ヒューマンインタフェース　101－6
（２００２．　１１．　２３）

巡回パズルのメディア変換とパズル・ジェネレータの試作

前田篤彦[1]・杉山公造[1]・間瀬健二[2]

玩具が持つ豊かな世界をインタフェースに応用するための系統的なアプローチの一環として、ライツアウト、ルービック・クロックなどの"巡回パズル"を一般的に表す抽象モデルを作成し、新しい表現メディア（グラフ表現、ボックス表現、サウンド表現など）への変換方式を考案し、巡回パズルのジェネレータを試作した。このジェネレータにより巡回パズルの多くのバリエーションを生成することを通して、メディア変換の得失についての知見をまとめた。また、置換パズルと巡回パズルへの適用結果に関し、全体的考察を行う。

Media Conversion of Cyclic Puzzles and Development of Cyclic Puzzle Generators

Atsuhiko Maeda[1], Kozo Sugiyama[1], Kenji Mase[2]

As a systematic approach to utilize toy worlds for human interfaces, “cyclic puzzles” such as Lights-out, Rubik’s clock are expressed as an abstract model. Then, new media called graph media and sound media is devised for converting the puzzles on the media and a cyclic puzzle generator is developed. Merits and demerits in introducing the new medial obtained through generating variations of the puzzles are summarized.

1．　はじめに

前稿[1]では、"置換パズル"に関するメディア変換とパズル・ジェネレータの試作を報告した。本稿では、"巡回パズル"に関する同様なアプローチを試みた結果を報告する。前稿と本稿で目指すのは、遊び行為に利用されている多様な玩具の性質を応用するための基礎的かつ系統的なアプローチを開発することにある。そのために数ある玩具の中から操作パズルを取り上げ、新しいアプローチを試みる。操作パズルを取り上げたのは、手に取って遊べる玩具として広く流布しており、様々な種類のものが存在することの他に、数理的な概念と相性がよく一般性があり、応用範囲が広いなど、我々のアプローチの特徴を示すのに適しているからである。新しいアプローチの考え方の骨子を図１に示す。その具体的な手順の概略は次のとおりである。

(1)　既存の巡回パズルを分析し、それらをまとめて表現できるような数理モデルを考案する。

(2)　数理的構造を保存したまま、表現メディアを変換したり、パラメータの値を変えたりして、パズルの具現化を行い、既存パズルにない新しい効果の発現やパズルのバリエーションを求める仕組みを考察する。パズルの具現化において次の二つを区別する。

(a) **パラメトリックな具現化**

(b) **創造的な具現化**

(3)　新しいメディアでのパラメトリックなパズル・ジェネレータを開発する。

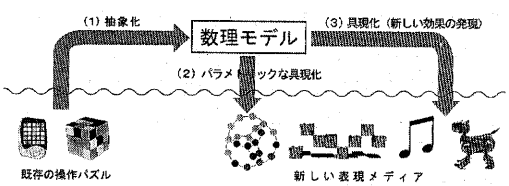

図１. 本稿でのアプローチ

以下において、２節では、巡回パズルを分析し抽象モデルを作成する。３節では、新たな表現メディアを提案し、パズル・ジェネレータの試作について述べ、表現メ

1北陸先端科学技術大学院大学, JAIST, {a_maeda, sugi}@jaist.ac.jp
2 名古屋大学, Univ. of Nagoya, mase@itc.nagoya-u.ac.jp

ディアの有効性や利点を検証する。4 節では本研究のまとめと今後の研究課題について述べる。

2. 巡回パズル

我々が巡回パズルとして分類するものに、ライツアウト(LO)やルービック・クロック(RCl)などがある(図 2 参照)。これらのパズルでは、同様な操作を何回か繰り返すと元の状態の戻るという巡回する性質を持っている。LO は 2 回押すと、RCl は 12 時間時計を進めると元に戻る。

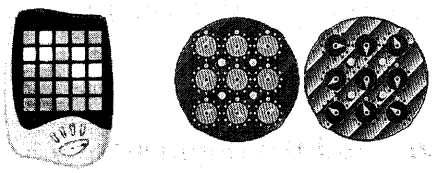

図 2. ライツアウト(左)とルービック・クロック(右)

LO は、押すことを繰り返すとライトが点いたり消えたりする 5 行 5 列の電光パネル・ボタンから構成されている。初期状態として与えられた電光パネルのライトの点灯パターンから、ボタンをうまく押すことにより、全てのライトが消えた状態を求める操作パズルである。このタスクは簡単には達成できない。それは、特定のライトを消したり点けたりしようとすると、そのライトの上下左右の隣接するライトが、反転する仕掛けになっているからである。同じボタンを続けて押すと状態は元に戻るので、各操作は位数 2 の巡回構造を持つ。

RCl は、本体の表と裏にそれぞれ 9 つずつ並んでいる、針一つの時計から構成されている。ある時計を操作すると、予め指定されたいくつかの時計も同時に予め指定された方向に同じ時間幅だけ動くようになっている。このようなある操作の他の時計への影響の仕方は、図 2 の RCl の写真の中ほどにある 4 つの小さい白いボタンにより変更することができる。このパズルでは、それぞれの時計がバラバラの時刻を示している状態から、すべての時計が 12 時を指すように操作することがユーザに課せられたタスクである。同じ時計を 12 時間進めると全ての状態が元に戻るので、各操作は位数 12 の巡回構造を持つ。このパズルも解くのはかなり難しい。

ライツアウトの分析

いま、ボタン付き電光パネルを要素と考えると、LO の場合は 25 個の要素からなっている。これを要素集合 $N=\{1, 2, \ldots, 25\}$ とする。各要素には、ライトが点いている(ON)、消えている(OFF)の二つの状態がある。これを状態変数のベクトル $\boldsymbol{y}=[y_1\, y_2 \ldots y_{25}]$, $y_i \in \{1, 0\}$ とし、要素 i のライトが点いている場合 $y_i=1$、消えている場合 $y_i=0$ とする。

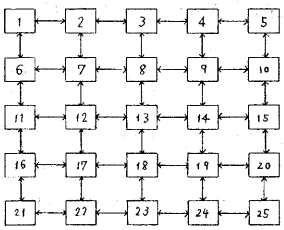

図 3. ライツアウトの要素の配置と影響関係

$$A=\begin{bmatrix}
1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&1&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&1&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&1&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&1&1&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&1&1&1&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1
\end{bmatrix}$$

図 4. ライツアウトの影響行列

操作は要素 i のボタンを押すという 25 の操作集合 $R=\{r_1, r_2, \ldots, r_n, \ldots, r_{25}\}$ に限られている。この操作により、状態変数が変化する。ボタン i を押したとき、どのライトを反転させるかはあらかじめ定められている。LO の場合は、物理的な配置に即して定義されており、図 3 に示したように上下左右のライトに影響を与える。図 3 はグラフ理論でいう平面グラフである。

図 3 に示した影響関係を影響行列 $A=[a_{ij}]$, $a_{ij} \in \{1, 2\}$ として図 4 に示す。$a_{ij}=1$ はボタン i を押したとき要素 j に影響し、0 のときは影響しないことを表す。自分自身には必ず影響するので対角成分 a_{ii} は全て 1 である。A の行ベクトル $\boldsymbol{\omega}_i=[a_{i1}\, a_{i2} \ldots a_{i25}]$ をみればボタン i を押したときにどの要素に影響を与えるか知ることができる。影響関係がユーザに明示的であるかどうかは、パズルの性格を決める重要な要素である。LO の場合は、矢印で示すなどして明示されているわけではないが、影響範囲が前後左右という分かりやすい構造になっているため、実際上明示されているといって問題はない。この点、後述する RCl では状況が異なっている。

ライツアウトの解法については良く調べられている[2, 3, 4]。各ボタンを 2 度押すと元に戻るので、解は各ボタ

ンを一度押すか押さないかで決まる。また、ボタンを押す順序は結果に影響しない。いま、解ベクトルを $x=[x_1 x_2 \ldots x_{25}]$, $x_i \in \{1, 0\}$、ボタン i を押す場合 $x_i=1$、押さない場合 $x_i=0$ と表し、排他的論理和(XOR)⊕と論理積を用いて表すと、次の式が成り立つ。

$$y_i = a_{1i}x_1 \oplus a_{2i}x_2 \oplus \cdots \oplus a_{25i}x_{25},\ i=1,\ldots,25$$

すなわち、連立一次方程式 $Ax=y$ となり、これを x について解く問題となる。本稿の目的は解法ではなくパズルのモデル化であるので、解法についてはここではこれ以上ふれないが、上の方程式が解を持つ必要十分条件や解法については文献[2]に詳しい。

いま、影響行列を A、状態を y とするとき、操作 r_i を行ない状態 y を更新することは、

$$y \leftarrow r_i(y) = y \oplus \omega_i = y + \omega_i \pmod 2$$

と書かれる。ここで mod2 は 2 で割った余りを求めることを意味する。

以上より、ライツアウトの抽象モデルを次のように構成することができる。

ライツアウトの抽象モデル：(y, qA, R, s, t)

(1) 要素の状態ベクトル：$y=[y_1, y_2, \ldots, y_{25}]$, $y_i \in \{1, 0\}$
(2) 巡回の位数あるいは要素の状態数：$q=2$
(3) 影響行列：$A=[a_{ij}]$, $i,j=1,\ldots,25$
影響関係はユーザに**明示的**である。
(4) 操作集合：$R=\{r_1, r_2, \ldots, r_{25}\}$
$y \leftarrow r_i(y) = y + \omega_i \pmod 2$
なお操作の**冗長度**は 2 であることが分かっている。
(5) 初期状態：s: 全てのライトが消えた状態から任意にボタンを何回か押した状態。
(6) 最終状態：t: 全てのライトが消えた状態

ルービック・クロックの分析

RCI の挙動は LO に比べるとかなり複雑である。しかし、基本的な仕組みは類似している[5]。次に示すのは、RCI におけるルールである。

(1) 直接操作して回せるのは、表と裏の四つ角のクロックのみ。
(2) 表の ON/OFF スイッチを ON にすると、裏で対応するスイッチが OFF になる。
(3) ON 状態スイッチに隣接する角の時計を回すと、その面にある全ての ON 状態スイッチの四方の時計が一緒に同方向に同じ角度だけ動く。また、裏側では全ての OFF 状態スイッチに隣接する角の時計が反対方向に同じ角度だけ動く。
(4) OFF 状態スイッチに隣接する角の時計を回すと、その面にある全ての OFF 状態スイッチに隣接する角の時計が同方向に同じ角度だけ動く。また、裏側では全ての ON 状態スイッチの四方の時計が一緒に反対方向に同じ角度だけ動く。

要素集合として、時計 $C=\{1, 2, \ldots, 18\}$ とスイッチ $D=\{1, 2, 3, 4\}$ とし、時計とスイッチの状態ベクトルをそれぞれ $y=[y_1 y_2 \ldots y_{18}]$、$y_i \in \{1, 2, \ldots, 12\}$ と $z=[z_1 z_2 z_3 z_4]$、$z_i \in \{0, 1\}$ とする。y_i は時計 i の針が指している時間であり、z_j は 0 のときスイッチ j が表側でオフ、1 のときはオンを意味する。図 5 に時計とスイッチの番号の割付を示す。

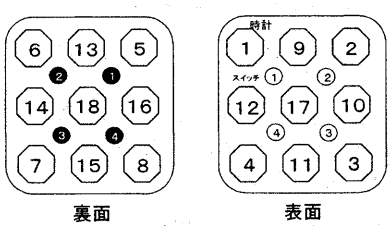

図 5. RCI における要素の配置と番号の割付

RCI におけるルールに、直接操作して回せるのは表と裏の四つ角のクロックのみなので、基本操作としては、1 から 8 までの時計を時計回りに 1 時間進めることをとればよい。2 時間回す場合には、この基本操作を 2 度行えばよいし、1 時間戻すには、この操作を 11 回繰り返せばよいからである。この基本操作を $R=\{r_1, r_2, \ldots, r_8\}$ とする。またスイッチを切り替える操作があるが、これは影響行列を切り替える機能としてとらえることとする。

影響行列(8 行×18 列)は 16 個用意しなければならない。すなわち、$A=\{A_0, A_1, \ldots, A_{15}\}$ であるが、このサフィックスは、4 つのスイッチの ON、OFF の状態を 10 進数に直したもので、その対応は次の通りである。すなわち、$[z_1 z_2 z_3 z_4]$ に対し次の番号をふる。すなわち、0:[0 0 0 0], 1:[1 0 0 0], 2:[0 1 0 0],…,3:[1 1 0 0], 15: [1 1 1 1]。

各々の行列において、行が基本操作に対応し、列が時計に対応する。影響行列の要素は-1、 0、 1 のいづれかを取る。-1 のときは影響先の時計の回転方向が操作とは反対である、0 のときは影響がない、1 のときは回転方向が同じことを意味する。影響行列の一部を図 6 に示す。-1 は m で表した。RCI において、影響関係がユーザに対して明示的に示されているだろうか。上述したルールが与えられているので、与えられていないとは言えないが、スイッチの切り替えなど複雑な機構を介在しているの

で、LO と比較するとユーザには殆ど明示的でないといっていいだろう。何度も操作している内に、少しずつ影響関係が飲み込めてくるというところであろうか。

$$A_0=\begin{bmatrix}
1&1&1&1&m&m&m&m&1&1&1&1&0&0&0&0&1&0\\
1&1&1&1&m&m&m&m&1&1&1&1&0&0&0&0&1&0\\
1&1&1&1&m&m&m&m&1&1&1&1&0&0&0&0&1&0\\
1&1&1&1&m&m&m&m&1&1&1&1&0&0&0&0&1&0\\
m&m&m&m&1&1&1&1&m&m&m&m&0&0&0&0&m&0\\
m&m&m&m&1&1&1&1&m&m&m&m&0&0&0&0&m&0\\
m&m&m&m&1&1&1&1&m&m&m&m&0&0&0&0&m&0\\
m&m&m&m&1&1&1&1&m&m&m&m&0&0&0&0&m&0
\end{bmatrix}$$

$$A_1=\begin{bmatrix}
1&0&0&0&m&0&0&0&0&0&0&0&m&0&0&m&0&m\\
0&1&1&1&0&m&m&m&1&1&1&1&0&0&0&0&1&0\\
0&1&1&1&0&m&m&m&1&1&1&1&0&0&0&0&1&0\\
0&1&1&1&0&m&m&m&1&1&1&1&0&0&0&0&1&0\\
m&0&0&0&1&0&0&0&0&0&0&0&1&0&0&1&0&1\\
0&m&m&m&0&1&1&1&m&m&m&m&0&0&0&0&m&0\\
0&m&m&m&0&1&1&1&m&m&m&m&0&0&0&0&m&0\\
0&m&m&m&0&1&1&1&m&m&m&m&0&0&0&0&m&0
\end{bmatrix}$$

$$A_2=\begin{bmatrix}
1&0&1&1&m&0&m&m&1&1&1&1&0&0&0&0&1&0\\
0&1&0&0&0&m&0&0&0&0&0&0&m&m&0&0&0&m\\
1&0&1&1&m&0&m&m&1&1&1&1&0&0&0&0&1&0\\
1&0&1&1&m&0&m&m&1&1&1&1&0&0&0&0&1&0\\
m&0&m&m&1&0&1&1&m&m&m&m&0&0&0&0&m&0\\
0&m&0&0&0&1&0&0&0&0&0&0&1&1&0&0&0&1\\
m&0&m&m&1&0&1&1&m&m&m&m&0&0&0&0&m&0\\
m&0&m&m&1&0&1&1&m&m&m&m&0&0&0&0&m&0
\end{bmatrix}$$

図6. 影響行列A_0、 A_1、 …、 A_{15}の一部

さて、今、スイッチの状態kに対応する影響行列をA_k、そのi番目の行ベクトルを$\boldsymbol{\omega}_i^k$で表すとき、状態Yを操作r_iを行い更新することは、

$$\boldsymbol{y}\leftarrow r_i(\boldsymbol{y})=\boldsymbol{y}+\boldsymbol{\omega}_i^k \pmod{12}$$

と書ける。ここで、影響行列を切り替えないで表現することもできる。影響行列を全て結合し128行×18列の1つの影響行列Aで表わすこともできるので、その行ベクトルを$\boldsymbol{\omega}_i$とすれば、上式は

$$\boldsymbol{y}\leftarrow r_i(\boldsymbol{y})=\boldsymbol{y}+\boldsymbol{\omega}_i \pmod{12}$$

となる。

以上より、RCl の抽象モデルは次のように構成される。

RCl の抽象モデル：$(\boldsymbol{y}, q, A, R, s, t)$

(1) 要素の状態ベクトル：$\boldsymbol{y}=[y_1, y_2, \ldots, y_{18}]$
(2) 巡回の位数あるいは要素の状態数：$q=12$
(3) 影響行列の集合：$A=\{A_0, A_2, \ldots, A_{15}\}$
影響関係はユーザに殆ど**明示的でない**。
(4) 操作集合：$R=\{r_1, r_2, \ldots, r_8\}$, スイッチ$k\in\{0,\ldots,15\}$

$$\boldsymbol{y}\leftarrow r_i(\boldsymbol{y})=\boldsymbol{y}+\boldsymbol{\omega}_i^k \pmod{12}$$

操作数は影響行列の切り替えを考慮すると全部で128あるが、次節で分析するように、実質必要なのは16の操作でよく、**冗長度**はなんと112である。
(5) 初期状態：s: 最終状態から任意の操作を何回か繰り返した状態。
(6) 最終状態：t: 全ての時計が12時を指した状態。

RCl の影響行列の分析 － 操作の冗長度を求める

影響行列A_0、 A_1、 …、 A_{15}を分析する。各行列は、行が表と裏の4つ角（スクエア）の時計を1時間進めるという操作に対応し、列は全ての時計に対応する。図7には、A_2を例として影響行列の構造を示している。

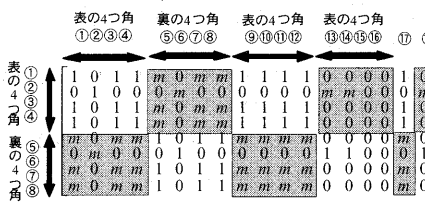

図7. 影響行列A_2の構造

全ての影響行列は、図7に見られるようにきれいな対称性を示している。つまり、表面（行列のうえ半分）と裏面（行列の下半分）は0の位置は同じであり、1とmの位置は入れ替わっている。これは、表の時計を進めたときと裏の時計を進めたときの時計の反応の仕方は、回転方向が異なるだけで動きの角度は全く同じであることを意味している。例えば、時計①を1時間進めることと時計⑤を一時間戻す（11時間進める）こととは同じである。逆に時計⑤を1時間進めることと時計①を1時間戻す（11時間進める）こととは同じである。また、表のスクエアの時計を進めたときの表のスクエアの時計の反応と裏のスクエアの時計の反応とが全く同じである（逆も成り立つ）。これは、①と⑤、②と⑥、③と⑦、④と⑧の時計の動きは常に向きだけが逆であることを意味する。従って、ゴールの時点で表のスクエアの時計が12時を指すときには、裏の時計も12時を指している。このことは、このパズルを解くためには表のスクエアの時計を見ていれば裏のスクエアの時計は見る必要がないことを意味する。

以上の考察から、図6に示した影響行列から冗長な要素を除去すると図8のような行列を得ることができる。図8の各行列において、行は①、②、③、④に、列は①、②、③、④、⑨、⑩、⑪、⑫、⑰、⑬、⑭、⑮、⑯、⑱にそれぞれ対応する。図8をみると、各行列において全く同じ値を持つ行が多くあり、この冗長さを取り除き簡単化すると図9のようになる。例えば、図9のA_0は、①、②、③、④のどの時計を動かしても同じ反応があることを意味している。

$$A_0=\begin{bmatrix}1\,1\,1\,1 & 1\,1\,1\,1\,1 & 0\,0\,0\,0\,0\\ 1\,1\,1\,1 & 1\,1\,1\,1\,1 & 0\,0\,0\,0\,0\\ 1\,1\,1\,1 & 1\,1\,1\,1\,1 & 0\,0\,0\,0\,0\\ 1\,1\,1\,1 & 1\,1\,1\,1\,1 & 0\,0\,0\,0\,0\end{bmatrix}$$

$$A_1=\begin{bmatrix}1\,0\,0\,0 & 0\,0\,0\,0\,0 & m\,0\,0\,m\,m\\ 0\,1\,1\,1 & 1\,1\,1\,1\,1 & 0\,0\,0\,0\,0\\ 0\,1\,1\,1 & 1\,1\,1\,1\,1 & 0\,0\,0\,0\,0\\ 0\,1\,1\,1 & 1\,1\,1\,1\,1 & 0\,0\,0\,0\,0\end{bmatrix}$$

$$A_2=\begin{bmatrix}1\,0\,1\,1 & 1\,1\,1\,1\,1 & 0\,0\,0\,0\,0\\ 0\,1\,0\,0 & 0\,0\,0\,0\,0 & m\,m\,0\,0\,m\\ 1\,0\,1\,1 & 1\,1\,1\,1\,1 & 0\,0\,0\,0\,0\\ 1\,0\,1\,1 & 1\,1\,1\,1\,1 & 0\,0\,0\,0\,0\end{bmatrix}$$

図8. 冗長さを除去した影響行列の一部

$$A_0=[1\,1\,1\,1 \quad 1\,1\,1\,1\,1 \quad 0\,0\,0\,0\,0]\begin{matrix}①\\②\\③\\④\end{matrix}$$

$$A_1=\begin{bmatrix}1\,0\,0\,0 & 0\,0\,0\,0\,0 & m\,0\,0\,m\,m\\ 0\,1\,1\,1 & 1\,1\,1\,1\,1 & 0\,0\,0\,0\,0\end{bmatrix}\begin{matrix}①\\②\\③\\④\end{matrix}$$

$$A_2=\begin{bmatrix}1\,0\,1\,1 & 1\,1\,1\,1\,1 & 0\,0\,0\,0\,0\\ 0\,1\,0\,0 & 0\,0\,0\,0\,0 & m\,m\,0\,0\,m\end{bmatrix}\begin{matrix}①\\③\\④\\②\end{matrix}$$

図9. さらに冗長さを除去した影響行列の一部

結局、図9は時計を動かす操作パターンに30種類の手があることを意味している。これを $P=\{p_1,\ldots,p_h,\ldots,p_{30}\}$ とする。P に含まれる各々の操作に意味解釈を加え、RCIの解法を手順化したものを図10に示す。操作数は影響行列の切り替えを考慮すると全部で128あるが、図10は、結局、16種類の操作があればルービック・クロックを解くことができることを意味している。従って、112の操作が冗長であることが分かる。この大きな**冗長度**もRCIというパズルの大きな特徴であることを認識しなければならない。

操作（表：＋1）	名称	①②③④	⑨⑩⑪⑫⑰	⑬⑭⑮⑯⑱	操　作
第一の操作【裏のダイアモンド⑬⑭⑮⑯の時間を揃える】					
●●○○③④	p_{23}	0 0 1 1	0 0 0 0 0	0 m m m m	⑬を＋1
○●●○④①	p_{18}	1 0 0 1	0 0 0 0 0	m 0 m m m	⑭を＋1
○○●●①②	p_6	1 1 0 0	0 0 0 0 0	m m 0 m m	⑮を＋1
●○○●②③	p_{13}	0 1 1 0	0 0 0 0 0	m m m 0 m	⑯を＋1
第二の操作【表のダイアモンド⑨⑩⑪⑫の時間を揃える】					
○○●●③④	p_3	0 0 1 1	0 1 1 1 1	0 0 0 0 0	⑨を－1
●○○●④①	p_{17}	1 0 0 1	1 0 1 1 1	0 0 0 0 0	⑩を－1
●●○○①②	p_{24}	1 1 0 0	1 1 0 1 1	0 0 0 0 0	⑪を－1
○●●○②③	p_{19}	0 1 1 0	1 1 1 0 1	0 0 0 0 0	⑫を－1
第三の操作【表のスクエア①②③④の時間を揃える】					
○●●●②③④	p_7	0 1 1 1	1 1 1 1	0 0 0 0 0	①を－1
●○●●③④①	p_4	1 0 1 1	1 1 1 1	0 0 0 0 0	②を－1
●●○●④①②	p_8	1 1 0 1	1 1 1 1	0 0 0 0 0	③を－1
●●●○①②③	p_{16}	1 1 1 0	1 1 1 1	0 0 0 0 0	④を－1
第四の操作【表のスクエア、表のダイアモンド、裏のダイアモンドの時間を12時に合わせる】					
【まず裏のダイアモンドを12時に合わせる】					
○○○○①②③④	p_{30}	1 1 1 1	0 0 0 0 0	m m m m	⑬⑭⑮⑯を－1
【表のスクエアと表のダイアモンド同じ時間に合わせる】					
合成操作	$p_{29}+p_{11}$	1 1 1 1	2 2 2 2 2	0 0 0 0 0	①②③④を－1
●○●○①③	p_{29}	1 0 1 0	1 1 1 1 1	0 0 0 0 0	②④を－1
○●○●②④	p_{11}	0 1 0 1	1 1 1 1 1	0 0 0 0 0	①③を－1
【さらに裏のダイアモンドと合わせる】					
●●●●①②③④	$p1$	1 1 1 1	1 1 1 1 1	0 0 0 0 0	⑬⑭⑮⑯を－1

・表のスクエアの時間が12時のとき必ず裏のスクエアも12時を指しているので、全ての時計が12時を指すこととなる。
・●●○○③④は、スイッチ①②をオフ、スイッチ③④をオンとして時計③または④を1時間進めることを意味する。

図10. RCIを解く手順

LOとRCIの抽象モデルを見てみると、影響行列をひとつの行列であらわしてしまうと、形式的には同じ格好をしている。しかし、RCIでは、各操作をスイッチの選択と時計の針を回すことの組としてデザインされており、パズルを見かけ上複雑にしている、影響行列の要素の値として3値を識別していること、演算定義も3値に対応していることから、後者が前者を含んでより一般的であることが分かる。従って、巡回パズルの抽象モデルは次のようになる。

巡回パズルの抽象モデル：(y, q, A, R, s, t)

(1) 要素の状態ベクトル：$y=[y_1, y_2, \ldots, y_m]$、$y_i \in \{1, 2, \ldots, q\}$
ここで m は要素数、q は状態数

(2) 影響行列の集合：$A=\{A_1, A_2, \ldots, A_p\}$,
ここで p は影響行列の数、各行列は n 行 m 列

(3) 操作集合：$R=\{r_1, r_2, \ldots, r_n\}$、ここで n は操作の数。

$$y \leftarrow r_i(y) = y + \omega_i^k \pmod q, \quad \text{ここで } k \in \{0, \ldots, p\}$$

は影響行列の切り替えを複数の操作により指定する。

(4) 操作の冗長度：d

(5) 初期状態：s: 最終状態から任意に操作を何回か繰り返した状態

(6) 最終状態：t: パズルのゴールとしてふさわしいある状態

3. 巡回パズルの柔軟な表現メディアへの変換とパズル・ジェネレータ

今まで見てきたLOや特にRCIは、かなり複雑なパズルであり、解くのにかなりな時間がかかる。あるいは、大体の人は解けないであきらめることも多いと思われる。巡回パズルの本質を有しながら、もう少し柔軟なメディアに変換することにより、ユーザの時間と能力に適合したパズルが簡単に生成することは意味のあることと考えられる。ここでは巡回パズルのメディア変換の第一段階として、いくつかの異なる単純なメディアで実現することを考える。すなわち、一次元グラフ表現、2次元グラフ表現、サウンド表現である。前者二者はパラメトリックな具現化に近く、後者は創造的な具現化を目指したものである。

コンピュータ・グラフィックスにより、影響関係をグラフとして表現する。グラフの頂点は、複数の状態を持ち、この状態を箱の面や色や時計の時間などにより表す。いわゆる**多色ライツアウト問題**と呼ばれるパズルである。辺は有向辺が基本であるが、LOのように全て双方向ならば無向辺で表しても良い。また、RCIのように複数種類の辺を用意する必要がある場合もある。影響関係は、LO、RCIともに規則性があったが、より一般的には任意のグラフとして定義できるのが望ましい。ただし、この場合には与えられたグラフに対して、全ての初期状態に対して解を持つかどうかが問題になる。これに関し、

参考文献[2]には、任意のグラフの入力パターンに対し、任意の解を1つ見つける問題を解く多項式オーダーのアルゴリズムが与えられている。さらに問題になるのは、グラフをどのように配置するかである。一般グラフの場合には、LO の場合のような整然とした配置を手作業で求めることは困難である。従って、この場合にもグラフ描画アルゴリズムを用いることが有効となる。

一次元グラフ表現

要素を一次元に配置する場合には、平面グラフの L-mapping [5]を用いることが有効である。これによれば、ある直線上に平面グラフの頂点を任意の順序に並べても、直線上の線分と半円により辺の交差なく描けることが保証されている。図 11 に 2×2、…、5×5 の LO の一元配置を示す。美しい対称性が表れている。

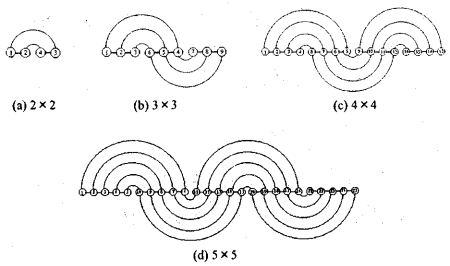

図 11. ライツアウトの一次元配置

二次元グラフ表現

要素の配置を二次元に拡げた表現である。これに用いることができる様々なグラフ描画アルゴリズム[7]が開発され、ソフトウェアが公開[8,9]されているのでそれらを利用することができる（前稿[1]の 5 節参照）。

グラフ表現に基づき、巡回パズルのジェネレータを開発した。単純なものから複雑なものまで、パラメータの値を変えると自由にパズルが構成できるソフトウェアである。その機能は次の通りである。

グラフ表現による巡回パズル・ジェネレータ

(1) パズルの定義機能、変更機能
ダイアログボックスにより、頂点数、巡回位数、影響行列を定義する機能を用意した。またインタラクティブにグラフを構築できる機能も用意した。さらにパズルの途中でも影響関係を修正したり、頂点や辺を追加・削除することができる。

(2) レイアウト機能、表示機能
自動描画法によりレイアウトを表示し、マニュアルでも修正できる。影響関係を明示する/しないを切り替えるスイッチを用意した。頂点の表示を箱、色、時計と切り変えることもできる。

(3) 初期状態設定機能
最終状態にランダムに操作を加えるシャッフル機能を実装し初期状態を与えることができる。

(4) 操作機能
各操作の実行は、マウスによるクリックやドラッグで行う。

(5) 保存機能、呼び出し機能
作成したパズルを保存し、呼び出す機能を実装した。保存に際しては、名前付け、自分のパズルとしての評価も記録できるようにした。

図 12 に一次元グラフ表現のジェネレータのユーザ・インタフェースを示す。左下のテキストによっても右下の図的なインタフェースによっても要素間の影響関係の定義をすることができ、パズルを生成することができる。

図 12 は箱が上下するものである。この他のバリエーションとして、多色ライツ・アウトに対応して箱が回転して数字や色が変わるもの、各要素が時計で表されるものを生成することもできる。これらを図 13 に示す。さらに、直交描画法を用いることによりより一般的な巡回パズルを生成することもできる（図 14 参照）。図 15 に一次元パズルの動作例を示す。

LO や RCI とグラフ表現との比較を次にまとめる。

(1) LO では配置の規則性により影響関係は結線を示さなくともユーザに明示的である。また RCI では、表と裏がある、スイッチの切り替えがあるなどにより、結線を描くのが困難なため影響関係は明示的ではない。これを一般グラフにすると影響関係を結線で明示することが必要な場合が出てくる。

(2) グラフ表現に適する影響関係はやはり平面グラフが望ましいように思われる。しかし少数の辺の交差によりパズルを複雑にすることも面白いかもしれない。

(3) レイアウトのアルゴリズムとして何を用いるかもバラエティーをあたえる要素としてとらえることができる。

(4) グラフ表現において、複数の影響行列を用いる場合にどのようなレイアウトを行っておけばよいかは興味深い問題である。頂点の位置を動かさないで結線だけを変えた場合にどの影響行列の表現でも出来るだけ良いレイアウトを得る問題である。

(5) RCI の場合は、裏と表があり一覧性がなかったが、グラフ表現では一覧性がある。

(6) 簡単なものでも解くのはかなり困難である。図 15 に示すシンプルなものでも、数人の人に挑戦してもらったところ、なかには解けずに諦めてしまう人もいた位である。

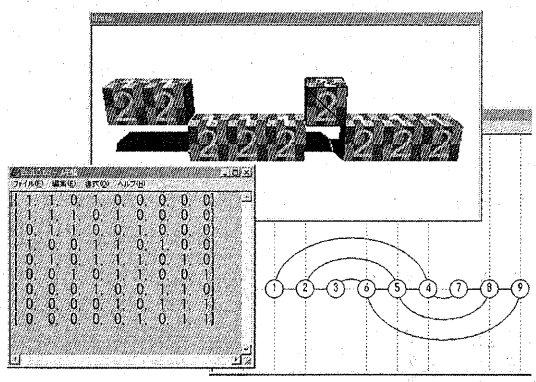

図 12. 一次元ジェネレータのインタフェース

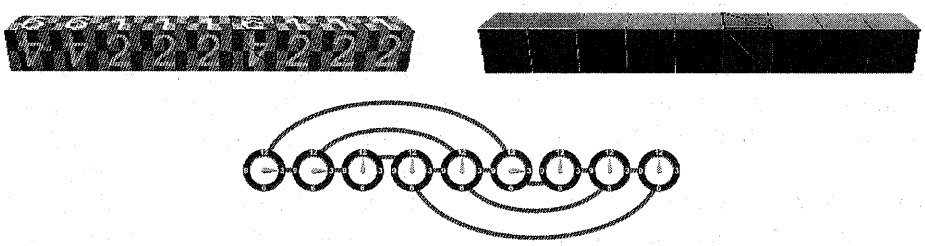

図 13. 一次元グラフ表現のバリエーション．上は箱が回転し数字が変わるもの、色が変わるもの．下は時計の時刻が変わるものである．

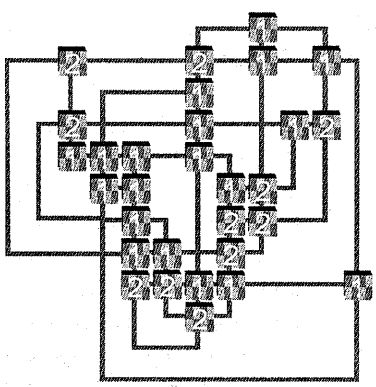

図 14. 二次元グラフ表現の例

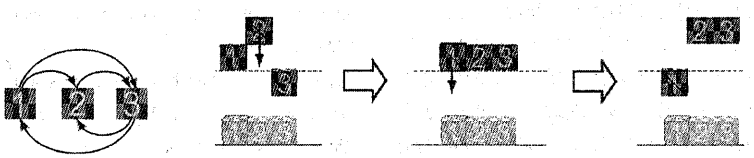

図 15. 一次元巡回パズルの動作例．上は影響関係の定義を示す。下は左より、要素 2 を下へ、要素 1 を下へ動かした場合の全体の動きを示す。半透明の３つの箱の位置は、ゴール（パズルを解いた状態）を示している．

サウンド表現

創造的な具現化として次に示す音による表現メディアへの変換を考えた。巡回パズルは、パズルの状態を表わす方法として、必ずしも視覚的な方法にたよる必要はなく、なんらかの方法で巡回する複数の要素で状態を表すことができさえすればよい。そこで音メディアへの表現として、ごく短いメロディを構成する音符のセットを要素とし、それらの音程の変化が巡回する表現を具現化するPCとMIDI音源を用いた方法を開発した。

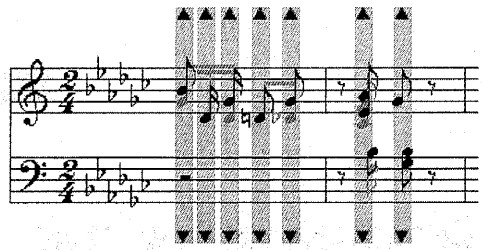

図16. サウンド・パズルのユーザ・インタフェース
(黒い音符は現状を、半透明の音符はゴールを示す。実際のインタフェースはもっと簡略化した表現をとっている。)

図16に試作した音による操作パズルのユーザ・インタフェースを示す。このパズルでは、はじめに図16のように、PC画面上に、あるメロディを表現した楽譜が表示されており、そこに表示されている各音符の背後にあるコントローラーで、それぞれの音符の音程の表示を上下にコントロールすることができるようになっている。音符の音程を変化させると、そのたびに、楽譜に記載されたとおりのメロディが流れる。このパズルでは、この音符表示をオンあるいはオフにし、おかしなメロディがなる状態から、もとの“音楽的な”メロディが鳴る状態へ戻すことが課せられる。しかし、このときやはり各音符の音程を操作すると、他の音符の音程も付随して動くというかたちで、巡回パズルが成立する構造を備えている。

このパズルに著者の一人が挑戦して気づいたことは、同じパラメータを備えた視覚的な操作パズルより、ずっと難しくなるということである。その理由は、音感が必要になることと、状態を表現している複数の音が、順にしか鳴らないことによる。また、音によるパズルは、ほかにも、巡回パズルのモデルをマッピングする様々な方法が考えられ、それ自体一つの研究課題といえる。例えば、著者らが試作した方法のほかに、複数の楽器の合奏によって作り出される楽曲の各パート、あるいは、ポリリズムを形成する複数の打楽器の各パートの鳴る鳴らないなどによって巡回操作を形成し、それぞれのパートを操作したときの従属関係を設定するといった方法などが考えられる。

評価のための予備実験として、7つの音からなる良く知られたメロディーの楽譜を用意し、DJ経験者、音大ピアノ科出身者などに楽譜を用いた視覚的パズルと音だけのパズルの両方を解く問題に挑戦してもらい感想を聞いた。それらをまとめると次のようであった。

(1) 大変新鮮で興味深いパズルであると感じた。影響関係がないか極く少ない場合でもこのことはいえる。
(2) 音だけのパズルは7音でも難しい。ルールを探索するというよりは、試行錯誤的に解いていく感じであった。4音位でもいいのではないか。
(3) 音に対する感受性を高める意味で、幼児から大人までの音感教育に役立つと考えられる。大いに将来性があると感じた。

4. まとめ

本論文では、いくつかの既存の巡回パズルを、その抽象的表現を介して、他の表現メディアに変換し、操作パズルの特色に加えて、新たな特色を付加する試みを行った。今後の研究課題として、音によるパズルのバリエーションを試作することや、それらのエデュテイメントとしての評価が残されている。また、パズル以外の玩具が持っている複雑な挙動に対しても、数理モデルによる分析ができないか、目を向ける必要がある。

参考文献

[1] 前田篤彦・杉山公造・間瀬健二：置換パズルのメディア変換とパズル・ジェネレータの試作、情報処理学会第101回ヒューマンインタフェース研究発表会、(2002).
[2] 青木史郎：ブール体における配置問題、豊橋工科大学平成9年度特別実験報告書、(1997).
[3]http://www.ic-net.or.jp/home/takaken/nt/light/index.html
[4]http://www.ic-net.or.jp/home/takaken/nt/light/light2.html
[5] http://www.dotsphinx.com/games/rubik/clock/
[6] Ozawa, T.: Planarity testing for IC layout with constraints for pin order and congestion between pins, IEEE Conf. Record of the 14th Asilomar Conf. on Circuits, Systems and Computers, 188-192, (1980).
[7] Sugiyama, K.: Graph drawing and applications for software and knowledge engineers, World Scientific, (2002).
[8] http://www.mpi-sb.mpg.de/AGD/
[9] http://www-pr.informatik.uni-tuebingen.de/yfiles/