

難しい迷路の条件に関する研究

吉田喜峰廣^{†1} 小泉康一^{†1} 大槻正伸^{†1}

概要: 迷路とは、ルールに従ってスタート地点から進んで行き、複雑に入り組んだ道を抜けてゴールまでたどり着くことを目指すパズルゲームである。昨今様々な迷路ゲームが作られ、人々の間で楽しまれている。本研究では難しい迷路の条件について、何が人間にとって難しいといえるのかプログラムを用いて調査し発見することを目的とする。迷路を実際に複数の被験者に解いてもらう実験を経て、進む方向の決定に迷路の構造が影響し、それによって特定の進みやすい通路に移動することがあることが分かった。

キーワード: 迷路, 迷路内探索, 迷路内探索における人間の行動心理

Study on the Condition of the Difficult Maze

YOSHIDA KIMIHIRO^{†1} KOICHI KOIZUMI^{†1} MASANOBU OHTSUKI^{†1}

Abstract: A maze is the puzzle game to arrive to the goal spot through a complicated aisle. Many kind of maze are enjoyed as a game by people. In this study, we are researching and discovering the condition of the difficult maze what is difficult for the human. In experiments of solving a maze by some test, we find that the human decide an aisle by structure of a maze and move to an aisle easy to go.

Keywords: Maze, Maze search, Ground rule in maze searching for the human

1. はじめに

迷路とは、ルールに従ってスタート地点から進んで行き、複雑に入り組んだ道を抜けてゴールまでたどり着くことを目指すパズルゲームである。迷路探索には確率的、心理的要素や運など様々な要素が関係してくるため、同じ迷路でも解く人や解き方によって難しく感じるか簡単に感じるかは全く異なる。迷路においてスタート地点からゴール地点までたどり着くのに移動した距離を移動距離とする。本研究では人間が実際に迷路を解くケースに焦点を当て、難しい迷路とはスタートからゴールまでの移動距離が最短のものに比べて大幅に超えてしまうことの多い迷路であるとする。難しい迷路の条件は、難しいことと深い関係を持つ迷路内に内在する項目（パラメータ）、つまり何が迷路を難しくしているのかであり、本研究における難しい迷路の定義から、難しい迷路の条件は移動距離の増加に関係する項目であると考えられる。このことから、迷路を実際に解き、移動距離と密接な関係を持つ迷路内に内在する事項を発見することで、難しい迷路の条件を発見することができると考える。

2. プログラムを用いた迷路の表現

本研究では複数の被験者に迷路を解いてもらう実験を行い、難しい迷路の条件を直感的に見つけていく。専用に

作成したプログラムを用いて迷路を表現、描画し、その中で迷路上を被験者に動いてもらい解いてもらうことにした。迷路の表現には二次元配列を使用しており、配列の変数 1 つ 1 つを 1 マスとし、壁を 1、通路を 0 などのデータ構造を用いて、それを図として描画する。スタート地点、ゴール地点の情報は x 軸と y 軸の座標で与える。スタート地点からゴール地点までの実際の移動距離は、プログラムによる 1 マス単位の迷路の表現に伴い、スタート地点からゴール地点までに実際に移動した 1 マス 1 マスの移動数で数える。

また、迷路のランダムな生成には、迷路自動生成アルゴリズムを用いる。図 1 に示すように大きく 3 種類あり、どれを用いるかによって迷路の構造に差が表れる[1][2]。実験ではこれらの迷路生成アルゴリズムを作成し、必要に応じてそれらを選択して運用する。

(a) 棒倒し法

迷路内に一定間隔に壁を配置し、一定の規則に従いながら配置した壁の四方 1 マスからランダムに一か所を壁にしていくことで迷路を作成する方法。

(b) 穴掘り法

迷路内全域を壁とし、その壁を掘り進めるように通路を延ばすことで迷路を作成する方法。

(c) 壁のぼし法

四方の端を除く迷路内全域を通路とし、壁を延ばして迷

^{†1} 福島工業高等専門学校 電気工学科
Electrical Engineering, National Institute of Technology, Fukushima College.

路を作る方法.



(a)棒倒し法 (b)穴掘り法 (c)壁のぼし法

図1 迷路自動生成法

Figure 1 Automatic creating method of the maze.

3. 実験

3.1 第一実験

第一実験として, 31×31 マスの迷路面積で迷路をランダムに生成し, それを①迷路の全体が見える状態と, ②視界の広さを現在位置から四方 ± 2 マスに限定する2つの状態で被験者に実際に解いてもらい迷路内での移動軌跡と移動距離を記録した. その後, 移動距離と迷路の構造に基づいたパラメータとで相関をとり, 移動距離と関係の深いパラメータを調べる.

3.2 迷路のパラメータ

迷路にはスタート地点からゴール地点に至るまでの長さや分岐点の数など, 構造に違いを与える要素が存在する. 第一実験ではこれらの要素について迷路内での移動距離に顕著な影響を与えると思われるものを迷路に内在するパラメータとして数値化し, 実験の結果と比較する. 第一実験においては, 以下の要素をパラメータとした.

(1) 正解ルート深さ

スタート地点からゴール地点まで最短距離でたどり着いた際の移動距離.

(2) 最大深さ

スタート地点から最も深い地点まで最短距離でたどり着いた際の移動距離.

(3) 分岐点の数

迷路内の分岐点の数. 迷路全域における分岐点の数と正解ルート内の分岐点の数の両方を含める.

(4) 外れ深さ

正解ルートから外れた場合の分岐マスから行き止まりマスまでの数のうち最大のもの.

3.3 実験結果

表1に第一実験の結果を示す.

表1 各パラメータにおける移動数との相関係数

Table 1 The coefficient of correlation with each parameter and the number of movement.

全域	計測回数	移動数との相関				
		正解ルート深さ	最大深さ	最大外れ深さ	正解ルート内分岐点	迷路全域内分岐点
棒倒し	7	0.821	0.536	-0.027	0.512	0.683
穴掘り	7	0.992	0.337	-0.422	0.302	0.159
壁のぼし	7	0.992	0.635	-0.794	0.716	0.207
視界制限	計測回数	移動数との相関				
		正解ルート深さ	最大深さ	最大外れ深さ	正解ルート内分岐点	迷路全域内分岐点
棒倒し	16	-0.221	-0.211	0.084	-0.304	0.162
穴掘り	16	0.352	0.252	-0.229	0.024	-0.526
壁のぼし	17	0.467	0.329	-0.081	0.070	-0.373

①の全域が見える状態では正解ルート深さととの相関が最も大きかった. これは, 先に正解ルートを目視で割り出してから迷路内の移動を開始することができるためであると考えられる. また, ②の視界を制限した場合には, 移動数と各パラメータとの間に大きな相関は得られなかった. そのため, ゴールが分からない状態では本実験で定めたパラメータのほかに, 迷路内の移動数に影響する要素があることが示唆される.

4. 第二実験

4.1 第二実験の概要

第二実験では, 迷路の大きさとスタート位置, ゴール位置をランダムとして迷路を作成し, それを視界の広さを現在位置から四方 ± 2 マスに制限した状態で複数の被験者に解いてもらって移動距離と移動軌跡を取得した. 迷路は前述の迷路自動生成アルゴリズムを用いたものを解いてもらい, その結果を踏まえて次に手で生成した迷路を解いてもらった. その結果から, 平均して移動数の多かった迷路の共通点を発見し, 移動数が多くなるとされる条件を考察することを本実験の目的とする.

4.2 実験結果

図2にある被験者の移動軌跡の画像の一例を示す. 黒が壁, 白が一度も通ったことのない通路, 灰色が一度でも通行した通路を示す. この迷路は手で生成した.

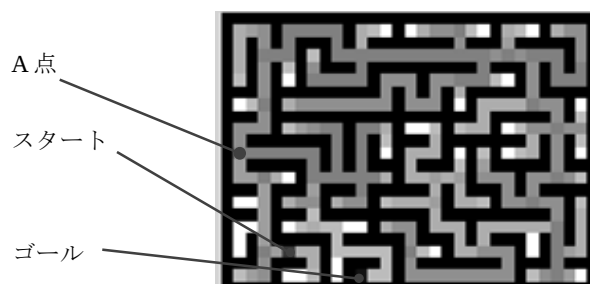


図2 第二実験で得られた移動軌跡の一例

Figure 2 An example of the track for second experiment.

A点はゴールに至るための重要なポイントで, ここから右方向に進むとゴールに近づくが, 上方向に進むと決してゴールにたどり着けない. 14名の被験者のうち, A点を通じた際に上に進んだ人は10名で, 右に進んだ人は4名となった. 上に進んだ場合は最終的にA点に戻ってこなければならないため, 移動数が大きく増加した人が多かった. この現象がみられた通路の特徴的な形状についてこの迷路のスタート地点近くのところを取り出して別に表現したものを図3に示す.

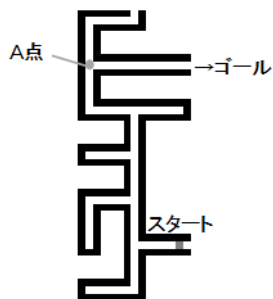


図3 図2の迷路の注目すべき構造

Figure 3 The notable part of the structure for the maze of figure 2.

図2の迷路を見ると、図3のようにスタート地点から上方向に進むような構造になっており、A点までの部分では仮に間違いルートを進んだとしても袋小路が小さいため、すぐに元の場所に戻てくる事ができる。そのため、迷路を進むうちに上以外に進む意識が薄れ、上に進もうとする意識が働くことでA点においても上に進んでしまったとみられる。このような進みやすい分岐点の存在により上方向に進んでしまうことが多かったことから、迷路の難しさに関係する要素は、迷路の構造による進みやすい分岐点の存在であると考えられる。

4.3 実験結果のシミュレーション

難しい迷路の条件については前述したが、進みやすい分岐点が発生した原因と考えられる迷路探索における人間の行動と同様な考えをプログラムによるシミュレーションを行うことで再現できないかと考えた。そこで迷路の自動解析プログラム上で動く迷路探索のシミュレーションプログラムを作成した。迷路の自動解析プログラムのおおまかなフローチャートを以下に示す。

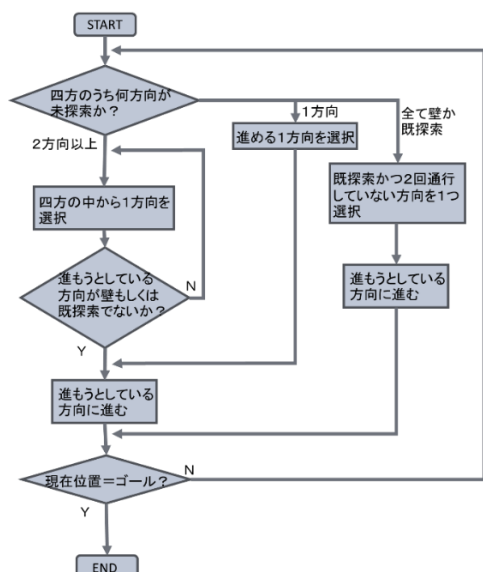


図4 迷路の自動解析プログラムのおおまかなフローチャート

Figure 4 Flow chart of automatic analysis for a maze.

上記のアルゴリズムの重要な部分は、分岐点にたどり着いたとき、どの方向に進むか選択するルールである。まず、方向に重みづけをする。具体的には図5のようにスタート地点から現在位置に至るまでの移動数をどの方向に進んだかそれぞれ記憶し、四方の移動数を選ばれる重みづけに対応させる。そして重みづけした上でランダムに移動方向を決定するようにした。こうすることにより迷路内で同じ方向に進めば進むほどその方向に進むような探索を行うことになる。これを利用して第二実験のときに見られた直進的な迷路探索を再現する。

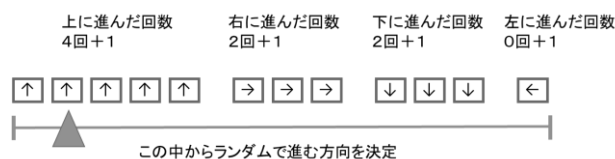


図5 シミュレーションプログラムにおける迷路内の分岐点での移動方向の選出例

Figure 5 An example to select the direction in the maze.

シミュレーションプログラムを用いて第二実験で使ったいくつかの迷路の探索を行った。以下に、図2の迷路におけるシミュレーションプログラムを用いた迷路探索の軌跡画像と運用結果を示す。図2と同様、黒が壁、白が一度も通ったことのない通路、灰色が一度でも通行した通路を示す。

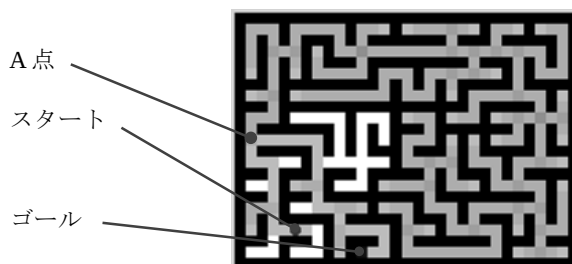


図6 シミュレーションプログラムを用いた移動軌跡

Figure 6 The truck using simulation program.

表 2 シミュレーションプログラムの運用結果
(プログラムの試行数 100 回)

Table 2 A result of simulation program.
(Number of run: 100)

	上	右
シミュレーション プログラムを用いた場合の A 点における方向別 移動回数	66	34
A 点における 移動確率[%]	66.0	34.0
移動確率から算出される 図 2 の迷路の A 点に おける移動人数[人]	9.2	4.8
第二実験における A 点での移動人数[人]	10	4

シミュレーションプログラムを用いた場合においても、A 点で上に進む場合が多かった。また、第二実験で得られた結果はシミュレーション結果に基づく移動確率にきわめて近いことが分かる。このことから、視界が見えない状態では人は直線的に迷路の奥へと進もうとする探索を行い、それによって進みやすい通路が発生していると思われる。

4.4 実験結果をうけて

A 点での分岐において、正解である右方向に進んだある一人の被験者には、スタート地点周辺の通路をくまなく探索しようとする傾向がみられた。これはグラフにおける幅優先探索によく似ており、A 点を上に進んだ被験者の傾向は深さ優先探索に類似していることが分かる。このことから、人間は迷路探索を行う際にある一定のアルゴリズムを思考しながら探索を行う傾向があり、人間の迷路内探索におけるアルゴリズムとしては、直感的に迷路を奥まで進んでいこうとする深さ優先型の探索とスタート地点周辺の通路から探索を行う幅優先型の探索があると考えられる。また、それらの 2 つの探索以外にも、右手法を利用して迷路探索を行ったと考えられる人などが見られたが、いずれの場合も、何かしらのアルゴリズムに則って探索を行い、完全にランダムな探索は避ける傾向があることが考えられる。

5. まとめ

本研究では、迷路の自動生成アルゴリズムを用いて、あるいは手動で迷路を作成し、それを複数の被験者に解いてもらう実験を行ってきた。その結果、難しい迷路の条件の一つが迷路の構造による進みやすい通路の存在であると考えられることが分かった。また、その通路の発生条件は同方向への繰り返しの移動や、直線的な通路の構造などであると思われることも分かった。

今後の方針としては、同じ地点にたどり着くことのできる複数の通路が含まれる迷路やその他特定条件下など、複雑な条件下における難しい迷路の条件の追究や、今回発見した難しい迷路の条件の具体的な数値化などがあげられる。

謝辞

実験にご協力いただきました福島工業高等専門学校電気工学科 5 年の皆さまに感謝の意を表させていただきます。

参考文献

- [1] 迷路自動生成アルゴリズム: <https://www5d.biglobe.ne.jp/stssk/maze/make.html> (参照 2017-02-3).
- [2] ピーター・グロゴノ, シャロン・H・ネルソン, “問題解決とプログラミング,”近代科学社, 1985.

付録

付録 A 迷路探索における人間の行動のシミュレーションプログラムのアルゴリズム

本項では 4.3 で示した迷路探索における人間の行動のシミュレーションプログラムを紹介する。

迷路全体を $a \times b$ マスとしたときの迷路の構造を $\text{maze}(a, b)$ とし変数によって 0: 未探索の通路, 1: 壁, 2: 一度だけ通行した通路, 3: 2 度以上通行した通路として示す。スタート地点を x_s, y_s , ゴール地点を x_g, y_g , 現在位置を p_x, p_y として座標データで示す。

また、移動軌跡の記憶には、 $\text{maze}(a, b)$ と同じ大きさを持つ $\text{save}(a, b)$ を用いる。迷路内の四方への移動数の記憶に $\text{directions}(k)$ を用い、 k によって 0: 総合数, 1: 上, 2: 右, 3: 下, 4: 左の移動数を記憶し、初期値として、 $\text{directions}(0)=4$, $\text{directions}(k)=1(k=1, 2, 3, 4)$ が記憶されている。各方向に進んだ場合、進んだ方向に対応する directions と $\text{directions}(0)$ に +1 される。

付録 A.1 アルゴリズムの流れ

入力 (迷路データ $\text{maze}(a, b)$, スタート位置 x_s, y_s , ゴール位置 x_g, y_g , 現在位置 p_x, p_y)

出力 (移動軌跡 $\text{save}(a, b)$)

$p_x=x_s$ and $p_y=y_s$ (現在位置をスタート位置に設定)

四方を確認

$w =$ (現在位置の四方 1 マスのうち、未探索のマスの数)

Select Case w

Case $w=0$

来た道に戻る処理: 四方 1 マスのうち一度通行したマスに戻る

Case $w=1$

一本道の処理: 未探索のマスに進む

Case $w>1$

分岐点の処理: シミュレーションプログラムに

おける分岐点の選択アルゴリズムの呼び出し

End If

If $p_x = x_g$ and $p_y = y_g$ then プログラムを終了する

Else 四方の確認に戻る

End Sub

付録 A.2 迷路内での移動アルゴリズム

入力 (迷路データ $\text{maze}(a, b)$, 現在位置 p_x, p_y , 移動する方向 r)

出力 (迷路内の 1 マスの移動)

Select Case r (r によって進む方向を決定)

Case 1 (上に進む場合)

If Not $\text{maze}(p_x, p_y - 1) = 0$ Then (上が通路でない場合)

方向の選択をやり直す

Else

$\text{maze}(p_x, p_y - 1) = 2$ (現在位置の 1 マス上を一度進んだ通路とする)

$p_y = p_y - 1$: $\text{save}(p_x, p_y) = \text{save}(p_x, p_y) + 1$

(現在位置を 1 マス上に移し, 移動軌跡を記録)

$\text{directions}(1) = \text{directions}(1) + 1$

$\text{directions}(0) = \text{directions}(0) + 1$ (上に進んだ回数と移動した総合数に+1)

End If

Case 2 (右に進む場合)

If Not $\text{maze}(p_x + 1, p_y) = 0$ Then (右が通路でない場合)

方向の選択をやり直す

Else

$\text{maze}(p_x + 1, p_y) = 2$ (現在位置の 1 マス右を一度進んだ通路とする)

$p_x = p_x + 1$: $\text{save}(p_x, p_y) = \text{save}(p_x, p_y) + 1$

(現在位置を 1 マス右に移し, 移動軌跡を記録)

$\text{directions}(2) = \text{directions}(2) + 1$

$\text{directions}(0) = \text{directions}(0) + 1$ (右に進んだ回数と移動した総合数に+1)

End If

Case 3 (下に進む場合)

If Not $\text{maze}(a, b + 1) = 0$ Then (下が通路でない場合)

方向の選択をやりなおす

Else

$\text{maze}(a, b + 1) = 2$ (現在位置の 1 マス下のマスを一度進んだ通路とする)

$p_y = p_y + 1$: $\text{save}(p_x, p_y) = \text{save}(p_x, p_y) + 1$

(現在位置を 1 マス下に移し, 移動軌跡

を記録)

$\text{directions}(3) = \text{directions}(3) + 1$

$\text{directions}(0) = \text{directions}(0) + 1$ (下にすすんだ回数と移動した総合数に+1)

End If

Case 4 (左に進む場合)

If Not $\text{maze}(p_x - 1, p_y) = 0$ Then (左が通路でない場合)

方向の選択をやり直す

Else

$\text{maze}(p_x - 1, p_y) = 2$ (現在位置の 1 マス左を一度進んだ通路とする)

$p_x = p_x - 1$: $\text{save}(p_x, p_y) = \text{save}(p_x, p_y) + 1$

(現在位置を 1 マス左に移し, 移動軌跡を記録)

$\text{directions}(4) = \text{directions}(4) + 1$

$\text{directions}(0) = \text{directions}(0) + 1$ (左に進んだ回数と移動した総合数に+1)

End If

End Select

付録 A.3 シミュレーションプログラムにおける分岐点の選択アルゴリズム

入力 (迷路データ $\text{maze}(a, b)$, 現在位置 p_x, p_y , 四方への移動数 $\text{directions}(0 \sim 4)$)

出力 (迷路内での 1 マスの通行)

$r = \text{Rnd}(1) * \text{directions}(0) + 0.5$ (四方の移動数を合計してその中から 1 つ選出する. 詳細は図 5 を参照)

If $r \leq \text{directions}(1)$ Then (上の移動数に当てはまる場合)

進行方向を上決定

ElseIf $\text{directions}(1) < r$ And $r \leq \text{directions}(1) + \text{directions}(2)$ Then (右の移動数に当てはまる場合)

進行方向を右に決定

ElseIf $\text{directions}(1) + \text{directions}(2) < r$ And $r \leq \text{directions}(1) + \text{directions}(2) + \text{directions}(3)$ Then (下の移動数に当てはまる場合)

進行方向を下に決定

ElseIf $\text{directions}(1) + \text{directions}(2) + \text{directions}(3) < r$ And $r \leq \text{directions}(1) + \text{directions}(2) + \text{directions}(3) + \text{directions}(4)$ Then (左の移動数に当てはまる場合)

進行方向を左に決定

End If

選択した方向に移動

$\text{cogr} = \text{cogr} + 1$ (移動数に+1)

End Sub