

大規模迷路探索シミュレーション - その1 平面の場合 -

関東職業能力開発大学校 加部 通明

The search simulation of a large scale Maze - No.1 In the case of a plane -

Michiaki KABE

要約 迷路探索とは、その探索過程で区画情報を得ながら出発地から目的地まで辿り着く路を発見する作業であり、特に、入口から出口までの最短経路を発見する問題が重要である。それは災害救助に応用できるからである。地震災害等により建物が倒壊し、入り組んだ路を迷路に例え、外から見えずに隠れている人達を探索することが、迷路探索そのものに相当するからである。

迷路の最短経路探索問題において、1回の試行で最短経路を求める手法は未だ存在しない。何故ならば、最初の試行では、迷路を構成する区画情報が何も得られていないからである。しかし、確実に目的地に辿り着く手法であれば、冗長ではあるが常に左手方向の路を選択して進む左手法を若干修正した手法等が存在する。

今回、様々な形状の大災害にでも対応できるように、大規模で複雑な平面迷路を作成し、迷路全体の情報を効率よく収集する為の探索方法を考察した。具体的には、トレモ一法を基に人間の知能に相当するパラメータを考案し、そのパラメータを変化させて迷路探索シミュレーションを行った。本稿では、そのシミュレーションにより初回の探索距離と準最短経路を発見するまでに要した総探索距離を調査した結果について報告する。

I はじめに

迷路に関する題材には、大きく分けて4つある。

- ①自律型ロボットが迷路を走行し、その最短走行時間を競うマイクロマウスと呼ばれる競技。
- ②迷路歩行時に迷路を選択する歩行行動を探る研究。
- ③迷路作成や迷路探索を行うアルゴリズムの研究。
- ④迷路の形や路の図柄を制作するデザイン分野。

②については、特に建築分野で空間と構造という環境の視点から人間行動に及ぼす影響を探るところから来ている⁽¹⁾。①と③については、左手法や求心法等決まり切った融通が利かない方法があるが⁽²⁾、余り知的ではない。更に、大規模な又は3次元の迷路探索では迷路の全区画情報を得にくい点がある。どんな迷路にも対応できる探索法には人間の行動様式を真似たトレモ一法がある⁽²⁾。本稿では、特に大規模な迷路を取り上げて、3次元への拡張をも視野に入れた立場からトレモ一法を若干修正した方法により迷路探索を考察した。

また、①～③に関連するものとして迷路探索シミュ

レーションがあり、その応用としてロボットによる災害救助活動が考えられる。迷路探索を実際に応用する為には、路に関する情報がなければならぬが、それにはいろいろな路を通らないとそれらの情報が得られない。1回の通り抜けて全体の路を把握できるためには、位置座標と進行方向との情報を持っていないならぬ。ロボットが情報を収集し、それをパソコンで情報処理をして目的地に安全に到着する為には、パソコン上で一度迷路探索シミュレーションを行ない、その結果が良ければ実際にロボットにその内容通りの行動を取らせるように指示すればよいと考えられる。

迷路探索問題は人工知能における探索問題の基本であり、本稿での路の選択操作では、プロダクションルールが間接的に応用されている。

今回作成したプログラムでは、迷路規模(縦×横)を640×1,024まで拡大して、大規模な迷路を通るロボットを別の物体に変えてその物体が通り抜ける迷路探索シミュレーションになっている。

探索に関しては、物体に実際の人間や動物がとろう

であろう行動に即した3通りの知能レベルを持たせ、1回の試行において、各知能レベルによってどのような迷路探索を行うかを調べた。更に、複数回の試行により入口と各区間の距離を表示した距離地図を作成して、迷路探索における最適化問題の準最適解を得た。

また、人が物体に変わったつもりで、迷路を通り抜けるゲームも制作した。迷路シミュレーションは、迷路に落とし穴や物体を捕獲する仕掛などを作ることでもでき、ゲームソフトの基本ともなっている。

II 人工知能からの考察

1 迷路探索問題の解釈⁽³⁾

具体的な迷路探索シミュレーションを考える前に、迷路探索問題を人工知能の視点から解釈しておく。

迷路探索問題は、初期状態から目標状態までの経路を発見する問題である。物体は初期状態から区画を一つずつ移り、その時々で物体の位置が連続的に変わる。迷路探索問題を解決するためには、物体が実際に移動しなければならない。この動きを問題解決のための作用素と呼ぶ。この作用素を適当に順次適用する事によって問題を解決する事が出来る。

迷路探索問題では、物体が迷路のどの位置にいるかによって、その問題の状態を表す事が出来る。従って、一つの作用素によって問題の対象の状態が変化する。この状態を表すのに状態空間なる概念を導入する。対象の状態を状態空間の1点で表せば、作用素による状態の遷移は状態空間における位置の移動とみなす事ができる。このようにして、迷路探索問題は状態空間における初期状態から目標状態への状態変遷移を求める探索問題と解釈する事ができる。

2 状態空間のグラフ化⁽⁴⁾

状態空間をグラフで表すことによって、グラフ理論での探索方法が応用できる。特に、最短経路を求める問題については有名なダイクストラの方法があり、規模が小さい状態空間ならば余り時間を掛けずに解ける^(註1)。しかし、今回取り上げる80×128以上の規模が大きい場合には、メモリ確保の問題でダイクストラの方法は実用的ではない^(註2)。そこで、発見的、経験的、確率的な探索方法の必要性が生まれる。

3 最適化へのアプローチ

大規模な状態空間の探索には、現在、大別すると次

の3通りの探索方法が上げられる。しかし、これらも非決定的な探索方法であり、重複や後戻りという試行錯誤を繰り返して解を求めることには変わりがない。

(1) ヒューリスティック探索⁽⁵⁾

ヒューリスティックな知識（経験や発見による役にとつ知識）を利用して解く手法である。迷路探索問題では、目標点の方向がわかっているならば、路の分岐点では目標に近いというヒューリスティックな知識を用いて探索する。

(2) 遺伝的アルゴリズム探索⁽⁶⁾

遺伝的アルゴリズムは生物の遺伝と淘汰の仕組みをモデル化して、複雑な状態空間の最適解を確率的に解く手法である。最近では、これを並列処理に適用することが期待されている。

(3) 分散探索⁽⁷⁾

分散人工知能の考え方の範疇に入るもので、状態空間をいくつかに分割して、その分割された状態空間での最適解を求め、大局的な最適解に結び付ける手法である。並列処理が高められるのがその特徴である。

本稿で取り上げる迷路探索手法は(1)のヒューリスティック探索に近い。

III 迷路作成について

1 迷路とは

迷路を作成するに当たって、先ず迷路の定義を考える必要がある。平面を格子状に区切った時の基本となる最小の図形を区画と呼ぶことにする。

[迷路の定義]

次の4条件を満たす区画の集合を迷路と呼ぶ。

- ① 入口（初期状態）と出口（目標状態）がそれぞれ1箇所ずつある。
- ② 1つの区画にきてはじめてその区画の情報（進路）がわかる。
- ③ 区画は外見上同一であり区別できない。
- ④ 入口から全ての区画に到達可能である。

2 具体的な迷路の作成方法

迷路作成の操作手段として、次の2方法を採用した。

(1) 手作業による

格子状に書かれた迷路で、マウスを用いて区画の線をクリックし消去したり追加したりする。手作業なのできちんと迷路の定義を満足しているかどうかを調べ

るのが煩わしいので、大規模には向かない。

(2)全自動による

迷路の定義に基づいて、制約が少なく自由度が大きい迷路を自動的に作成する。迷路作りにはいろいろあるが⁽⁸⁾、今回は次のような方法により作成する。

①出口に到るまでの道を1つ確保する。

規模が大きい場合には騙し道を1本確保する。

②各区画からこれら2本の道に通ずる道を作る。

③空白の区画が集まっているところを道を塞がないように壁を作る。

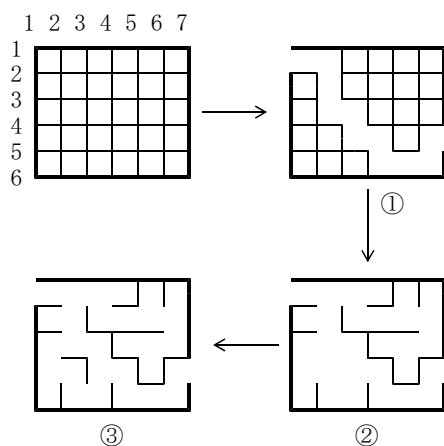


図1 迷路の作成手順

3 迷路の規模

迷路は区画が多いほど複雑になって面白い。パソコンで表示する場合、1区画は最低でも6ピクセルぐらい必要だから、パソコンの画面制限により最大80×128程度の迷路が作成できる。更に大きい規模の迷路については、複数画面を利用して迷路を表示する階層的方法を採用する。その場合、1画面では40×64規模の迷路を表示させ、その表示画面が全体のどの部分に相当するかを明示するために、迷路の左下角に全体を分割した格子状の正方形でその位置に赤色を付けて示している(図2)。この階層的な表示方法の採用によって、メモリの制約上次の8通りの迷路を作成した。

$$(5 \cdot 2^{n-1}) \times (8 \cdot 2^{n-1}), \quad n=1,2,3,4,5,6,7,8$$

IV 探索方法について

1 具体的な探索手法

1. 1 区画の選択 (状態空間について)

物体が迷路を通るには、区画1つ1つに情報を持たせその区画情報により物体が判断して迷路を選択する。

区画に持たせる情報には次のようなものがある。

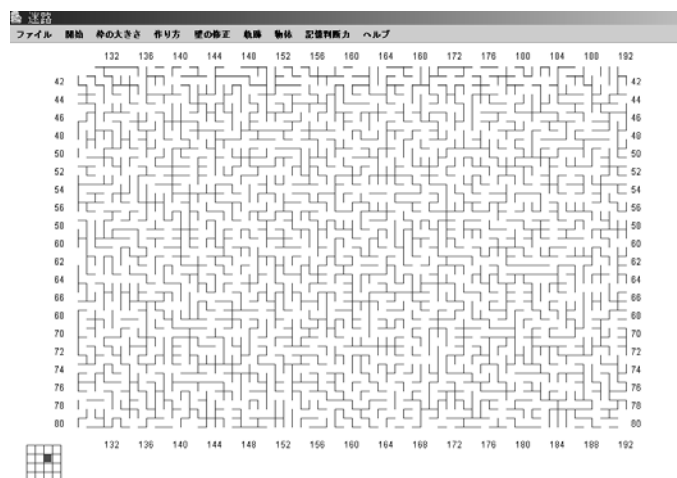


図2 160×256の大規模な迷路の複数画面による表示

[区画の基本情報]

壁の有無、区画通過回数、一方向からの進入の有無、入口からの距離(到達区画数)

プログラムでは、区画を図3のように配列を用いて表す。この時、隣合う区画(I,J)と区画(I,J+1)との境の壁の有無情報が一致しなければならない。また、区画(I,J)の集合が状態空間の具体的な表現になっている。

(I-1, J-1)	(I-1, J)	(I-1, J+1)
(I, J-1)	(I, J)	(I, J+1)
(I+1, J-1)	(I+1, J)	(I+1, J+1)

図3 区画の配列

1. 2 移動規則 (作用素について)

物体が移動する際、何の規則もないと、物体の運動がつまらないものになってしまうばかりか、出口に到達するまでかなりの時間を要する事にもなる。そこで、次のようなトレモー法に準拠した移動規則を設ける。

①来た道は先が行き止まりでない限り引き返さない。

②分岐点では、通過回数の少ない路を選ぶ。

同一の場合は乱数により選択する。

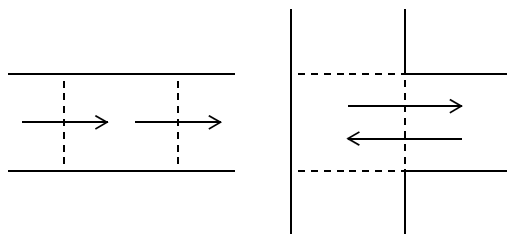


図4 移動規則

1. 3 プロダクションルールについて

プロダクションルールとは、ある条件を満たせばある結果を引き起こすという判断操作基準を与えるものである。

迷路探索では、いきなり壁のない方に進むということとはせずに、先ず着目区画の周辺区画の通過回数を調べる。それを調べる時に、上で説明した移動規則に基づいて、プロダクションルールを使用し、探索プログラムを作成する。

具体的には、次の手順で処理を行う。

- ①最初に周辺区画の通過回数をHighValueにする。
- ②左に壁がなく、左から進入していなければ左区画の通過回数を左通過回数として得る。
- ③右、上下についても②と同じ処理を行う。
- ④周辺通過回数が全てHighValueならば進入した区画の通過回数をその向きの通過回数とする。
- ⑤周辺区画の通過回数が最小の区画を選択して進む。同じ値が存在する場合には、乱数で決定する。

2 知能レベルの設定

1回の試行で出口までの最短経路を発見することは不可能であろうが、出来るだけ短い距離を発見するためにはどうしたらよいだろうか。以前通った路や行き止まりの路などをきちんと記憶しておき、無駄な行動をとらないために、その記憶が分岐点での区画選択の判断基準に活かされなければならない。

この記憶による操作を物体にも適用させ、それ以外に、次に掲げるような処理機能を持たせ最短経路発見の一助にしたい。これらの操作は一種の学習機能の役割を果たし、人間の知能に相当するものと考えられる。もし明確な知能レベルを設定することができれば、準最短経路の発見に繋がるものと思われるが、現段階では、以下のような設定に留まった。①の有効性を示すために、それに劣るような知能レベルを設定し、3種類の知能レベルを用意した。知能レベルの程度は、①>②>③の順と思われるが、それについてはV章の2でシミュレーションにより実際に調べた。

[知能レベル]

- ①やや高い：通過方向を調べ逆方向には進まないようにする。(ループ時の判断基準にもなる)
- ②普通:追加機能無し。
- ③やや低い：通過回数を減らし、忘却機能を持たせる。

3 距離地図の作成 (最適化問題の探索アルゴリズム)

入口から行き止まり区画を除いた各区画までの距離を過去の試行を利用して求め、所謂迷路の距離地図を作成する。その作成方法を以下に説明する。

各区画(I,J)の入口から現在の試行における仮りの距離PD(I,J)は

$$PD(I, J) = C + ND(I, J)$$

となる。

但し、Cは迷路の区画間の距離を表し、C=1で一定である。また、ND(I,J)は区画(I,J)に辿り着く1つ手前の区画の距離を表す。D(I,J)を区画(I,J)での距離の決定値とすれば、ND(I,J)は路が繋がっている周辺区画の距離の最小値である。即ち、

$$\left. \begin{matrix} D(I-1, J) \\ D(I+1, J) \\ D(I, J-1) \\ D(I, J+1) \end{matrix} \right\} \text{の最小値} = ND(I, J)$$

である。

よって、BD(I,J)を区画(I,J)の前の試行の距離とすれば、現在の試行における距離D(I,J)は、

$$D(I, J) = \begin{cases} BD(I, J), & \text{if } BD(I, J) \leq PD(I, J) \\ PD(I, J), & \text{if } BD(I, J) > PD(I, J) \end{cases}$$

となる。よって、D(I,J)の集合が距離地図になるが、最適化問題という観点から、行き止まりの区画の距離は不要である。そこで、行き止まりの区画にはHighValueをいれて余分な探索をしなくて済むようにする。

この距離地図を利用すれば、出口から入口へ距離を頼りに逆に遡れば準最短距離が得られる。

V シミュレーション

1 迷路探索プログラムについて

今回、III、IV章のアルゴリズムにより迷路プログラムをグラフィック機能とマンマシンインタフェースが充実しているJavaで作成した。その具体的な処理操作内容を表1で示し、それをパソコン上で動かした様子を図5で示した。そこには20×32規模の迷路で虫が軌跡を付けて迷路を通り抜ける様子が描かれている。

2 シミュレーション内容

迷路探索を行う場合、以下の①、②に示すように2つの観点から、探索距離がどのように変化するかという問題を、次のような実験環境で、シミュレーションを行い調査した。

表1 処理操作メニュー

メニュー	処理操作内容
ファイル	開く、保存、終了
開始	初回 2回目以降→過去を頼りにして 2回目以降→経路決定 ダイクストラの方法 ($n \leq 4$) 左手法→初回 左手法→2回目以降
枠の大きさ	$(5 \times 2^{n-1}) \times (8 \times 2^{n-1})$ 、 $1 \leq n \leq 8$
作り方	自動生成、手作業
壁の修正	追加、削除
軌跡	有り、無し
物体	虫、ボール
記憶判断力	やや高い、普通、やや低い
ヘルプ	概要、遊び方

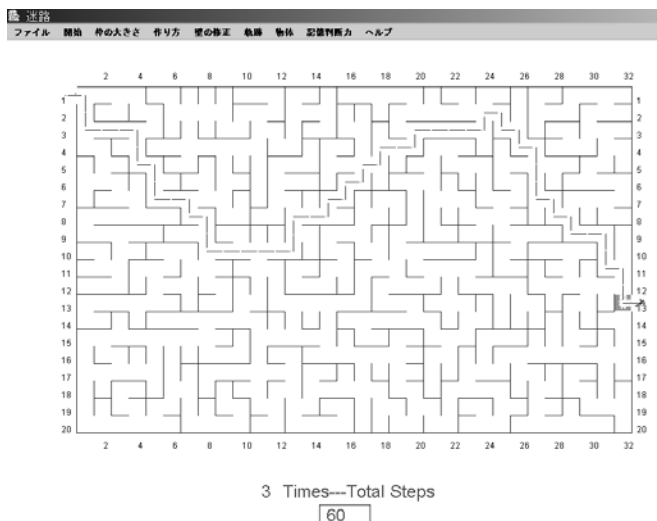


図5 規模20×32の迷路における虫の移動

[実験環境]

OS : Windows Me

CPU : Celeron(TM)

メモリ : 120MB

①知能レベルの違いによる探索距離の変化

②迷路規模の違いによる探索距離の変化

具体的には、 160×256 規模の迷路を用いて準最短経路が決定するまでの総探索距離とその初回の探索距離を調べ^(注3)、それらを10回行った時の平均値を表2に示した。また、規模による違いがあるかを、知能レベル普通で20回初回を試行した結果の平均値を表3に示した。表中の規模 n とは、 5×8 を基本として、 $(5 \cdot 2^{n-1}) \times (8 \cdot 2^{n-1})$ なる大きさを意味する。

表2 知能レベルと初回距離及び総探索距離との関係

知能レベル	やや高い	普通	やや低い
初回距離	81,687	63,282	78,615
総探索距離	163,409	171,172	183,602

表3 迷路規模と平均探索距離との関係

規模	3	4	5	6	7
平均値	1,279	7,790	19,889	76,200	430,031

3 結果の考察

表2、表3よりそれぞれ次のことが分かる。

[知能レベルと平均探索距離との関係]

①知能レベルの差が迷路探索に明確に表れていない。
・初回距離だけを考えれば、今回の方法より決まり切った左手法の方が短い、それだけで最短経路の発見に利用するのは難しい^(注4)。

・新しい機能を考える必要がある。(例えば、選択方向の選択度数を記憶しそれを判断に活かす。)

②効率的に区画情報を得るのは難しい。

・総探索距離については予想通り、知能レベルの高いと思われる順に距離が短くなっている。

・経路決定までの総探索距離が長いので、左手法等、他の手法との併用を考える必要がある。

[迷路規模と平均探索距離との関係]

・区画数が4倍になっているので、平均探索距離も4倍になるはずだが、その関係が明確ではない。規模に関係すると言うよりも、迷路の内容によって同じ規模でありながら探索距離にかなりの差が生じる。

4 迷路ゲームプログラムについて

上記1で作成した迷路プログラムを基にして、その迷路を人間自らがパソコン上で迷路の通り抜けを体験するプログラムを作成した。所謂、迷路通り抜けシミュレーションゲームである。その操作画面を図6に示した。迷路を進む途中で、自分がどこにいるか迷ってしまった場合には、表4で「ヘルプメニュー」→「現在の位置」→「軌跡あり」を選択して、開始から現在地まで通り抜けた迷路を軌跡を付けて表示することが出来る(図7)。

VI おわりに

今回、大規模な平面迷路の作成方法やその通り抜けアルゴリズムを考察し、準最短経路を発見するまでの

表4 ゲームプログラムの処理操作メニュー

メニュー	処理操作内容
ファイル	開く、保存、終了
開始	初回、2回以上、 3回以降後のPCによる経路決定 →過去の実績を元にして →自動的に
枠の大きさ	5×8、10×16、20×32、40×64
作り方	自動生成、手作業
壁の修正	追加、削除
再開	再開、中止
ヘルプ	概要、遊び方、 現在の位置→軌跡なし →軌跡あり

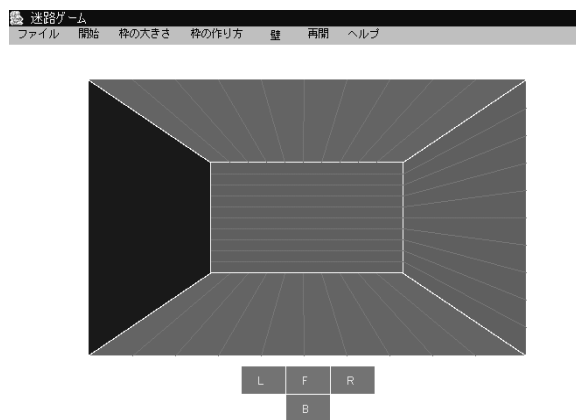


図6 迷路ゲーム操作画面

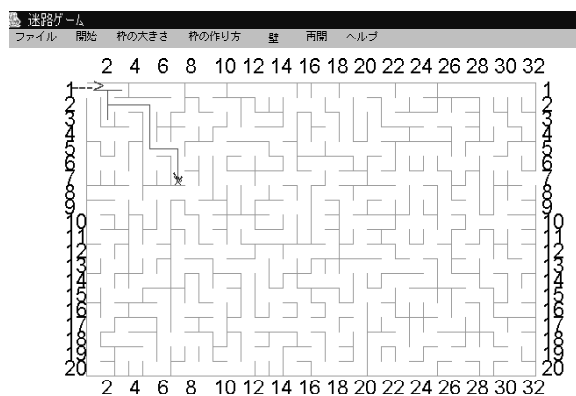


図7 迷路ゲームのヘルプ機能

シミュレーションを行った。その結果、今回考案した

手法だけを用いて、準最短経路を発見するには総探索距離が長いので改善する必要がある。その改善策として、複数の手法を併用するハイブリット方法が考えられる。それを今度は、3次元迷路探索問題に応用する予定である。

人間が迷路を通り抜けるゲームでは、CGを駆使しもっと臨場感あふれる画面に上げることが必要であるが、CGについては取り上げなかった。

[注]

(注1) グラフ理論は迷路探索では決して実用的ではない。何故かと言えば、それは全ての区画の情報が既知となったものを対象として探索するからである。しかし、最初から全体像が分ってしまえば、それはもはや迷路といえる代物ではない。迷路が迷路であると言われる由縁は、先も後も何も見えない状況が最初にあり、実際に通った路のみの情報だけしか得られないと言う制約があるからである。

(注2) ダイクストラの方法では、何も工夫をせずに単純に確保すれば規模の2乗個の重み係数が必要である。例えば、80×128規模の場合、重み係数を1バイトとすると、 $(80 \times 128)^2 = 104,857,600$ バイト必要とする。

(注3) 連続して3回同じ探索距離となった場合に、準最短経路が収束したと見なし、それで決定したものと捉える。また、総探索距離は初回から決定する前までの各試行の探索距離の和である。

(注4) 左手法による初回距離は2,306であり、準最短経路問題では初回と2回目に使用し、3回目以降には本手法を使用の方が総探索距離が短いと推測される。

[参考文献]

- (1)http://www.mukogawa-u.ac.jp/~okazaki/OK/maze/maze_j.htm
- (2)<http://www.informatics.tuad.ac.jp/tenji/tenji03/shiragami-lab/199970120/3.doc>
- (3)白井良明 人工知能の理論 コロナ社
- (4)野崎昭弘 離散系の数学 近代科学社
- (5)情報科学辞典 岩波書店
- (6)山田武士、中野良平 遺伝的アルゴリズムによるジョブショップ問題の新解法 情報処理学会人工知能研究報告 92-18 1992
- (7)北村泰彦、その他 分散探索における通信制御 情報処理学会人工知能研究報告 93-69 1993
- (8)奥村晴彦 C言語によるアルゴリズム事典 技術評論社