉射撃問題とする。

高さ  $h$  の 2-セグメント木の斉射撃問題に対する斉射撃アルゴリズムを以下に示す。ただし,  $n$  個のセルからなる枝をその下に持つ節点の中でレベルの一番小さい節点のレベルを  $k$  とする<sup>(注5)</sup>。

### 【アルゴリズム】

(1) 時刻  $t=0$  にレベル 0 の節点 (将軍) は, 自分とつながっている全ての枝に第 1 信号<sup>(注6)</sup>を送出する。

(2)  $i=1$  から  $h+\frac{3}{2}k+1$  まで, 次のことを繰り返す。第  $i$  信号を受け取った節点は状態  $i$  となり, 自分とつながっている全ての枝 (親と子の両方向) に第  $i+1$  信号を送出する。第  $i$  信号, 第  $j$  ( $\neq i$ ) 信号を同時に受け取った場合, 節点は状態  $\max(i, j)$  となり, 自分とつながっている全ての枝に第  $\max(i, j)+1$  信号を送出する。また, 枝に含まれるジョイントセル ( $2n+1$  個のセルからなる枝の真ん中にあるジョイントセル) も第  $i$  信号と第  $j$  ( $\neq i$ ) 信号を同時に受け取る場合がある。このとき, このジョイントセルは自分がジョイントセルであることを認識し, 節点と同じ動作をする (以後, このジョイントセルも節点という)。すなわち, 状態  $\max(i, j)$  となり, 自分とつながっている全ての枝に第  $\max(i, j)+1$  信号を送出する。

(3) 時刻  $t=(h+2k+1)(n+1)$  にレベル  $h-1$  の節点は第  $h+\frac{3}{2}k+1$  信号を受け取り, 将軍状態となる。同時に第  $h+\frac{3}{2}k+1$  信号を受け取った他の節点も将軍状態となる。

(4) 将軍状態となった節点は, 自分とつながっている全ての枝をそれぞれ長さ  $n+2$  のセル列と考えて, 最小時間アルゴリズムを適用する<sup>(注7)</sup>。

このアルゴリズムについて以下のことがいえる。

[補題 1] 時刻  $t=(h+2k+1)(n+1)$  には, 枝に含まれるジョイントセルは全て自分がジョイントセルであることを認識し, 節点と同じ動作をしている。

(証明) レベル  $k$  の節点のうち  $n$  個のセルからなる枝をその下に持つ節点を  $v_k$  とし<sup>(注8)</sup>, その枝のもう一方の節点 (レベル  $k+1$  の節点) を  $v_{k+1}$  とする。また, 将軍を  $v_0$  とし,  $v_0$  から  $v_k$  に至る径路上にあるレベル  $2, 4, \dots, k-2$  の節点をそれぞれ  $v_2, v_4, \dots, v_{k-2}$  とする。

時刻  $t=0$  に将軍  $v_0$  が送出した第 1 信号は, 時刻

(注5): 当然のことながら,  $k$  は偶数であることに注意されたい。

(注6): 第  $i$  信号 ( $1 \leq i \leq h+\frac{3}{2}k+1$ ) は, 信号を受け取った節点を状態  $i$  に遷移させる速度  $\frac{1}{2}$  の信号である。ただし, 第  $h+\frac{3}{2}k+1$  信号は, 受け取った節点を将軍状態に遷移させる。

(注7): 時刻  $t=(h+2k+1)(n+1)$  に第  $h+\frac{3}{2}k+1$  信号を受け取らなかった節点は, 時刻  $t=(h+2k+2)(n+1)$  に射撃信号を受け取り, 射撃信号を返す。すなわち, 将軍とは反対側の端にあるセルの役割を果たす。

(注8): 複数個存在する場合もあるが, その場合にはそのうちの 1 つを  $v_k$  とする。

$t = 2(n+1)$  にレベル 2 の節点に到達する。レベル 2 の各節点は状態 1 となり、自分とつながっている全ての枝（親と子の両方向）に第 2 信号を送出する。時刻  $t = 4(n+1)$  に將軍  $v_0$  及びレベル 4 の節点に第 2 信号が到達し、各節点は状態 2 となり、第 3 信号を送出する。

このように時刻  $t = 0$  から  $k(n+1)$  まで、レベル  $0, 4, 8, \dots$  の節点とレベル  $2, 6, 10, \dots$  の節点は、交互に信号の送信側と受信側となって、状態の値、信号の値を増やしながら同期して信号のやり取りを行う。

時刻  $t = k(n+1)$  にレベル  $k$  の節点  $v_k$  は第  $\frac{k}{2}$  信号を受け取って状態  $\frac{k}{2}$  となり、自分とつながっている全ての枝に第  $\frac{k}{2} + 1$  信号を送出する。

時刻  $t = (k+1)(n+1)$  にレベル  $k+1$  の節点  $v_{k+1}$  は、 $v_k$  から送出された第  $\frac{k}{2} + 1$  信号を受け取って状態  $\frac{k}{2} + 1$  となり、自分とつながっている全ての枝に第  $\frac{k}{2} + 2$  信号を送出する。

時刻  $t = (k+2)(n+1)$  に節点  $v_k$  は、 $v_{k+1}$  から送出された第  $\frac{k}{2} + 2$  信号を受け取って状態  $\frac{k}{2} + 2$  となり、自分とつながっている全ての枝に第  $\frac{k}{2} + 3$  信号を送出する。このとき、レベル  $k-2$  の節点  $v_{k-2}$  は第  $\frac{k}{2} + 1$  信号を受け取って状態  $\frac{k}{2} + 1$  となり、自分とつながっている全ての枝に第  $\frac{k}{2} + 2$  信号を送出する。 $v_k$  は  $n$  個のセルからなる枝をその下に持っているの、送信側になるのがこれまでより  $2(n+1)$  だけ早くなり、そのため  $v_k$  と  $v_{k-2}$  が同時に送信側になる。

時刻  $t = (k+3)(n+1)$  に  $v_k$  と  $v_{k-2}$  を結ぶ枝の真ん中にあるセルは、 $v_k$  から送出された第  $\frac{k}{2} + 3$  信号と  $v_{k-2}$  から送出された第  $\frac{k}{2} + 2$  信号を同時に受け取る。このことによって、このセルは自分がジョイントセルであることを認識し、状態  $\frac{k}{2} + 3$  となり、自分とつながっている全ての枝に第  $\frac{k}{2} + 4$  信号を送出する。

$v_k$  から始まるタイミングのずれは根まで続く。根に向かって  $v_{k-2}$  と  $v_{k-4}, \dots, v_2$  と  $v_0$  を結ぶ  $2n+1$  個のセルからなる枝を、同様に 2 つの  $n$  個のセルからなる枝と節点に分割しながら信号を伝える。

時刻  $t = (2k+2)(n+1)$  に將軍  $v_0$  は第  $\frac{3}{2}k+2$  信号を受け取って、状態  $\frac{3}{2}k+2$  となる。このときも信号のずれは続いている。

したがって、次に全ての葉に向かってレベル  $h-1$  まで信号を伝達することで、2-セグメント木に含まれる全ての  $2n+1$  個のセルからなる枝を分割することができる。つまり、時刻  $t = (h+2k+1)(n+1)$  には、枝に含まれるジョイントセルは全て自分がジョイントセルであることを認識し、節点と同じ動作をしている。□

**[補題 2]** 時刻  $t = (h+2k+1)(n+1)$  にレベル  $h-1$  の節点が第  $h + \frac{3}{2}k + 1$  信号を受け取って將軍状態となった

とき、レベル  $h-3, h-5, \dots$  の節点も同時に全て將軍状態となる。

(証明) 時刻  $t = (k+1)(n+1)$  にレベル  $k+1$  の節点  $v_{k+1}$  は第  $\frac{k}{2} + 1$  信号を受け取って状態  $\frac{k}{2} + 1$  となる。このとき、 $v_{k+1}$  の状態  $\frac{k}{2} + 1$  はこの木の全ての節点の中で最大の値となっている。さらに、時刻  $t = (2k+2)(n+1)$  に將軍  $v_0$  は第  $\frac{3}{2}k+2$  信号を受け取って状態  $\frac{3}{2}k+2$  となる。この將軍の状態  $\frac{3}{2}k+2$  はこの木の全ての節点の中で最大の値となっている。したがって、その直後に將軍が葉に向かって送出する第  $\frac{3}{2}k+3$  信号は同時刻において最大の信号である。そして、この信号が葉に向かって  $2n+1$  個のセルからなる枝を分割すると同時に、偶数レベルの節点と奇数レベルの節点の同期をとる。

したがって、時刻  $t = (h+2k+1)(n+1)$  にレベル  $h-1$  の節点が第  $h + \frac{3}{2}k + 1$  信号を受け取って將軍状態となったとき、レベル  $h-3, h-5, \dots$  の節点も同時に將軍状態となる。□

**[定理 1]** アルゴリズムは時刻  $t = (h+2k+3)(n+1)$  で一斉射撃状態に遷移する。

(証明) 補題 1, 補題 2 より、時刻  $t = (h+2k+1)(n+1)$  には、枝に含まれるジョイントセルは全て自分がジョイントセルであることを認識し、またレベル  $h-1, h-3, h-5, \dots$  の節点が全て將軍状態となっている。したがって、將軍状態となっている節点を將軍とした  $n+2$  個の 1 次元セル列の一斉射撃問題に分割することができる<sup>(注9)</sup>。それぞれに最小時間アルゴリズムを用いると  $n+2$  個の 1 次元セル列の一斉射撃問題は  $2(n+1)$  ステップで射撃状態に遷移させることができる。よって、このアルゴリズムは時刻  $t = (h+2k+3)(n+1)$  で一斉射撃状態に遷移する。□

このアルゴリズムは、 $k = h-1$  のときに最も時間がかかる。図 13 は、この最悪の場合の 2-セグメント木が状態遷移する過程を上から下へと時系列順に並べたものである。横軸はセル空間で、 $N_l$  ( $0 \leq l \leq h$ ) はレベル  $l$  の全ての節点を表している。縦軸は時間を表している。黒い点はそのセルが將軍状態になったことを示している。⊕ は枝に含まれるジョイントセルが自分がジョイントセルであることを認識し、節点と同じ動作をすることを示している。

表 2 に、2-セグメント木の一斉射撃アルゴリズムを用いて解くことのできる英字パターンの射撃時間をまとめた。

## 5. あとがき

本稿では山下等 [17, 18] の定義した 7 セグメントディスプレイ上の数字パターンの一斉射撃問題を拡張し、16 セ

(注9)：この時、將軍状態になっていないレベル  $h, h-2, h-4, \dots$  の節点は壁の役割をすることに注意されたい。

## 文 献

- [1] E. F. Moore, "The firing squad synchronization problem," in *Sequential Machines, Selected Papers* (E. F. Moore ed.), Addison-Wesley, Reading MA., pp.213–214, 1964.
- [2] M. Minsky and J. McCarthy, "Computation : Finite and infinite machines," Prentice Hall, pp.28–29, 1967.
- [3] E. Goto, "A minimal-time solution of the firing squad synchronization problem," *Course Notes for Applied Mathematics 298*, Harvard University, pp.52–59, 1962.
- [4] A. Waksman, "An optimum solution to the firing squad synchronization problem," *Information and Control*, 9, pp.66–78, 1966.
- [5] R. Balzer, "An 8-state minimal time solution to the firing squad synchronization problem," *Information and Control*, 10, pp.22–42, 1967.
- [6] H. D. Gerken, "Über Synchronisations-Probleme bei Zellularautomaten," *Diplomarbeit, Institut für Theoretische Informatik, Technische Universität Braunschweig*, 1-50, 1987.
- [7] J. Mazoyer, "A six-state minimal time solution to the firing squad synchronization problem," *Theoretical Computer Science*, 50, pp.183–238, 1987.
- [8] K. Culik, "Variations of the firing squad problem and applications," *Information Processing Letters*, 30, pp.153–157, 1989.
- [9] I. Shinahr, "Two and three-dimensional firing squad synchronization problems," *Information and Control*, 24, pp.163–180, 1974.
- [10] K. Kobayashi, "The firing squad synchronization problem for two dimensional arrays," *Information and Control*, 34, pp.153–157, 1977.
- [11] Z. Roka, "The firing squad synchronization problem on Cayley graphs," *Proc. of MFCS'95, Prague (Czech Republic)*, 1995. *Lecture Notes in Computer Science*, 969, pp.402–411, 1995.
- [12] Y. Nishitani and N. Honda, "The firing squad synchronization problem for graphs," *Theoretical Computer Science*, 14, pp.39–61, 1981.
- [13] K. Imai and K. Morita, "Firing squad synchronization problem in reversible cellular automata," *Theoretical Computer Science*, 165, pp.475–482, 1996.
- [14] K. Imai, K. Morita, and K. Sako, "Firing squad synchronization problem in number-conserving cellular automata," *Proc. of the IFIP Workshop on Cellular Automata, Santiago (Chile)*, 1998.
- [15] J. Mazoyer, "On optimal solutions to the firing squad synchronization problem," *Theoretical Computer Science*, 168 (2), pp.367–404, 1996.
- [16] H. Umeo, "A design of cellular algorithms for 1-bit inter-cell communications and related cellular algorithms," *Proc. of MCU'98*, 1, pp.210–227, 1998.
- [17] 山下, 竹正, 酒井, 広瀬, "7セグメントディスプレイ上の数字パターンの一斉射撃問題," *信学技報, COMP2007-28*, pp.79–84, 2007.
- [18] K. Yamashita, E. Takemasa, M. Sakai, and S. Hirose, "The firing squad synchronization problems for number patterns on seven-segment display," submitted for publication.

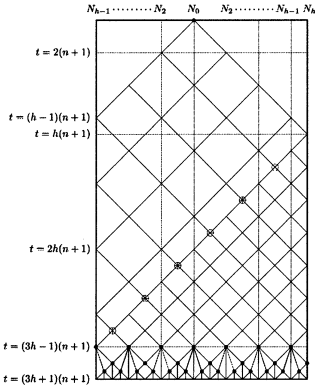


図 13 2-セグメント木に対する一斉射撃アルゴリズムの時間空間図式

表 2 2-セグメント木の一斉射撃アルゴリズムを用いて解ける英字パターンとその射撃時間

英字 パターン	2-セグメント木		射撃時間
	高さ $h$	$k$	
D	3	0	$6(n+1)$
E	4	0	$7(n+1)$
F	2	0	$5(n+1)$
G	8	6	$23(n+1)$
H	4	0	$7(n+1)$
J	5	4	$16(n+1)$
M	4	0	$7(n+1)$
O	4	0	$7(n+1)$
P	3	0	$6(n+1)$
Q	4	0	$7(n+1)$
R	3	0	$6(n+1)$
S	8	2	$15(n+1)$
T	2	0	$5(n+1)$
W	6	2	$13(n+1)$

グメントディスプレイ上の英字パターンの一斉射撃問題を新しく提案した。また、山下等 [17, 18] の定義したセグメント木を拡張した 2-セグメント木を定義し、2-セグメント木の一斉射撃問題を新しく提案した。

高さ  $h$  で、 $n$  個のセルからなる枝をその下に持つ節点の中でレベルが一番小さい節点のレベルが  $k$  である 2-セグメント木の一斉射撃問題に対して、 $(h+2k+3)(n+1)$  ステップの一斉射撃アルゴリズムを示した。

また、山下等 [17, 18] に示されているセグメント木の一斉射撃アルゴリズム、及び本稿で示した 2-セグメント木の一斉射撃アルゴリズムを用いて、全ての英字パターンの一斉射撃問題が解けることを示した。

本稿で扱ったそれぞれの問題に対して、さらに高速なアルゴリズムが存在するか、またそれらの問題を解く最小時間を解明することが今後の課題である。