修 士 論 文 の 和 文 要 旨

研究科・専攻	大学院 情報システム学研究科 情報メディアシステム学専攻 博士前期課程		
氏　　名	園部　雄万	学籍番号	1350018
論 文 題 目	マイクロマウスにおける迷路探索の効率化に関する研究		

要　旨

マイクロマウスは、迷路の探索を行う小型の自律移動ロボットである。IEEE によって提唱され、日本では 1980 年より毎年、競技会が開催されている。ロボットは 16×16 区画からなる迷路を自律的に走行し、スタートからゴールまでに要した時間を競う。競技時間の制限のもと、効率的に迷路を探索し最適な経路を見つけ、素早く走行することが求められる。このようなマイクロマウスにおいて本研究では、迷路の最短経路を見つける探索（最短経路探索）を扱い、これを効率的に行う方法を明らかにする。

本研究ではまず、最短経路探索の検証に備えて迷路の生成を行った。マイクロマウスの迷路は独自の特徴を多く持ち、それらは探索の検証結果に影響を及ぼすことが考えられた。実際の競技会で用いられた迷路には限りがあったため、類似性のある迷路を生成する方法を提案し、数を補うようにした。なお、実際の迷路との類似性を評価する指標を導入し、これによって生成した迷路に類似性があることを確認した。

効率的な最短経路探索を実現するために、本研究では、2 つの提案を行った。1 つは、見つけようとする迷路の最短経路とはならないような､探索の必要がない区画を判別し探索から除外（枝刈り）するというものである。もう 1 つは、探索が必要な区画を順に辿る際にその巡回路を最適化するというものである。これは、計画的に巡回することによって、探索が必要な区画の見過ごしを抑えることを目的としたものである。10,000 種類の迷路を用いた検証の結果、探索の効率化として、枝刈りによる効果は確認できたが、巡回路の最適化による効果はわずかであった。

上記 2 つの提案に関連して、通常は探索の必要がある区画を見過ごすと再探索のために余計にコストが掛かることになるが、枝刈りを行うようにしておくと見過ごした区画が枝刈りされ、結果的に探索しなくても良くなる場合がある。つまり、枝刈りを行うことと見過ごさないことは、二律背反の関係にあると考えられた。そのため、どの程度の見過ごしによって探索を最も効率的に行えるか検証したところ、見過ごしは極力行ったほうが良いという結果を得られた。ただし、枝刈りを多数発生させる必要があり、この検証ではその方法も見出すことができた。

以上のように、最短経路探索を効率的に行う方法をいくつか明らかにすることができた。

平成 26 年度修士論文

マイクロマウスにおける迷路探索の効率化に関する研究

大学院情報システム学研究科
情報メディアシステム学専攻

学 籍 番 号 ： 1350018
氏　　名 ： 園部 雄万
主任指導教員 ： 末廣 尚士 教授
指 導 教 員 ： 工藤 俊亮 准教授
指 導 教 員 ： 田野 俊一 教授
提出年月日 ： 平成 27 年 2 月 9 日

目次

第 1 章　緒言 5
1.1　研究背景 5
1.2　関連研究 5
1.2.1　マイクロマウス 5
1.2.2　実時間探索 6
1.3　研究目的 8
1.4　論文構成 8
第 2 章　マイクロマウス 9
2.1　マイクロマウス競技 9
2.1.1　競技規定 9
2.1.2　基本的なマイクロマウスロボット 11
2.1.3　マイクロマウス迷路の特殊性 11
2.1.4　探索走行と最短走行 12
2.1.5　作戦の検討 13
2.2　マイクロマウスの探索 16
2.2.1　探索行動と探索アルゴリズム 16
2.2.2　マウスが保持する迷路情報 16
2.2.3　探索行動の分類 16
2.2.4　既存の探索アルゴリズム 17
2.2.5　動作の評価関数 19
第 3 章　迷路生成 20
3.1　迷路の評価 20
3.1.1　局所経路パターン 20
3.1.2　公式迷路の評価 21
3.2　迷路の生成 23
3.2.1　マイクロマウス迷路の制約 23
3.2.2　一般的な分野における迷路生成 23
3.2.3　公式迷路の作成方法を参考にした迷路生成方法の提案 27
3.2.4　公式迷路の作成方法を参考にした迷路生成 28
3.2.5　迷路生成方法の改良 31
第 4 章　最短経路探索アルゴリズム 34
4.1　既存の探索アルゴリズムの検証 34
4.1.1　方法 34
4.1.2　結果 35

4.1.3 考察 35
4.1.4 まとめ 36
4.2 効率的な最短経路探索を行うために 37
4.3 不要探索区画の枝刈り 38
4.3.1 不要探索区画 38
4.3.2 暫定迷路最短経路による枝刈り 38
4.3.3 検証 38
4.4 巡回路の最適化 40
4.4.1 要探索区画の最適巡回路 40
4.4.2 準最適巡回 40
4.4.3 検証 40
4.4.4 枝刈りとの併用 42
4.5 要探索区画の見過ごし 44
4.5.1 方法 44
4.5.2 結果 44
4.5.3 考察 48
4.5.4 まとめ 49
4.6 最短経路探索に関する提案のまとめ 50
第 5 章 結言 51
5.1 本研究のまとめ 51
5.2 今後の課題 52

図目次

図 2.1　マイクロマウス迷路と一般的な座標系 10
図 2.2　マイクロマウス迷路の多くは壁伝いにゴールできない 12
図 2.3　検討した競技を有利に進めるための作戦..... 15
図 3.1　局所経路パターン 21
図 3.2　公式迷路群の直進パターン..... 22
図 3.3　公式迷路群の旋回パターン..... 22
図 3.4　公式迷路群の直進と旋回の分布 22
図 3.5　初期迷路..... 23
図 3.6　各壁存在率における有効迷路数 24
図 3.7　ランダム迷路群 1 の迷路 25
図 3.8　ランダム迷路群 1 の直進パターン 26
図 3.9　ランダム迷路群 1 の旋回パターン 26
図 3.10　ランダム迷路群 1 の直進と旋回の分布 26
図 3.11　ランダム迷路群 1 と 公式迷路群の分布 26
図 3.12　ランダム迷路群 2 の迷路 29
図 3.13　ランダム迷路群 2 の直進パターン 30
図 3.14　ランダム迷路群 2 の旋回パターン 30
図 3.15　ランダム迷路群 2 の直進と旋回の分布 30
図 3.16　ランダム迷路群 2 と 公式迷路群の分布 30
図 3.17　ランダム迷路群 3 の迷路 32
図 3.18　ランダム迷路群 3 の直進パターン 33
図 3.19　ランダム迷路群 3 の旋回パターン 33
図 3.20　ランダム迷路群 3 の直進と旋回の分布 33
図 3.21　ランダム迷路群 3 と 公式迷路群の分布 33
図 4.1　or-opt 法..... 41
図 4.2　A5（近傍巡回＋見過ごし枝刈り）におけるマージンと進入区画数の平均値の関係..... 45
図 4.3　A6（準最適巡回＋見過ごし枝刈り）におけるマージンと進入区画数の平均値の関係 45
図 4.4　A5（近傍巡回＋見過ごし枝刈り）におけるマージンと移動区画数の平均値の関係..... 46
図 4.5　A6（準最適巡回＋見過ごし枝刈り）におけるマージンと移動区画数の平均値の関係 46
図 4.6　A5（近傍巡回＋見過ごし枝刈り）におけるマージンと移動区画数の標準偏差の関係 47
図 4.7　A6(準最適巡回＋見過ごし枝刈り)におけるマージンと移動区画数の標準偏差の関係..... 47

表目次

表 3.1　公式迷路群の局所経路パターン評価の統計値 22
表 3.2　ランダム迷路群 1 の局所経路パターン評価の統計値 26
表 3.3　ランダム迷路群 2 の局所経路パターン評価の統計値 30
表 3.4　ランダム迷路群 3 の局所経路パターン評価の統計値 33
表 4.1　公式迷路群における各探索アルゴリズムのゴールまでの移動区画数 35
表 4.2　ランダム迷路群における各探索アルゴリズムのゴールまでの移動区画数 35
表 4.3　A1（近傍巡回）の探索結果 39
表 4.4　A2（近傍巡回＋枝刈り）の探索結果 39
表 4.5　A3（準最適巡回）の探索結果 41
表 4.6　A4（準最適巡回＋枝刈り）の探索結果 43
表 4.7　公式迷路群における A1～A4 の探索結果のまとめ 43
表 4.8　ランダム迷路群における A1～A4 の探索結果のまとめ 43
表 4.9　A5（近傍巡回＋見過ごし枝刈り）の探索結果（マージン=0） 48
表 4.10　A6（準最適巡回＋見過ごし枝刈り）の探索結果（マージン=0） 48

第1章　緒言

1.1　研究背景

マイクロマウス競技は、ロボットに迷路を探索させ走破に要した時間を競うロボット競技である。1977 年に IEEE によって提唱され、日本においては「全日本マイクロマウス大会」という競技会が 1980 年より開催されている。他にも、全国各地で地区大会が開催されている。マイクロマウスロボット（以下、マウス）は、競技者の操作なしに、迷路を自律的に走行しなければならない。迷路は大会ごとに異なり、競技者は迷路形状に関する情報をマウスに与えることはできないため、マウスは実際に迷路を走行しながら壁の有無を調べていく。制限時間のもと効率的に迷路を探索して最適な経路を見つけ、素早く走行することが求められる。

迷路の探索手法としては種々のものが存在するが、マイクロマウス競技においては多くの場合、足立法が用いられる。これは、1984 年に足立氏によって提案された探索アルゴリズムである。具体的な検証例は少ないが、その効率の良さは多くの競技者の知るところである。しかし、足立法は、迷路のゴールに到達することを目的とした探索（以下、目的地探索）に特化したものであり、競技で求められる迷路のスタートからゴールまでの最短経路を見つけるための探索（以下、最短経路探索）には適用できない。後者の課題を実現する効率的な探索アルゴリズムは、未だ明らかにされていない。これが確立されれば、競技をより有利に進められることが期待できる。

1.2　関連研究

1.2.1　マイクロマウス

マイクロマウス競技は 35 年以上の長い歴史を持つことから、競技会における技術の変遷などを紹介した文献がいくつかある。「マイクロマウスの歩んだ路」 [1]では、主な競技規定が 30 年間ほとんど変わらないこと、主流の探索アルゴリズムの紹介、主に曲がり角での走行方法の変遷、機体の構成要素であるセンサ、モータ、バッテリなどの変遷などが紹介されている。著者は 1984 年より競技会に出場し、全日本大会では 9 回の優勝を成し遂げているという。一方、「開かれた環境が育むマイクロマウスの技術進化」 [2]では、前述の文献と同じく探索方法や機体構成要素の紹介のほかに、なぜ競技会が長く開催され続けているのかということが語られている。こちらの著者には、比較的最近のトップ選手 3 名が含まれている。いずれの文献も競技を第一線で見てきた視点で書かれており、競技参加者にとっては興味深い内容となっている。しかし、迷路の探索方法に関しては紹介のみに留まっており、足立法が最も効率が良いという旨の記述はあるが、具体的な検証はない。そして、競技では最適な経路を見つける探索（最短経路探索）が求められることや、現状、決定的な方法はないということが述べられている。

マイクロマウスの迷路探索のみを扱った研究もあるが、既存の探索アルゴリズムを改良したもの、近似的なものもしくは同一と見なせるものを再提案した例が多く見られる。Mishra ら [3]

は、壁伝い法（2.2.4.2 節）を紹介するとともに、トレモー法（2.2.4.1 節）の亜種と足立法と同一なものを提案し、シミュレーションによる検証を行っている。なお、詳細なシミュレーション条件は明記されていない。J. Cai ら [4]は、求心法（2.2.4.4 節）の亜種を提案し、公式迷路 6 種類を用いたシミュレーションによる検証を行っている。Dang ら [5]は、足立法と同一なものを提案し、シミュレーションによる検証を行っている。Z. Cai ら [6]は、足立法と同一なものを紹介するとともに、未探索の壁を確率的に考慮するようにした評価関数を用いた足立法を提案し、2 種類の迷路を用いたシミュレーションによる検証を行っている。以上のように、足立法に取って替わるような全く新しい探索アルゴリズムは提案されていない。それだけでなく、シミュレーションに用いられる迷路数が比較的少ないなど、検証結果の信頼性は低い。また、いずれもゴールに到達することを目的とした目的地探索に関する研究であり、本研究で扱う最短経路探索に関するものではない。

1.2.2 実時間探索

マイクロマウスにおける迷路探索は、実時間探索に属する。実時間探索は、エージェントの動作を逐次的に計画、実行することによって問題の解決を図るものである。探索と言えば A*アルゴリズムなどが有名であるが、こちらは問題解決までの動作列をあらかじめ計画しておくことから、オフライン探索と呼ばれる。これは、実時間探索ではエージェントは問題空間において、近傍の局所的な状態しか取得できないという制限が設けられていることによる。このような制限のある問題が考えられるようになったのは、1980 年代中頃に自律移動ロボットや実時間システムのプランニングに関する興味が高まったことが背景にあるとされる [7]。ここでの自律移動ロボットとは、機体周囲の限られた環境を観測するようなセンサを搭載した、目的地までの移動タスクをこなすものであると見られる。これはマイクロマウスロボットにも当てはまることから、実時間探索とマイクロマウスの親和性が窺える。

オフライン探索と同様に、実時間探索のアルゴリズムには様々なものが提案されている。当然ではあるが両者いずれの探索アルゴリズムも、エージェントが目標状態に到達することが保証される（完全性の保証）。しかし、例えばオフライン探索である A*が問題解決と同時に初期状態から目標状態までの最適解を得られるのに対し、実時間探索アルゴリズムは最適解を得られるとは限らない。

実時間探索アルゴリズムとして最初に提案されたのは、RTA*（Real-Time A*）と LRTA*（Learning Real-Time A*）である [8]。これらは、問題空間の各状態点に目標状態までの推定値を保持させるようにし、エージェントの移動によって取得した情報で以て推定値を更新しながら問題解決を図るというものである。このとき、エージェントは隣接点のみ評価（推定値を取得）し、エージェントがいる点の推定値を更新する。RTA*は推定値の更新に、隣接点の推定値のうち 2 番目に小さい値を用いるが、LRTA*は最小値を用いる。この差によって、RTA*は LRTA*より少ない移動で目標状態に到達することが期待でき、探索の効率が良い。その一方で、LRTA*は RTA*に比べて学習性能が良いとされるが、これに関しては後述する。

問題解決性能の向上のために、上記のアルゴリズムを応用したものがいくつか提案されてい

る。中には、初期状態と目標状態の 2 方向から探索を行う RTBS（Real-Time Bidirectional Search） [9]や、複数のエージェントを用いる MARTA*（Multi-Agent Real-Time A*） [10]、目標状態点が移動することを想定した MTS（Moving Target Search） [11] [12] [13] [14]のように、問題設定としてマイクロマウスの競技規定 [15]にそぐわないものもある。

RTA*や LRTA*を効率化する提案として、その弱点を補うということが行われる。RTA*などは、目標状態へと近付く方向に広がる袋小路に入り込むと抜け出しにくいという特徴がある。袋小路を脱出するには、それを構成する各点の推定値を入口から深まるにつれて大きくなるように更新しなければならず、エージェントの隣接点のみの狭い範囲の評価では非効率的となるためである。そこで、より広い範囲の点を評価もしくは更新対象にすることによって、袋小路から抜け出しやすくしたアルゴリズムがいくつか提案されている [16] [17] [18] [19] [20] [21] [22]。これらは単に袋小路の脱出を目的としたものではなく、より広範な情報の利用によって探索の効率化を図ったものである。利用する情報が増えると、時間的または空間的な計算量に影響が表れる。

一方で、問題空間が探索アルゴリズムの性能に与える影響ついて議論されることがある。水澤ら [23]は、ある領域に分布するパラメータにおける問題空間においてアルゴリズムの性能が急激に変化するということを鑑みて、迷路における障害物密度の影響を表す指標を導入するとともに、既存アルゴリズム（RTA*、LRTA*、MARTA*）による検証を行い、特定の障害物密度においていずれの探索結果も同様に悪化することを示している。

2.1.3 節でも述べるが、マイクロマウス迷路は他の分野の迷路にはないような特徴をいくつか持っている。そのひとつに、問題空間が比較的小さいということがある。これは、競技規定によって決まっているものである。前述のように、近年の実時間探索アルゴリズムは、その効率化と計算量はトレードオフの関係にあるが、問題空間が固定で小さいマイクロマウスは絶好の問題設定であるように思える。しかし、ここまでに述べたアルゴリズムはいずれもマイクロマウスに限定したものではなく、比較的大規模な問題空間を想定している点を考慮すれば、改善の余地があると考えられる。

本研究は、初期状態と目標状態の最適解となるエージェントの動作列を見つけるという問題を扱う。上記までのアルゴリズムは、最適解が得られるとは限らない。ただし、LRTA*は、初期状態と目標状態を往復するような反復的な試行によって最適解に収束することが示されている [8] [24]。このような LRTA*の収束性に注目した研究例は存在するが [7]、LRTA*を用いている点において、前述のとおり改善の余地が見込める。以上のように、マイクロマウスにおける最短経路探索に特化した探索アルゴリズムの研究、報告例はこれまでにない。

1.3　研究目的

本研究の目的は、マイクロマウス競技における迷路探索、特に最短経路探索に関して効率化に有効な提案を行うことである。実際の競技会で使用された 253 種類の迷路や、ランダムに生成した 10,000 種類の迷路を用いてシミュレーションを行い、提案の有効性を示す。また、その過程としてではあるが、既存の探索アルゴリズムに関する検証も行い、既存アルゴリズムについて信頼性のある検証結果を示す。

1.4　論文構成

第 2 章では、マイクロマウス競技の紹介を行い、競技を有利に進めるための作戦について検討し、その中で探索行動はどのような位置付けであるかを示す。また、マイクロマウスにおける探索とは、どのようなものであるか、既存の探索アルゴリズムの紹介とともに示す。第 3 章では、探索の検証に用いる迷路の生成を行う。第 4 章では、既存の探索アルゴリズムおよび提案したものついて検証を行う。第 5 章では、本研究のまとめと今後の課題を示す。

第2章　マイクロマウス

2.1　マイクロマウス競技

マイクロマウス競技には、ハーフ競技とクラシック競技の 2 種類がある。1980 年から実施されているマイクロマウス競技はクラシック競技に分類されるものであり、ハーフ競技は 2009 年から同時並行的に開始された。クラシックという呼び名はそのときに導入され、それ以降は単に「マイクロマウス競技」と言うとハーフ競技を指すということになった。ハーフ競技は、従来のクラシック競技に対して、迷路やマウスの大きさが半分に設定されている。クラシック競技の迷路の区画数は 16×16 であるが、ハーフ競技は可変で最大 32×32 である。制限時間はやや増加したが、探索を効率的に行うことがより一層に求められる。

しかし、迷路の大きさに違いがあっても、探索は可能な限り効率的に行ったほうが、競技を有利に進められるという点は変わらない。クラシック競技は 30 年以上の歴史があり、その間に用いられた迷路の数は 300 種類を超える。探索の研究においては、このような資産の利用は有益であると考えられる。そこで、本研究では、主にクラシック競技を扱うことにする。なお、第 4 章で探索アルゴリズムの提案を行っているが、クラシック競技に特化したものではないため、本研究の成果はハーフ競技においても有効であると期待している。

本節の以降では、マイクロマウスクラシック競技における競技規定を紹介し、競技を有利に進めるための作戦を検討することによって、その中でどのような探索行動を行うべきか明らかにする。

2.1.1　競技規定

本節では、マイクロマウス競技規定 [15]について、作戦検討および探索行動を行う上で考慮すべき事柄に絞って説明する。まず、迷路（図 2.1）に関する規定は、次のとおりである。

- 迷路は格子状に区画分けされ、16×16 区画からなる。
- 区画と区画の間には壁が存在し、壁を固定するために壁の両端には柱が存在する。
- 迷路外周にはすべて壁が存在する。
- 迷路のスタートは、四隅のいずれか 1 区画にあり、北側にだけ壁がなく、残りの三方には壁が存在する。
- 迷路のゴールは、中央の 4 区画にあり、ゴール内に壁と柱は存在しない。

競技進行に関する規定は、次のとおりである。

- 競技は、マウスがスタートからゴール到達までの走行に要した時間を競う。
- 時間計測は複数回行うことができる。ほとんど場合、5 回に設定される。
- 複数回の時間計測のうち、最短のものがそのマウスの記録となる。
- 競技者は、迷路公開前に迷路に関する情報をマウスに入力してはならない。（細かな壁配置

など、競技規定にない迷路情報を入力してはならない）

- マウスは、ゴール後も迷路を走行、探索することができる。
- スタート後、再びスタートに戻ってきたときに、1 回の走行が完了したと見なされる。
- マウスが迷路を走行中に停止し、競技者の手によってスタートに戻されると、1 回の走行と見なされる。
- 各走行で得た迷路の壁情報は、基本的にその次の走行に持ち越すことができる。
- 基本的には、既定回数の走行を完了するか、制限時間を超えると、競技終了となる。制限時間は多くの場合、5 分または 7 分である。

なお、マイクロマウスにおける「区画」とは、基本的に迷路を構成する単位区画のことを指し、この単位区画を複数含むような領域のことを指すものではない。本稿では、単位区画のことを単に「区画」と呼び、複数の単位区画を指す場合は、「区画群」もしくは「領域」と表現する。

競技規定にはマウス（ロボット）に関する記述もあるが、実際の競技会に見られる基本的なマウスについては次節で述べるため、ここでは割愛する。

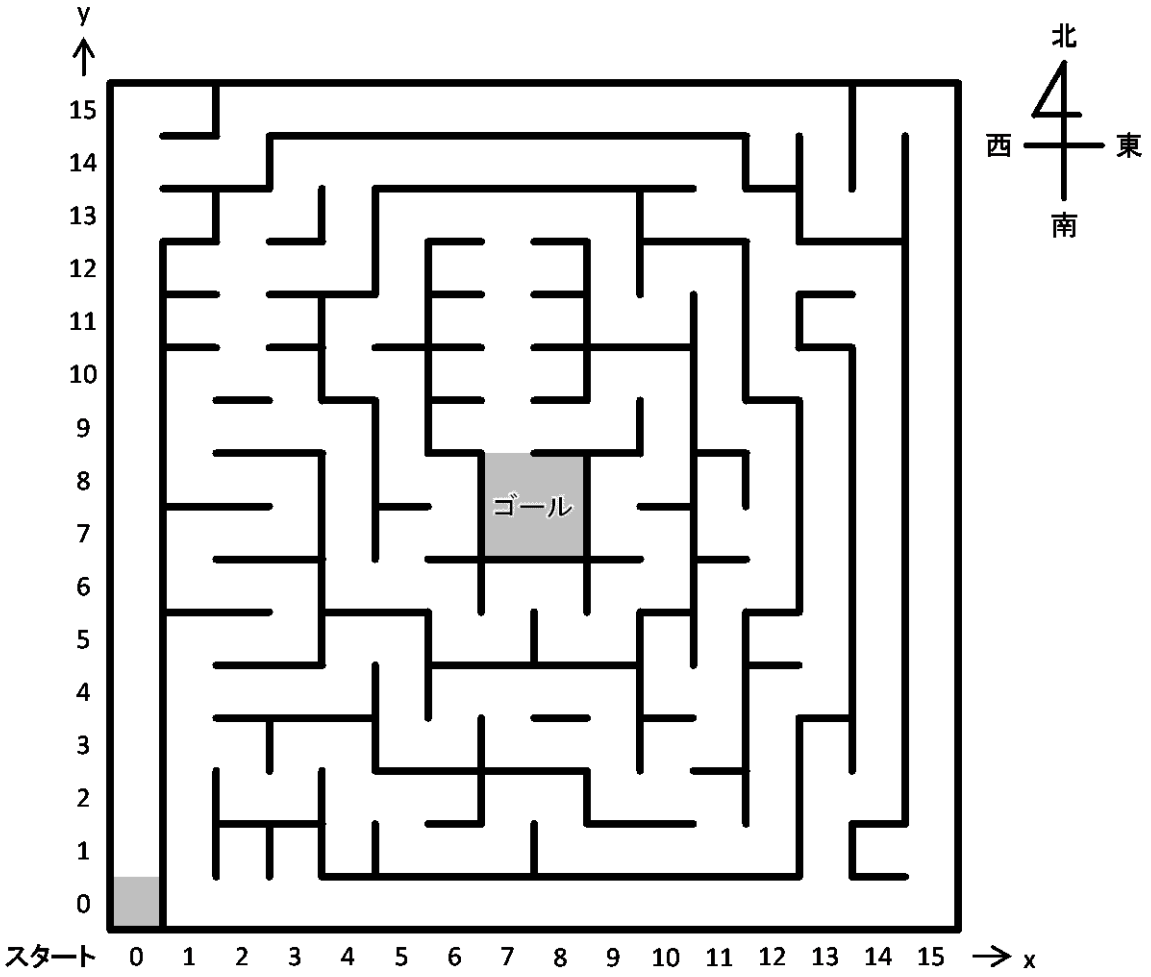


図 2.1　マイクロマウス迷路と一般的な座標系

2.1.2　基本的なマイクロマウスロボット

マウスに関して、競技規定によって定められていることは、主に「競技者の操縦なしに動作すること」と「サイズ制限」である。これを満たせば、基本的には常識の範囲内で何でも良い。しかし、実際に見られるマウスは、ハードウェアに共通の部分が多い。マウスの基本の構成要素は、次のとおりである。

- 移動用アクチュエータ：
  迷路内を走行するために用いられる。二輪の車両型が多く、DC モータやステッピングモータが用いられる。
- 壁センサ：
  迷路の壁の有無を検出するために用いられる。ロボットの位置・姿勢制御にも用いられる。非接触で高速に動作する光学式のセンサが用いられる。
- CPU：
  モータやセンサの制御、壁情報の保持、マウスの動作計画などに用いられる。汎用 IO、シリアル通信、AD 変換、タイマなどを備えたワンチップマイコンが用いられる。
- バッテリ：
  電気的な構成要素を駆動源として用いられる。比較的に小型軽量なリチウムポリマが用いられる。

マウスのサイズ制限（25 cm 四方、高さ制限なし）は、1 区画（18 cm 四方）より大きいが、実際の機体は、ある程度余裕を以て区画に収めるようにする場合がほとんどである。そのため、迷路を走行する際には、区画単位で動作を考えることが多い。

2.1.3　マイクロマウス迷路の特殊性

格子状の区画群から構成されるマイクロマウスは、様々な壁配置を取ることができる。しかし、競技会で使用される迷路は機械的にランダムに生成されているわけではなく、実際に走行するマウスの動作を考慮して、人の手によって作成される。マイクロマウスという競技を成立させるために、様々な工夫が施されている。前年までの競技内容を考慮して、競技者が越えられるか越えられないかという微妙なラインの迷路課題が出題される。これらは、主にマウスのハードウェア的な、動作特性や走行制御方法を意識したものであり、本研究では詳しく扱わない。しかし、迷路の壁配置として表れるこのようなことは、ソフトウェア的な探索を考える立場においても、適用する探索の挙動に全く影響を与えないものであるとは思えない。実際にマイクロマウスの迷路には、他の分野の迷路では見られないような特徴を多く持つ。いくつか例を挙げると次のとおりである。

- ゴールが迷路内部にある。
- スタートからゴールまでの経路が複数あることが多い。
- スタートからゴールまで、壁伝いでは到達できないことが多い。
- 袋小路が少ない。

・ 閉鎖領域があることがある。

なお、袋小路とは、ある一つの分岐区画以降、どこを通ってもスタートまたはゴールに到達できない領域を指すとする。また、閉鎖領域とは、壁で囲まれているためにマウスが進入不可能な領域を指すとする。

迷路の一般的な解法の一つに、袋小路をすべて塞ぐと、スタートからゴールまでの経路が浮かび上がるというものがある。マイクロマウス迷路においてもこれは正しいが、スタートからゴールまでの迂回路が複数あるため、必ずしも最適な経路が得られるわけではない。そのような迂回路を設定しようとすると、袋小路は必然的に少なくなる。ゴールに到達しようとするとき、袋小路は、ゴールへと通じていないため、抜け出すためには一度通った道を再び通らなくてはならないことがある。探索行動においては、既知の経路を通ることは新しい情報を得ることができないため、コストが無用に掛かることになる。また、同じく迷路の一般的な解法の一つに、左手もしくは右手沿いの壁を伝うという方法（2.2.4.2 節）があるが、マイクロマウス迷路ではこの解法への対策が行われており、絶対にゴールに到達できないように基本的には設計される（図 2.2）。これは、競技規定に記述はないが、知らない競技者はほぼいないと思われる。以上のように、ここで示しただけでも、マイクロマウスの迷路は探索に影響を及ぼす特徴を持っている。

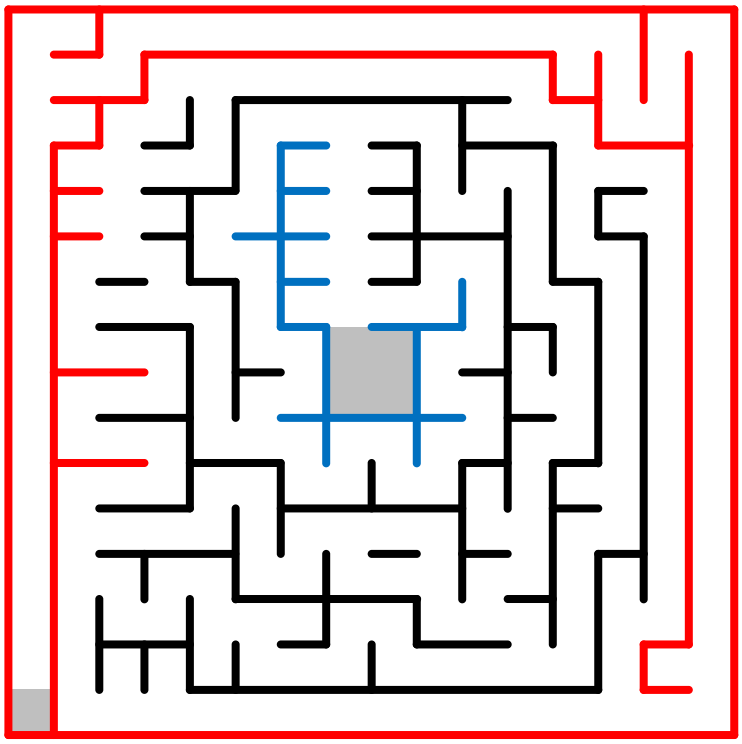


図 2.2　マイクロマウス迷路の多くは壁伝いにゴールできない
（スタート沿いの壁（赤）とゴール沿いの壁（青）が不連続のため）

2.1.4　探索走行と最短走行

マイクロマウス競技においては、マウスが壁の有無を調べる走行を「探索走行」、壁情報の更

新を行わずスタートからゴールまで高速で走り抜ける走行を「最短走行」と呼ぶ。

最短走行と言った場合は、スタートからゴールまで走るという意味合いがあるが、探索走行の場合は必ずしもそうではない。最短走行を行ったあと、ゴールからスタートまで壁の有無を調べながら走ることや、最適な経路を見つけるための走行も探索走行と言う。

また、最短走行と言っても、必ずしもそのマウスに最適な経路を通るとは限らない。これは、何がそのマウスにとっての最適であるかは、厳密には分からないということではない。迷路の探索が不十分で、そのマウスが最適と評価できるような経路がまだ見つかっていない状態であっても、既知のいずれかの経路を走行することも最短走行と呼ぶためである。

「高速で走り抜ける」ということも曖昧であるため、要するに、壁情報の更新を行うものが探索走行、行わないものが最短走行である。走行速度は基本的に最短走行のほうが速いが、スタートからゴールまでの経路が判明していないと最短走行は行えない。そのため、走行速度が遅いことも併せて、走行の安定性は探索走行のほうが高い。

2.1.5　作戦の検討

実際の競技で競技者が取る作戦には様々ものがある。例えば、1 回目の走行では、スタートからゴールまでの往路、復路ともに探索走行を行い、2 回目以降の往路は最短走行、復路は探索走行を行う、というものがある。これは、2.2.4.5 節で述べる探索アルゴリズムの足立法が、1 回の試行では最適な経路を導出できないことがあるため、重ねて探索を行うことによって徐々に最適な経路を通るようにしてより良い記録を出す、というものである。最適な経路を見つけるための探索走行では、長距離を走ることになるためタイヤに埃が溜まりやすく、走行が不安定になり壁に衝突する危険が大きい。現在の競技会において、マウス自身が壁に衝突したことを認識した上で、自力で復帰するという機能が実装されている例はほとんどない。壁に衝突するとハードウェア的な損害の他に、探索走行によって得た壁情報や自己位置情報が乱れることがあるため、マウスは競技者に回収され 1 回分の走行を消費することになる。先の作戦のように、最適な経路を見つけるための探索走行を分割すると、長距離を走ることが少なくなるため衝突のリスクを減らすことができる。なお、各走行間では、競技者はタイヤの汚れを取り除くなど、軽微なメンテナンスを行える。しかし、この作戦では、最適な経路を見つけるまでに、制限のある走行回数を多く消費してしまう。迷路の形状によっては、制限回数内の探索走行で最適な経路を発見できないこともあり得る。

その他の作戦としては、1 回目の走行からゴールに到達することよりも最適な経路を見つけることを優先して探索走行を行い、2 回目以降で最適な経路を通る最短走行を行い、各回で走行パラメータ（直進経路や旋回経路における速度や、加速度）を調整してより良い記録を出す、というものがある。1 回目の走行においては、最適な経路を探す過程でゴールに到達する。ときには、ゴールせずに探索を完了する場合もある。これは、壁配置さえ確認できれば、ゴール区画に進入せずとも、スタートからゴールまでの経路が確定するからである。このような作戦は、2 回目以降の最短走行で記録を出すことを前提にしており、1 回目で良い記録を出すことはほぼ考慮しない。探索走行よりも最短走行のほうが良い記録となることがほとんどであるため、1 回目の走行

を最適な経路の発見に消費するというのは理にかなった方法である。しかし、長距離の探索を行うことになるため、探索走行の制御の安定性をある程度に確保する必要がある。壁への衝突の他に、探索走行時の走行速度を積極的に上げることが難しいと考えられ、迷路によっては最適な経路の探索に時間を使いすぎてしまい、その後の最短走行を十分に行えなくなる可能性が高まる。マウスの走行制御によほどの自信がなければ取れない、上級者向けの作戦である。

以上のことを考慮して、競技を有利に進められる作戦を考える。まず、競技の目的である、より良い記録を出すことを優先するようにする。そのためには、時間計測を伴う最短走行を多く行うことが望まれる。それには最適な経路を見つける必要があるが、迷路の探索に対して時間計測には回数制限があるため、最適な経路は 1 回目の走行で見つけておいたほうが良い。これは、上記の後者の作戦と同様であるが、1 回目の走行の往路ではゴールに到達することを優先して探索走行を行うようにする。探索においては、その探索の目的を考慮した方法で動作を行ったほうが効率的であるため、走行距離を抑えることができる。同時に、探索走行は最短走行ほど速くはないが走行の安定性が高いため、できる限り良い記録を残すことができ、2 回目以降の最短走行が何かの理由で全て失敗したときの保険となる。そして、1 回目の走行の復路では、最適な経路を見つけるための探索走行を行うようにする。まとめると、この作戦のフローチャートは、図 2.3 のようになる。どの走行を選択するかは、人が判断してもマウスが判断しても良い。

なお、いずれかの走行中に壁に衝突するなどして、マウスの走行が停止した場合は、復帰動作は行わず、競技者の手でスタートに戻すとする。これは、復帰動作が実際に行われる例が少なく、実装も決して容易ではないと予想できるためである。走行回数を節約するには、有効なものではある。

また、スタート時に最適な経路が未知であっても、ゴールまでの経路が 1 つ以上既知であった場合は、ゴールまでの探索走行と最短走行のどちらを行っても良いというようにした。これは、汎用的な判断基準を示すことは難しいためである。最短走行を選択した場合は、1 回目の走行記録より良いものを期待できるが、安定性において不安が残り、走行に失敗してしまうと復路での探索走行を行えなくなり、経路の更新がなく同様の失敗を繰り返す可能性がある。一方の探索走行を選択した場合は、記録は期待できないものの、スタートからゴールまでの経路が新しく見つかる可能性があり、最短走行選択時の失敗の繰り返しを抜け出せる。何度もこの場面に陥るなら、最短走行と探索走行を交互に行っても良いかもしれないが、走行回数を消費することは好ましくない。走行の安定性に依存するため、競技会直前までにどこまで安定性を高めることができたかに応じて、その場で判断することが望ましい。

以上に、競技を有利に進めるための作戦を示した。実際の競技会において、この作戦を取る競技者は多いように見受けられる。ただし、この作戦においてもやはり最適な経路の探索で、長距離の走行が余儀なくされる。この作戦を実行するためには、最適な経路を見つける探索を効率的に行うことが求められる。第 4 章では、これを扱っている。この作戦における探索行動は、まずゴールまでの探索走行を行ったあと、最適な経路を見つけるための探索走行を行うというものである。第 4 章に備えて先に述べておくと、最適な経路を見つける探索走行の前には、一度ゴールに到達しており、スタート-ゴール間の経路が 1 つ以上見つかっているという前提があること

が重要である。

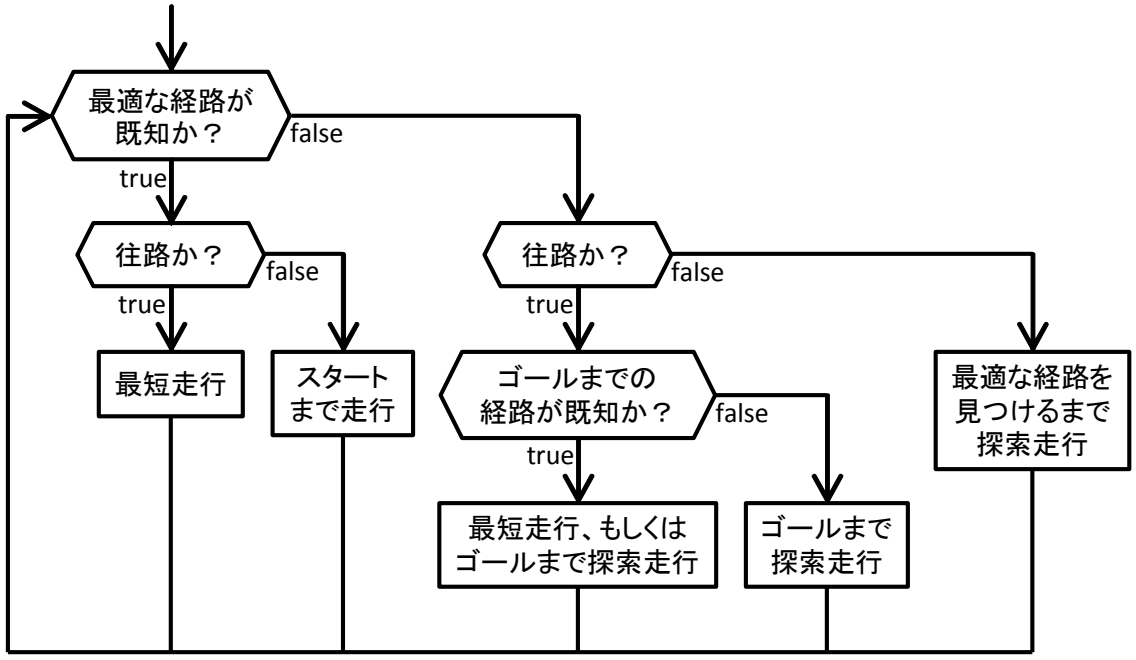


図 2.3　検討した競技を有利に進めるための作戦（スタート前にどの走行を行うかの選択）

2.2　マイクロマウスの探索

2.2.1　探索行動と探索アルゴリズム

マイクロマウスにおける探索行動は、迷路を走行しながらセンサによって壁の有無を検出し、壁配置の情報を獲得していくものである。これは、次の処理を逐次的に繰り返すことによって実現する。

(1) 状態の更新（壁の検出および壁の有無情報の記憶、自己位置の更新）
(2) 動作の計画（進行方向の決定）
(3) 動作の実行（直進や旋回）

マイクロマウスにおける探索アルゴリズムは、上記の(2)に位置するものである。つまり、探索アルゴリズムは、(1)で得た壁情報や現在地情報を入力として受け取り、実行可能な動作の候補の中から、探索行動の目的（2.2.3 節で述べる）を達成するために必要な動作を選んで組み合わせ、(3)に出力するということを行う。

2.2.2　マウスが保持する迷路情報

マイクロマウスが走行する迷路は、競技規定によって規格化されているため、これを抽象的に表現することは容易である。迷路の大きさ、つまり設置できる壁の位置や数は決まっている。あとは、壁の状態を表す必要がる。壁が取り得る状態は、次の 3 つである。

・ 未知：壁の有無が分からない状態
・ なし：壁が設置されていない状態
・ あり：壁が設置されている状態

多くの場合、マウスは、マイコンなどのメモリ内部に、上記の状態を取る壁で構成された仮想的な迷路を保持している。探索行動は、この仮想的な迷路の更新を行いつつ進行するものである。

2.2.3　探索行動の分類

マイクロマウスにおける探索行動は、達成しようとする目的に応じていくつかに分類できる。

・ 目的地探索：ゴールなどの任意の目的地に到達する。
・ 全面探索：進入可能な迷路の全区画を訪問する。
・ 最短経路探索：スタートとゴールなどの任意の 2 地点間の最短経路を見つける。

いずれの探索においても、迷路の壁情報の取得は行う。なお、本節以前に述べていた「最適な経路を見つける探索」は、最短経路探索のことである。

次節で示すように探索アルゴリズムには様々な種類があるが、それぞれが上記のいずれか一つにしか適用できないわけではない。ある探索アルゴリズムは、上記の目的を複数達成できる場合がある。例えば、足立法は、確実に目的を達成できるものに限れば、適用できるのは目的地探索のみであるが、トレモー法は、上記 3 つの目的をすべて達成することができ、いずれの探索に

も適用できる。これについては、次節でも述べる。

2.2.4　既存の探索アルゴリズム

マイクロマウスの迷路探索に利用できる既存の探索アルゴリズムを紹介する。いずれの探索アルゴリズムにおいても明確な定義が存在しないため、本研究ではここで示す定義に基づいて、これらの探索アルゴリズムを扱うことにする。

2.2.4.1　トレモー法

トレモー法は、フランスの数学者 M・トレモーによって提案された探索アルゴリズムである[25]。別名として、拡張左手法と呼ばれることもあるが、本研究ではトレモー法と拡張左手法は、別の探索アルゴリズムとして扱う（2.2.4.3 節）。トレモー法のアルゴリズムは、次のとおりである。

(1) マウスが進入した区画を記録するようにしておき、現在地に隣接する 4 つの区画のうち、「現在地との間の壁が『なし』である」かつ「未進入である」区画の方向へ進行する。
(2) (1)で進行方向を決定できない場合は、それまでに通ってきた区画をバックトラックするように進行する。

なお、(1)において、進行できる方向が複数ある場合については、ここでは特に定義を与えない。つまり、進行方向の候補なら、いずれに進んでも良いとする。

トレモー法は、迷路上の進入可能なすべての区画を訪問することができる。このため、目的地探索、全面探索、最短経路探索のいずれにも適用することができる。ただし、しらみつぶしに探索を行うものであるため、いずれの探索においても効率が良いとは言えない。また、全面探索に適用した場合、アルゴリズムの特性上、進入可能な区画をすべて訪問しバックトラックによってスタートに戻るまでには、必ず$\Big(\big(\text{進入可能な区画数}-1\big)\times 2\Big)$個の区画の移動を要する。

2.2.4.2　壁伝い法

左手法もしくは右手法は、いずれかの手を迷路の壁に沿わせて伝っていくとゴールに到達できると言われる有名な探索アルゴリズムである。本研究では、この 2 つをまとめて壁伝い法と呼ぶことにする。左手法のアルゴリズムは次のようなものであるが、右手法はこれの左右対称である。

(1) マウスの左の壁が「なし」であれば、その方向に進行する。
(2) (1)を満たさず、マウスの前の壁が「なし」であれば、その方向に進行する。
(3) (2)を満たさず、マウスの右の壁が「なし」であれば、その方向に進行する。
(4) (3)を満たさず、マウスの後の壁が「なし」であれば、その方向に進行する。

壁伝い法は、探索によって得られた過去の壁情報を必要とせず、マウスがいる区画の壁情報があれば十分である。しかし、壁伝い法では、スタートの壁とひと続きになっている区画しか訪問することができない。マイクロマウスの迷路は、スタートとゴールの壁はひと続きではないため、壁伝い法ではゴールできない。これは、意図的に設計されているものであり、実際の競技会にお

いて、壁伝い法を用いることは適切でない。このようなことから壁伝い法は、2.2.3 節のいずれの探索に必ずしも適用できない。

2.2.4.3　拡張壁伝い法

本研究では、拡張左手法と拡張右手法をまとめて拡張壁伝い法と呼ぶ。トレモー法では、進行方向の候補が複数ある場合はいずれでも良いとしたが、拡張壁伝い法は壁伝い法をもとに一意に決定できるようにしたものである。ゴールに到達できるように、壁伝い法をトレモー法的に拡張したと捉えることもできる。拡張左手法のアルゴリズムは次のとおりであるが、拡張右手法はこの左右対称である。

(1) マウスが進入した区画を記録するようにしておき、現在地に隣接する 4 つの区画のうち、「現在地との間の壁が『なし』である」かつ「未進入である」区画の方向を取得する。
(2) 取得した方向のうちから、左手法的に一方向を選択し、進行する。
(3) (2)で進行方向を決定できない場合は、それまでに通ってきた区画をバックトラックするように進行する。

拡張壁伝い法は、トレモー法と同じく、迷路上の進入可能な区画をすべて訪問することができる。同様に、目的地探索、全面探索、最短経路探索のすべてに適用できる。

2.2.4.4　求心法

求心法は、トレモー法を基本に、進行方向の候補が複数ある場合は、ゴール（目的地）がある方向を優先的に選択するようにしたものである。目的地の方角は、現在地とのマンハッタン距離を用いて判断する。実用上は、マンハッタン距離に限らず任意の距離関数を用いることができるが、本研究では簡単のためにマンハッタン距離で考える。求心法のアルゴリズムは、次のとおりである。

(1) マウスが進入した区画を記録するようにしておき、現在地に隣接する 4 つの区画のうち、「現在地との間の壁が『なし』である」かつ「未進入である」区画を取得する。
(2) 取得した区画それぞれについて、その区画から目的地までのマンハッタン距離を計算する。
(3) マンハッタン距離を計算したうち、最も小さい値の区画の方向へ進行する。
(4) (3)で進行方向を決定できない場合は、それまでに通ってきた区画をバックトラックするように進行する。

なお、(3)において、進行方向を一意に決定できない場合があるが、ここでは特に定義を与えない。進行方向の候補なら、いずれに進んでも良いとする。

求心法は、トレモー法と同じく、迷路上の進入可能な区画をすべて訪問することができる。同様に、目的地探索、全面探索、最短経路探索のすべてに適用できる。目的地に関する情報を考慮するため、拡張壁伝い法に比べて効率が良い（4.1 節）。

2.2.4.5　足立法

足立法は、1984 年に足立氏によって提案された探索アルゴリズムである。未進入区画の考慮やバックトラックといったトレモー法的な方法を用いず、探索で得た壁情報やゴール（目的地）に関する情報を積極的に利用することによって、目的地に到達することに特化したものである。足立法のアルゴリズムは次のとおりである。

(1) 現在地から目的地までの理想最短経路を導出する。
(2) 導出した理想最短経路の方向に進行する。

なお、理想最短経路とは、「未知」の壁は「なし」と仮定した迷路上における、任意 2 地点間の最短経路を表すとする。また、(1)で導出する理想最短経路は、複数存在する場合があるが、その際の処理は、ここでは特に定義を与えない。いずれの理想最短経路を選んでも良いとする。

足立法は、実際の競技会で最もよく利用される探索アルゴリズムである。その理由は主に、「効率が良い」、「拡張性が高い」、「少しの修正で最短走行に利用可能」などがある。「効率が良い」ことについては、4.1 節で検証を行う。「拡張性が高い」とは、進行方向の決定に利用する理想最短経路は、様々な評価関数で導出することができるためである。例えば、実際の競技会では、経路の長さを区画数で評価するものが基本的に用いられるが、旋回の多い経路のコストを大きく見積もる評価関数なども用いられることがある。これによって、旋回よりも直進が得意であるといった、機体固有の動作に応じて探索行動を行うことができる。また、「少しの修正で最短走行に利用可能」における修正とは、前述の(1)において理想最短経路を求めるのではなく、「未知」の壁を「あり」と仮定した最短経路を求めるようにすることである。「未知」壁を「あり」と扱うと、未探索の領域には経路を引けなくなるため探索済みの経路しか通らず、壁情報の更新を行わなくても良くなり、最短走行を実現することができる。プログラム上でこのような変更を行うことは容易であるため、足立法による探索走行でゴールに到達できるようになった競技初級者が、即時的に最短走行を行うために利用することが多い。

足立法は、目的地に到達することに特化した探索アルゴリズムであるため、目的地探索にはもちろん適用できる。全面探索に利用されることもあるが、足立法単体では不可能であり、何かしらの探索アルゴリズムの構成要素として用いられるということである。また、足立法は一回の試行では、迷路のスタートからゴールまでの最短経路（以下、迷路最短経路）を見つけられないことがあるため、必ずしも最短経路探索に適用できるわけではない。複数回の試行、つまり足立法でスタートとゴール間を往復すると迷路最短経路を見つけられるとされるが [1]、複数存在する場合はそのすべてを見つけることは基本的には保証されない。足立法で用いる動作の評価関数を工夫すれば、それが可能になることはある。

2.2.5　動作の評価関数

ここまでに述べてきた「最適な経路」または「最短経路」としては、ロボットの動作を評価する関数に応じて様々なものを考えることができる。本研究では簡単のために、区画を単位とした移動区画数を評価値（コスト）とする関数によって得られるとする。

第3章　迷路生成

2.1.3 節で述べたとおり、マイクロマウスの迷路は、その他の場面における迷路が持たないような特徴を多く持つ。何かしらの偏りがある問題空間に適用される探索アルゴリズムは、そのような問題空間で検証を行ったほうが相応しいと考える。しかし、実際の競技会で使用された迷路（以下、公式迷路）を用いて、過去に何度かの探索のシミュレーションを行ったところ、探索アルゴリズム間の差異を識別することが難しい場面があった。これは、シミュレーションに用いる迷路の数を増やすことによって解決できると考える。公式迷路は、基本的に人の手によって作成されるため、同様の方法で迷路数を確保することは困難である。そこで、公式迷路に似た迷路を自動生成する方法について検討し、その方法を用いて迷路数を補うことにする。

3.1　迷路の評価

公式迷路に似た迷路を生成するために、迷路の特徴を評価する指標を導入する。この評価指標の結果を以て、迷路同士の類似度を測る。

3.1.1　局所経路パターン

迷路の特徴を捉えるための評価指標は数多くのものを考えることができるが、ここでは「経路」に注目する。公式迷路は、走行するマウスの動作を考慮して経路を中心に設計されるためである。経路は、さまざまな形状を取り得る上に、長さに応じてその複雑さも増す。しかし、いかに長い経路であっても、それはより短い経路の組み合わせであると見なすことができる。マウスが実行できる動作は機体によって異なるが、どのようなものであっても「直進」と「旋回」が基本である。そこで、「直進」と「旋回」に大別されるような局所的な経路パターンを考えることにする。これは、図 3.1 のような 6 種類である。この経路パターンそれぞれが、どの程度出現するかを数えることによって、迷路を評価する。なお、パターンの経路上に「壁がない」ことのみを出現の判断とし、経路沿いなど経路外の壁の有無は考慮しないことにする。また、閉鎖領域中のパターンはカウントしない。

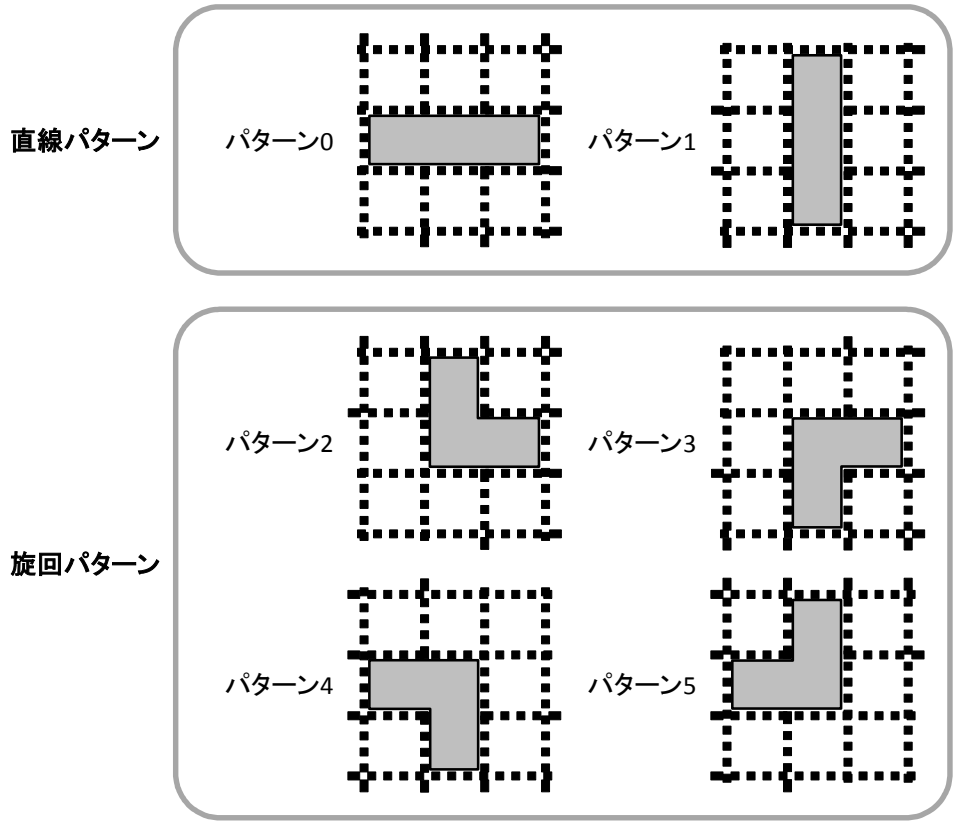


図 3.1　局所経路パターン

3.1.2　公式迷路の評価

前節で提案した評価指標である局所経路パターンによって、公式迷路がどのように評価されるか確認する。

3.1.2.1　方法

253 種類の公式迷路ひとつひとつに対して、6 種類の経路パターン（図 3.1）それぞれが出現する回数を数えた。

3.1.2.2　結果

直進パターン（パターン 0、1）の出現回数のヒストグラムを図 3.2 に、旋回パターン（パターン 2～5）の出現回数のヒストグラムを図 3.3 に示し、それぞれの最小値、最大値、平均値、標準偏差を表 3.1 に示す。また、各迷路における直進パターンと旋回パターンの出現回数の分布図を図 3.4 に示す。なお、ヒストグラム中の縦点線は平均値を、分布図中の直線は点群の回帰直線を表す。

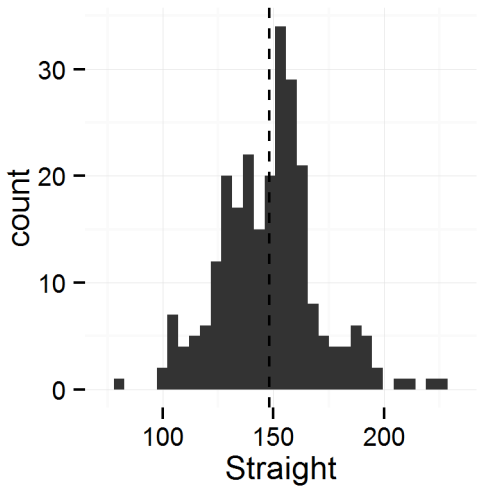


図 3.2　公式迷路群の直進パターン

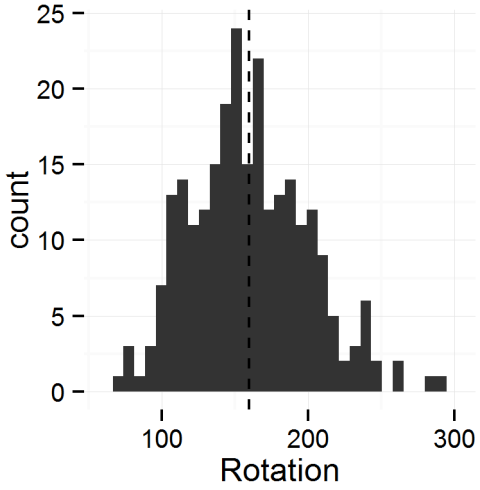


図 3.3　公式迷路群の旋回パターン

表 3.1　公式迷路群の局所経路パターン評価の統計値

	最小値	最大値	平均値	標準偏差
直進パターン	80	226	148.1	22.1
旋回パターン	73	294	159.6	40.1

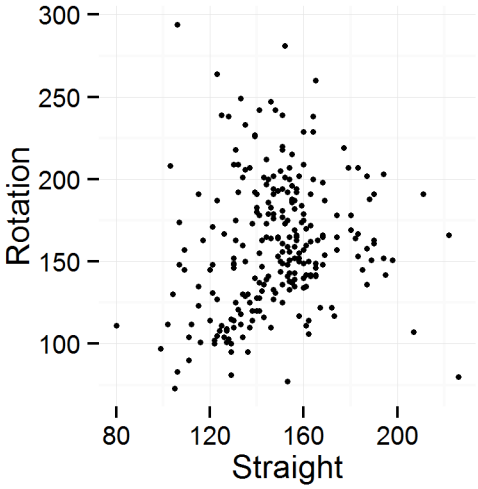


図 3.4　公式迷路群の直進と旋回の分布

3.2　迷路の生成

ここでは、迷路探索の検証で用いるために、公式迷路に似た迷路を生成する。生成した迷路は、3.1.1 節で提案した評価指標によって評価を行い、公式迷路との類似性の有無を確認する。

3.2.1　マイクロマウス迷路の制約

マイクロマウスの迷路は、競技規定（2.1.1 節）によって満たすべき条件がある。迷路生成に関わる要点のみを説明すると、次のとおりである。

- 迷路外周には必ず壁が存在する。
- スタート区画の北には壁がなく、東には壁がある。
- 複数区画からなるゴール内には壁がない。
- 柱が 1 枚も接続しない柱があってはならない。ただし、ゴール内と閉鎖領域内は除く。
- スタートからゴールまでの経路が 1 つ以上存在する。

なお、競技規定には、スタートからゴールまでの経路が必ず存在するとは明記されていないが、競技の特性から明らかであるため、他と同様なものとして扱う。

16×16 区画の迷路には、最大で 544 枚の壁を設置することができる。しかし、上記制約のために、自由に壁を設置できるのは 474 箇所である。以降の迷路生成では、この 474 の壁設置可能位置に壁を配置することを考える。なお、上記の制約を満たす迷路を有効迷路と呼ぶことにする。また、図 3.5 のように、壁設置可能位置に壁がない迷路を初期迷路と呼ぶことにする。

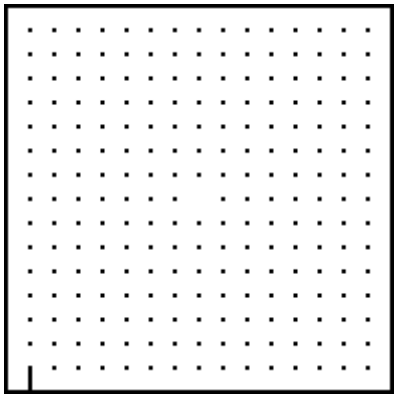


図 3.5　初期迷路

3.2.2　一般的な分野における迷路生成

予備実験として、既存の迷路生成法を用いて、公式迷路に似た迷路が生成できるか確認する。迷路は、一般的な探索アルゴリズムの検証においても用いられることがある。その際には、任意の確率で以て障害物を発生させ、迷路を構成するという方法が用いられることがある。ここでは、これと同様の方法で迷路を生成し、局所経路パターンによる評価を試みる。

3.2.2.1　生成方法

任意の確率（以下、壁存在率）で壁を設置し、迷路を構成する。迷路生成の手順は、次のとおりである。

(1) 初期迷路を用意する。
(2) 壁設置可能位置ひとつひとつについて、任意の壁存在率で壁を設置する。

実際の生成は、次のように行った。

・ 壁存在率： 10 %～90 %（5 %刻み）
・ 生成数： 壁存在率ごとに 1,000,000 個ずつ

なお、ここで生成した迷路群は、「ランダム迷路群 1」と呼ぶことにする。

3.2.2.2　生成結果と考察

ここでの生成方法では、スタートからゴールまでの経路（以下、迷路経路）が存在しないことがあり得る。また、閉鎖領域以外で、壁が 1 枚も接続しない柱が存在することもあり得る。このようなことがなく、マイクロマウス迷路として有効な迷路が、各壁存在率においてどの程度生成されたかを図 3.6 に示す。

有効迷路が 1 個以上生成されたのは、壁存在率が 45 %～80 %のときであった。また、1,000,000 個あたりの有効迷路生成率が最も高かったのは、壁存在率が 60 %のときであった。有効迷路のうち、全く同じ壁配置のものは一組もなかった。実際に生成された迷路の例を図 3.7 に示す。いずれの迷路においても、大部分を閉鎖領域が占めている。迷路経路は短く、数も少ない。また、壁存在率が異なるいずれにおいても最短のマンハッタン距離で結ぶような似た迷路最短経路となっている。壁存在率の違いは、主に閉鎖領域に表れていることが分かる。公式迷路とは、見た目がかけ離れた迷路が生成された。

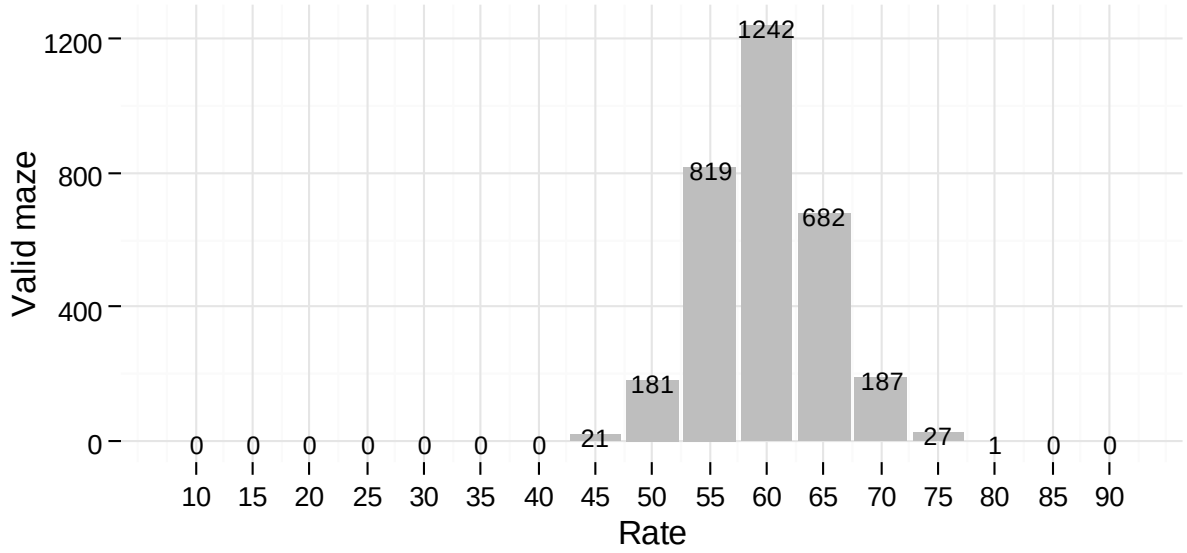
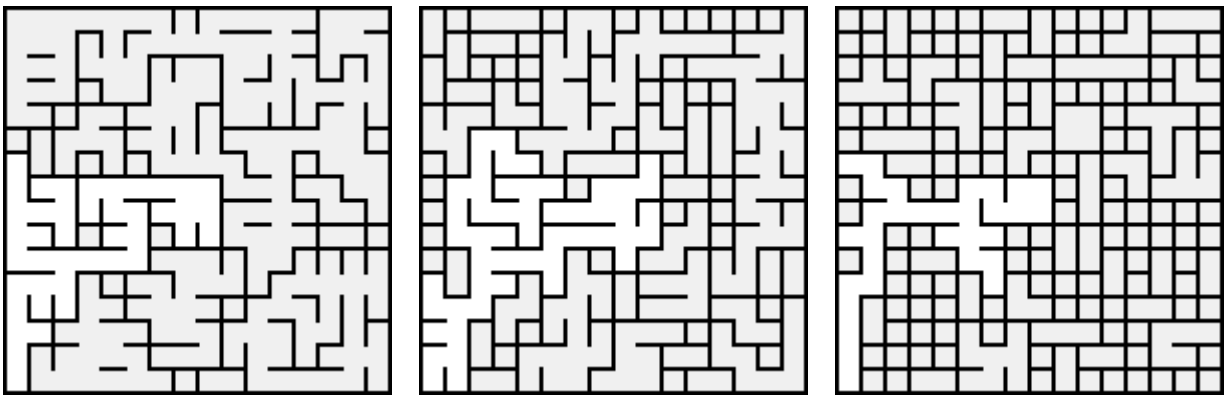


図 3.6　各壁存在率における有効迷路数



(a) 壁存在率 45 %の例　　(b) 壁存在率 60 %の例　　(c) 壁存在率 80 %の例

図 3.7　ランダム迷路群 1 の迷路

3.2.2.3　生成迷路の評価

3.1.2 節での公式迷路の評価と同様に、ランダム迷路群 1 を局所経路パターンによって評価した。結果を、3.1.2 節と同様の形式で、図 3.8～図 3.10、表 3.2 に示す。また、図 3.11 には、直進パターンと旋回パターンの出現回数の分布について、公式迷路群のプロットと重ねたものを示す。

直進パターンと旋回パターンの出現回数の平均値（表 3.2）は、いずれも公式迷路群（表 3.1）と比べて、小さかった。これは、ランダム迷路群 1 は、経路パターンをカウントしない閉鎖領域を多く含むことが寄与していると考えられる。また、いずれのパターンの分布も、公式迷路群の分布とほとんど重なりがなかった。これは、図 3.11 からよく分かる。そして、直進パターンと旋回パターンの分布（図 3.10）を見ると、両者に正の相関があることが分かる。このときの回帰直線の傾きは、およそ 2 である。これは、6 種類の経路パターンはいずれも、ある 1 区画の 4 枚の壁のうちの 2 枚が存在しないことによって決定されるものであり、直進パターンと旋回パターンの種類の数が、1 : 2 であることによる。図 3.4 に示したとおり、公式迷路群にこのような相関は見られなかった。以上のように、公式迷路に似た迷路を生成することは困難であることと確認できた。

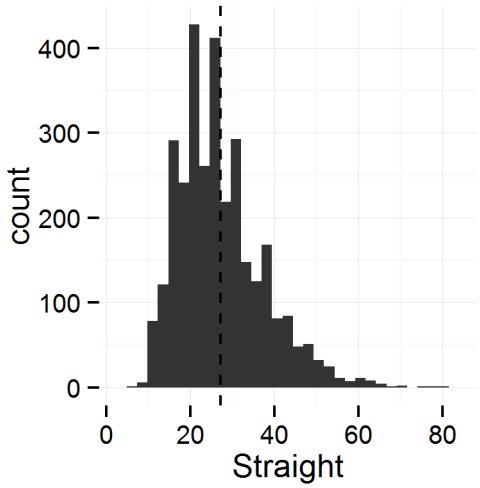


図 3.8　ランダム迷路群 1 の直進パターン

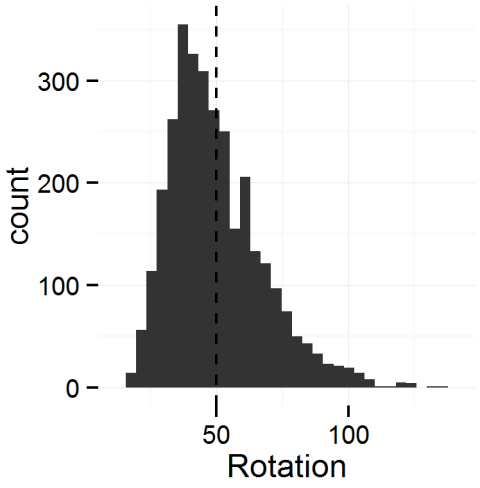


図 3.9　ランダム迷路群 1 の旋回パターン

表 3.2　ランダム迷路群 1 の局所経路パターン評価の統計値

	最小値	最大値	平均値	標準偏差
直進パターン	7	81	27.2	10.1
旋回パターン	16	134	50.0	17.7

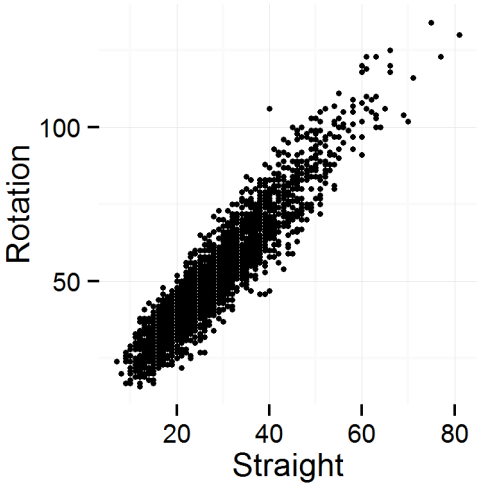


図 3.10　ランダム迷路群 1 の直進と旋回の分布

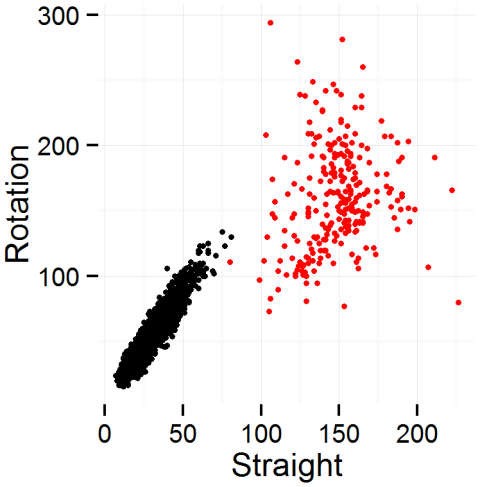


図 3.11　ランダム迷路群 1（黒）と公式迷路群（赤）の分布

3.2.3　公式迷路の作成方法を参考にした迷路生成方法の提案

前節の迷路生成方法は、迷路をランダムに得ることはできたが、「迷路経路が短く、多様性がない」、「閉鎖領域が大きい」、「マイクロマウス迷路の制約を満たす迷路が生成されにくい」という問題があった。本節では、人が迷路を作成する際の方法を参考にして、この問題を解決するための迷路生成方法を提案する。

3.2.3.1　最短経路保存法

公式迷路の作成方法は、人によって異なると予想されるため一概には言えないが、多くの場合は最短走行用の経路を決めるところから始めると考えられる。最短走行用経路の壁を配置したあと、探索走行で通り得る経路を考慮しつつ、決定し切れていない壁設置可能位置に壁を配置するものと予想される。このような作成方法を参考にするときに重要となるのは、配置したいと考えている経路を保ちつつ壁を配置するということである。これは、その経路上に壁を配置して経路を塞がないようにするということより、保存したい経路より短い迂回路が残らないように壁を配置するということが要である。ここでは、次のような処理によって実現することを考えた。

(1) 経路を入力する。
(2) 入力経路を通る壁を取り除く。
(3) 入力経路の始点と終点を結ぶ最短経路をすべて取得する。
(4) 取得した最短経路のいずれかが通る壁設置可能位置をすべて取得する。
(5) 取得した壁設置可能位置のうち、「壁が設置されていない」かつ「入力経路を通らない」ものの中から、ランダムに 1 つを選択し壁を設置する。
　壁を設置できなければ、処理を終了する。
(6) (3)へ戻る。

以上の処理を、最短経路保存法と呼ぶことにする。この方法を用いると、任意の経路についてその経路を保ちつつ、それと同じかより短い経路を塞ぐように壁を配置することができ、人による迷路作成をある程度に再現できる。

また、壁を設置する範囲を限定することによる利点もある。3.2.2 節の方法は、迷路の広範囲に無秩序に壁を配置するために、マイクロマウス迷路の制約（3.2.1 節）を満たすことが困難であったと見られる。上記の方法では、任意の経路が塞がれることがない上に、壁の配置範囲は任意の経路周辺に限られるため、そのようなことは起こりづらいと考えられる。

3.2.3.2　柱有効化法

3.2.3.2 節の最短経路保存法は、マイクロマウス迷路の制約（3.2.1 節）を完全に満たすことを考慮したものではないため、迷路上に壁が一枚も接続しない柱（無効な柱）が残り得る。これは、次のような処理によって、解決する。

(1) 無効な柱に隣接する壁設置可能位置をすべて取得する。
(2) 得られた壁設置可能位置の中から、ランダムに 1 つを選択し壁を設置する。

壁を設置できなければ、処理を終了する。

(3) (1)へ戻る。

以上の処理を、柱有効化法と呼ぶことにする。これによって、最短経路保存法では設置し切れない壁を補う。

3.2.4 公式迷路の作成方法を参考にした迷路生成

本節では、提案した最短経路保存法と柱有効化法を用いて、迷路の生成を行う。3.2.3.1 節で述べたとおり、公式迷路は最短走行用の経路が存在するが、それ以外に迂回路として部分的な経路が多く存在している。ここで行う迷路生成では、迷路最短経路を以て最短経路保存法を適用するだけでなく、複数の部分経路による最短経路保存法も行う。これによって、公式迷路に似た迷路を生成することを試みる。

3.2.4.1 生成方法

最短経路保存法の適用においては配置したい経路を用意する必要があるが、これをゼロから生成することは容易ではない。そこで、公式迷路群の中からサンプリングすることによって、必要な経路を得ることにする。これを踏まえて、ここで行う迷路生成の手順は、次のとおりである。

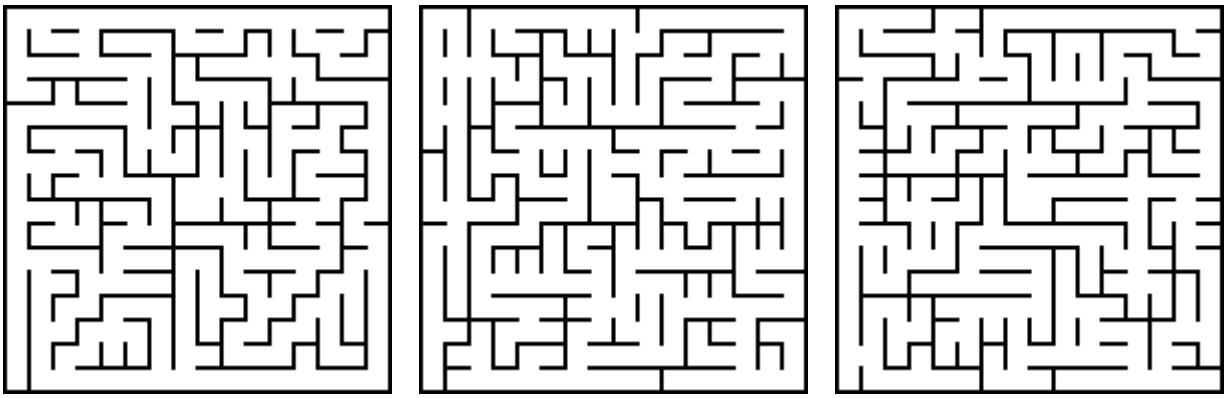
(1) 初期迷路を用意する。
(2) 指定のサンプリング数の経路を取得するまで、(3)～(5)の処理を繰り返す。
(3) 公式迷路群から、迷路をランダムに 1 つ選択する。
(4) 選択した迷路から、2 つの異なる区画をランダムに選択する。
(5) 選択した迷路から、2 つの区画を結ぶ最短経路をランダムに 1 つ取得する。
経路がなければ、(3)へ戻る。
(6) サンプリングした経路ひとつひとつについて、最短経路保存法を適用する。
(7) 公式迷路群から、迷路をランダムに 1 つ選択する。
(8) 選択した迷路から、迷路最短経路をランダムに 1 つ取得する。
(9) 取得した迷路最短経路について、最短経路保存法を適用する。
(10) 取得した迷路最短経路を塞がないように、柱有効化法を適用する。

実際の生成は、次のように行った。

- 公式迷路群： 3.1.2 節の 253 種類
- サンプリング経路数： 100
- 生成数： 1,000 個

ここで生成した迷路群は、「ランダム迷路群 2」と呼ぶことにする。

なお、最短走行用の経路が、移動区画数評価の迷路最短経路と同一とは限らないが、ここでは、最短走行用の経路を意図的に配置することはせず、意図的に配置した移動区画数評価の迷路最短経路の迂回路として得られることを期待した。



(a) 1 番目の生成迷路　　(b) 500 番目の生成迷路　　(c) 1000 番目の生成迷路

図 3.12　ランダム迷路群 2 の迷路

3.2.4.2　生成結果と評価結果

壁配置が全く同じ迷路は、一組も生成されなかった。生成した迷路の例を図 3.12 に示す。また、3.1.2 節での公式迷路の評価と同様に、ランダム迷路群 2 を局所経路パターンによって評価した。結果を、3.2.2.3 節と同様の形式で、図 3.13～図 3.16、表 3.3 に示す。

3.2.4.3　考察

ランダム迷路群 2 における直進パターン、旋回パターンそれぞれの出現回数の分布は、いずれも公式迷路群の最小値と最大値に収まっていた（表 3.1、表 3.3）。直進パターンと旋回パターンの出現回数の分布（図 3.15）では、相関は見られなかった。図 3.12 に示した見た目からも分かるとおり、公式迷路に似た迷路を生成できた。

しかし、表 3.3 の平均値を比較すると、ランダム迷路群 2 は公式迷路群（表 3.1）に対して、直進パターンの出現回数はほぼ同値であったが、旋回パターンの出現回数は大きかった。これは、最短経路保存法を繰り返し用いたことが原因と考えられる。最短経路保存法で保たれるのは任意の 1 つの経路だけであり、何度も用いると古い経路の上に新しい経路が上書きされる。新旧の経路が交差することによって十字路や三叉路ができ、旋回パターンが多く現れたと見られる。また、最短経路保存法と柱有効化法は、それぞれの壁配置方法の都合上、閉鎖領域を生成しない。これは、図 3.12 に示した迷路例からもある程度に確認できる。

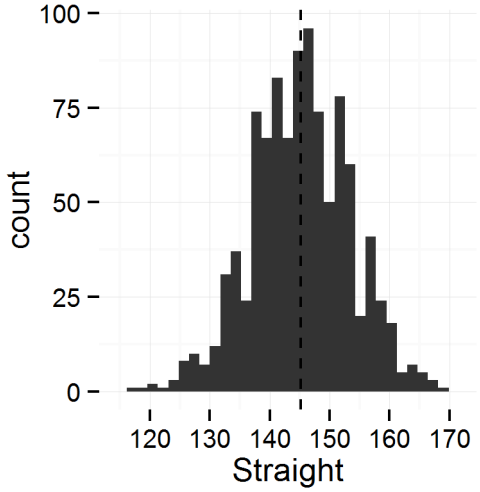


図 3.13　ランダム迷路群 2 の直進パターン

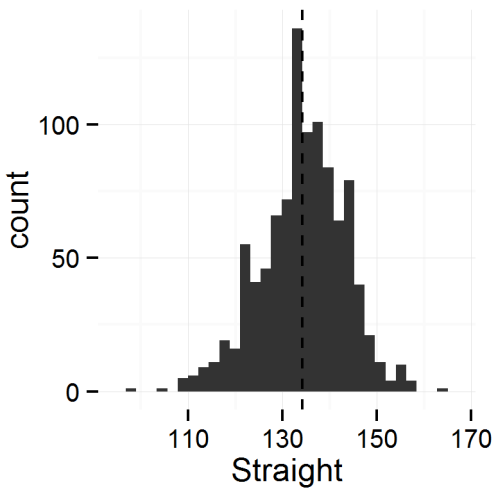


図 3.14　ランダム迷路群 2 の旋回パターン

表 3.3　ランダム迷路群 2 の局所経路パターン評価の統計値

	最小値	最大値	平均値	標準偏差
直進パターン	117	169	145.2	8.2
旋回パターン	167	265	222.1	11.9

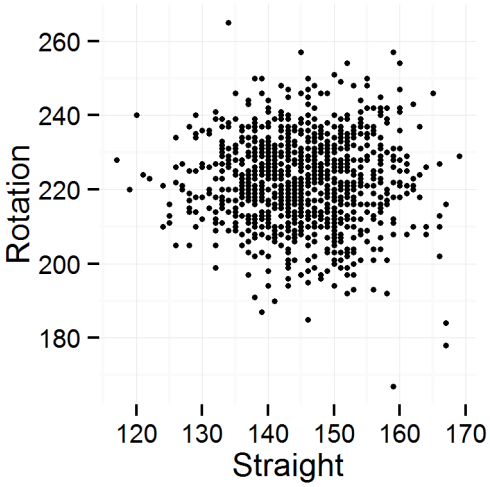


図 3.15　ランダム迷路群 2 の直進と旋回の分布

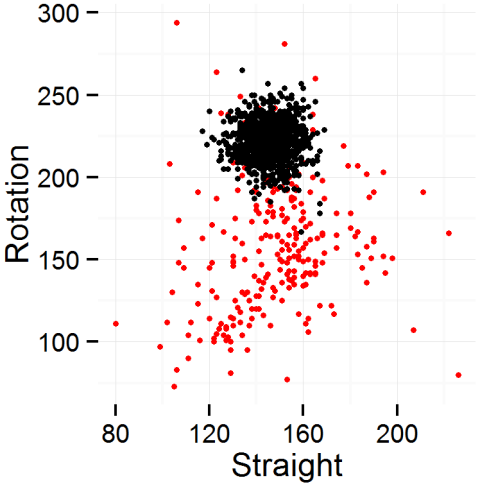


図 3.16　ランダム迷路群 2（黒）と公式迷路群（赤）の分布

3.2.5　迷路生成方法の改良

3.2.4 節で示した方法は、平均的な公式迷路と比べて旋回パターンが多い迷路が生成される傾向にあったが、本節ではこれを改良する。

3.2.5.1　生成方法

3.2.4 節の方法で旋回経路が多く現れる原因は、最短経路保存法で配置する経路同士が交差することが原因と考えられる。そこで、最短経路保存法で任意の経路を配置した直後に、その経路沿いにある確率で壁を配置するようにする。これによって、旋回パターンの出現を抑える。この改良を加えた迷路生成の手順は、次のとおりである。

(1) 初期迷路を用意する。
(2) 指定のサンプリング数の経路を取得するまで、(3)～(5)の処理を繰り返す。
(3) 公式迷路群から、迷路をランダムに 1 つ選択する。
(4) 選択した迷路から、2 つの異なる区画を選択する。
(5) 選択した迷路から、2 つの区画を結ぶ最短経路をランダムに 1 つ取得する。
　経路がなければ、(3)へ戻る。
(6) サンプリングした経路ひとつひとつについて、(7)～(9)の処理を行う。
(7) サンプリング経路で以て、最短経路保存法を適用する。
(8) サンプリング経路沿いで、壁が設置されていない壁設置可能位置をすべて取得する。
(9) 取得した壁位置ひとつひとつについて、任意の確率で壁を設置する。
(10) 公式迷路群から、迷路をランダムに 1 つ選択する。
(11) 選択した迷路から、迷路最短経路をランダムに 1 つ取得する。
(12) 取得した迷路最短経路について、最短経路保存法を適用する。
(13) 取得した迷路最短経路を塞がないように、柱有効化法を適用する。

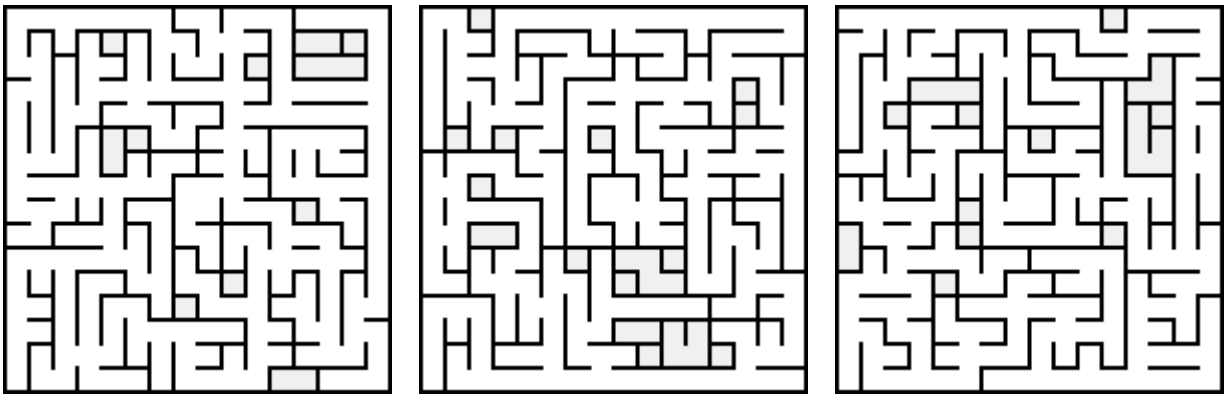
実際の生成は、次のように行った。

- 公式迷路群： 3.1.2 節の 253 種類
- サンプリング経路数： 100
- 経路沿いの壁を設置する確率： 25 %
- 生成数： 1,000 個

ここで生成した迷路群は、「ランダム迷路群 3」と呼ぶことにする。

3.2.5.2　生成結果と評価結果

壁配置が全く同じ迷路は、一組も生成されなかった。生成した迷路の例を図 3.17 に示す。また、3.1.2 節での公式迷路の評価と同様に、ランダム迷路群 3 を局所経路パターンによって評価した。結果を、3.2.2.3 節と同様の形式で、図 3.18～図 3.21、表 3.4 に示す。



(a) 1 番目の生成迷路　　(b) 500 番目の生成迷路　　(c) 1000 番目の生成迷路

図 3.17　ランダム迷路群 3 の迷路

3.2.5.3　考察

ランダム迷路群 3 における直進パターン、旋回パターンそれぞれの出現回数の分布は、いずれも公式迷路群の最小値と最大値に収まっていた（表 3.1、表 3.4）。直進パターンと旋回パターンの出現回数の分布（図 3.20）では、相関は見られなかった。表 3.4 の平均値を見ると、ランダム迷路群 2 と比べて、直進パターン、旋回パターンともに出現回数が減少しており、経路沿いに壁を配置した効果が表れている。公式迷路群（表 3.1）と比較すると、ランダム迷路群 2 より直進パターンの出現回数の差は開いてしまったが、旋回パターンの出現回数は公式迷路群に近付けることができた。図 3.21 より、全体的にも公式迷路群の分布の平均値に近付いている。また、図 3.17 を見ると、迷路中に閉鎖領域が現れていることが分かる。これも、経路沿いに壁を配置するようにしたことによるものである。以上より、公式迷路に似た迷路を生成することができた。

次章以降の探索アルゴリズムの検証では、ここで生成した迷路に用いることにする。ランダム迷路群 3 の分布は、公式迷路群に比べて偏りが見られ、平均ともほとんど等しいと判断できるものはない。現状、この改善策はない。しかし、公式迷路群の分布の範囲内に収まっており、類似はあると判断できることから、ランダム迷路群 3 を採用する。

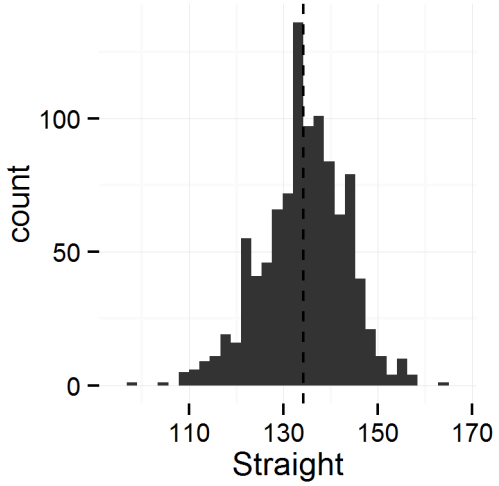


図 3.18　ランダム迷路群 3 の直進パターン

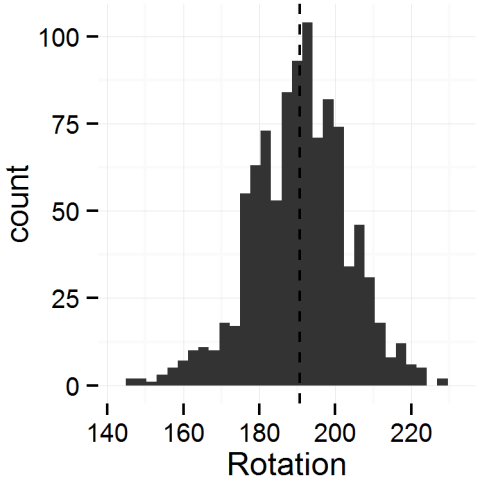


図 3.19　ランダム迷路群 3 の旋回パターン

表 3.4　ランダム迷路群 3 の局所経路パターン評価の統計値

	最小値	最大値	平均値	標準偏差
直進パターン	97	163	134.2	8.9
旋回パターン	146	228	190.6	12.7

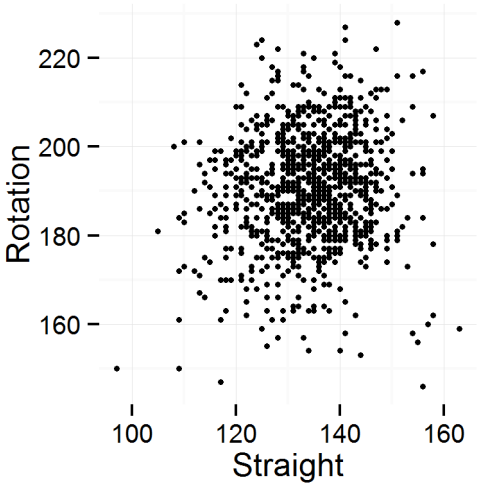


図 3.20　ランダム迷路群 3 の直進と旋回の分布

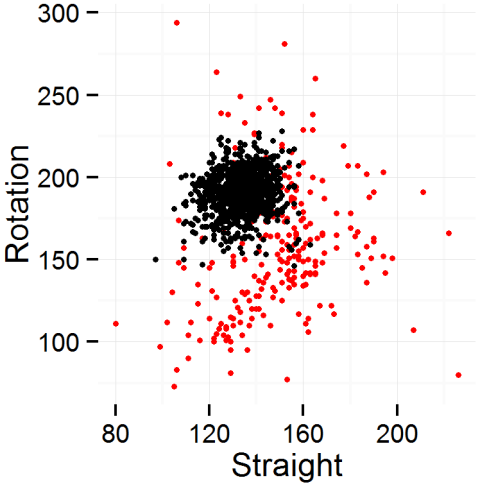


図 3.21　ランダム迷路群 3（黒）と公式迷路群（赤）の分布

第4章　最短経路探索アルゴリズム

本章では、効率的な最短経路探索を実現するための提案および検証を行う。まず既存の探索アルゴリズムについて検証を行い、どのようなことが探索の効率化に有効であるか確認する。その後、効率的な最短経路探索を実現するために、最短経路探索アルゴリズムを構成する要素アルゴリズムを 2 つ提案する。最後に、その 2 つの要素アルゴリズムを組み合わせた際に想定される問題について検証を行う。

4.1　既存の探索アルゴリズムの検証

最短経路探索について考える前に、目的地探索に利用される既存のいくつかの探索アルゴリズムの検証を行う。これによって、実際の競技会で多用される足立法がなぜ効率が良いと言われるのか検証し、その考え方を最短経路探索に取り入れるようにする。

4.1.1　方法

いくつかの目的地探索アルゴリズムについて、シミュレーションによって複数の迷路を探索したとき、スタートからゴールに到達するまでに要した移動区画数を調べた。検証を行った探索アルゴリズムは、次の 5 種類である。

- トレモー法
- 拡張左手法
- 拡張右手法
- 求心法
- 足立法

各探索アルゴリズムについては、2.2.4 節を参照してほしい。なお、トレモー法、求心法、足立法は、進行方向を一意に決定できないことがあるが、その場合は、候補の中からランダムに一方向を選択するようにした。

検証には、次の 2 種類の迷路群を用いた。

- 公式迷路群（253 種類）
- ランダム迷路群（3.2.5 節のランダム迷路群 3 と同様に生成した 10,000 種類）

その他、シミュレーションにおける細かな設定は、次のとおりである。基本的には、競技規定に準じる。

- マウスの位置は、区画単位で扱う。
- マウスは、隣接する 4 区画のみへ移動する。
- マウスの動作の評価関数には、移動区画数を用いる。
- ロボットは、進入した区画の前、左、右の壁の有無状態を検出できる。

この設定は、以降の探索シミュレーションでも共通に用いる。

4.1.2 結果

公式迷路群におけるシミュレーション結果を表 4.1、ランダム迷路群における結果を表 4.2 に示す。いずれの場合においても、ゴールに到達することができた。

表 4.1　公式迷路群における各探索アルゴリズムのゴールまでの移動区画数

	トレモー法	拡張左手法	拡張右手法	求心法	足立法
平均値	**221.5**	**277.9**	**278.2**	**150.9**	**97.3**
最大値	429	428	440	456	237
中央値	229.0	284.0	282.0	119.0	91.0
最小値	42	76	66	28	28
標準偏差	103.1	69.0	69.1	89.5	34.0

表 4.2　ランダム迷路群における各探索アルゴリズムのゴールまでの移動区画数

	トレモー法	拡張左手法	拡張右手法	求心法	足立法
平均値	**227.9**	**214.6**	**215.6**	**170.9**	**136.0**
最大値	458	438	418	434	335
中央値	226.0	207.0	208.0	153.0	130.0
最小値	30	46	38	24	24
標準偏差	78.2	73.2	72.5	74.5	45.9

4.1.3 考察

表 4.1、表 4.2 において平均値を見ると、迷路群の違いによらず足立法が最も良く、それに求心法が続く結果となった。トレモー法と拡張壁伝い法（拡張左手法と拡張右手法）の優位性は、迷路群によって異なり、公式迷路群ではトレモー法のほうが良く、ランダム迷路群では拡張壁伝い法のほうが良い結果となった。

2 種類の迷路群においてトレモー法の結果に差異があまりないことを考慮すると、それに対して差が大きい拡張壁伝い法は、迷路の種類による影響を大きく受けやすいものと見られる。ここでのトレモー法と拡張壁伝い法の違いは、進行方向の選択方法のみである。トレモー法はランダムであるのに対し、拡張壁伝い法は壁の配置に依存して進行方向を決定する。それによって、拡張壁伝い法は迷路外周沿いから徐々に中心へと近づいていくという挙動を示しやすい。このような癖が、いずれかの迷路群に多く存在する何かしらの特徴に合致してしまい、迷路群による差が大きくなったと考えられる。

拡張左手法と拡張右手法を比較すると、両者の差はほとんどなかった。わずかに拡張左手法のほうが良い結果であった。拡張左手法は外周を時計回りに辿るように進むのに対し、拡張右手法は外周を反時計回りに辿るように進む。競技規定によって、迷路の角に位置するスタートは、外周を時計回りで出発するように壁が配置されることになっている。そのため、拡張右手法では反時計回りの軌道に乗るのに、拡張左手法よりわずかに移動区画数が多くなる。このことが、微妙な差として表れたと考えられる。迷路形状の偏りを考慮せず競技規定のみで考えるのであれば、

マイクロマウス競技においては、拡張右手法より拡張左手法のほうがわずかに有利であると確認できた。

続いて、トレモー法と求心法を比較する。探索結果は、求心法のほうが良かった。両者の差異は、進行方向の選択方法にある。求心法は、現在地から見たゴールの方角を考慮して進行方向を決定する。それに対してトレモー法は、ゴールに関する情報を一切考慮せずランダムに決定する。このことから目的地探索においては、ランダムであるより目的地に関する情報を考慮したほうが有利であるということが確認できた。

最後に、求心法と足立法を比較する。足立法はトレモー法的な方法を取らないが、ゴールに関する情報を考慮するという点は求心法と共通である。足立法は、既知の壁の有無を考慮した、現在地からゴールまでの移動区画数を利用する。既知の壁がなければ、足立法と求心法の進行方向の選択方法は等価である。探索結果は足立法のほうが良かったことから、既知の探索情報を利用すると、探索を効率化できることが確認できた。

また、ここで足立法が優位であるのは、他の探索アルゴリズムがトレモー法的な方法を取ることにもよる。これには二つのことが考えられる。まず一つ目は、トレモー法的な方法は、バックトラックを行うということがある。バックトラックは、一度通った経路をもう一度通るという挙動を取る。マイクロマウス迷路のように、ある位置から別の位置までの移動経路が複数あるのなら、迂回したほうが良い場合がある。二つ目は、進行方向の選択方法の違いにある。トレモー法的な方法は未探索の方向を優先するが、足立法は基本的には未探索であるかを考慮しない。これによって、例えばマウスが未探索の一本道を進んでいるとすると、トレモー法などは道なりに進み続けるが、足立法はゴールから遠ざかるようなら引き返すということができる。つまり、未知の情報を得ようとするより、目的地に到達することを優先したほうが、効率が良いと考えられる。この点からも、足立法は他の探索アルゴリズムより効率的であると考えられる。

以上のように、足立法の効率の良さを確認することができた。足立法は、他のアルゴリズムより標準偏差も小さく表れており、安定して探索を行えていることも分かる。公式迷路は足立法に有利な設計になっていることを伺わせる結果もあったが、ランダム迷路群の結果も鑑みれば、その配慮によらず足立法の優位性は明確である。

4.1.4　まとめ

ここでの検証によって得られた、探索の効率化に関わる事柄は、次のとおりである。

- ランダム性が強い探索アルゴリズムに対して、ある偏った挙動を見せる探索アルゴリズムは、迷路に存在する偏った特徴の影響を受けやすい。（トレモー法と拡張壁伝い法の比較）
- マイクロマウス迷路においては、拡張右手法より拡張左手法のほうが、効率が良い。
- ランダムであるより、目的地に関する情報を考慮したほうが、効率が良い。（トレモー法と求心法の比較）
- 探索中に得た情報を利用したほうが、効率が良い。（求心法と足立法の比較）
- 未知の情報を得ようとするより、目的地に到達することを優先したほうが、効率が良い。
- 足立法は、検証に用いた他の探索アルゴリズムのいずれより、効率が良く、安定性が高い。

4.2　効率的な最短経路探索を行うために

迷路最短経路を見つけるために、実際の競技においては、最短経路探索より全面探索が行われることが多い。全面探索は、迷路最短経路を見つけようとするのではなく、迷路形状の全体像を把握しようとするものである。全面探索として、どのような探索アルゴリズムが用いられているかは、定かではないが、基本的には、未探索の区画を記憶するようにしておき、順々に訪問していくというものであると考えられる。区画を巡回する方法（以下、巡回アルゴリズム）としては、マウスの現在地から最も近い区画を順に辿るというもの（以下、近傍巡回）がよく用いられているようである。全面探索を行ったあと、A*アルゴリズムなどを使ってオフラインに迷路最短経路を求める。

全面探索では、訪問する必要がない区画も含めて探索を行っていると推測できる。最短経路探索としてオンラインに迷路最短経路の探索を行う場合、このような区画を除外（以下、枝刈り）できれば、探索を効率化できる。また、近傍巡回は直近の区画のみを評価するため、迷路によっては先に探索しておいたほうが、探索行動全体としては効率が良くなるような区画を見過ごしていることが考えられる。探索の終盤にそのような区画が残っていれば、マウスはその区画を調べるためだけに、既探索区画を再び通ることになる。見過ごしを抑えられるように、先の見通しを立てて巡回を行うことができれば、探索の効率化を期待できる。

4.3　不要探索区画の枝刈り

4.1節の検証によって､探索中に得た情報を利用すると探索を効率化できることが確認できた。これを参考に、既知の探索情報を利用して、迷路最短経路とならない区画を探索から除外（枝刈り）し、探索を効率化する要素アルゴリズムを提案する。

4.3.1　不要探索区画

最短経路探索は、迷路最短経路を見つけることが目的である。マイクロマウス迷路には、迷路最短経路が複数存在することが多い。しかし、迷路中のすべての区画が、迷路最短経路であるとは考えづらい。迷路最短経路とならない区画は、探索する必要がない。このような区画を不要探索区画と呼ぶことにする。

不要探索区画の判定方法について述べる。まず、迷路中のある区画を経由するスタート-ゴール間の経路を考える。そのコスト（以下、経由コスト）が迷路最短経路のコストより大きいとき、その区画は迷路最短経路になり得ないため、不要探索区画と判定できる。

4.3.2　暫定迷路最短経路による枝刈り

不要探索区画を枝刈りするためには迷路最短経路が必要である。しかし、迷路最短経路の探索中に、未知の迷路最短経路を利用することは不可能である。そこで、迷路最短経路でなく、探索中に既知となったスタート-ゴール間の最短経路（以下、暫定迷路最短経路）を用いることを考える。暫定迷路最短経路のコストが、迷路最短経路のコストより小さくなることはあり得ないため、不要探索区画の枝刈りに破綻が生じることはない。また、この方法では、暫定迷路最短経路が 1 つ以上既知となっている必要がある。2.1.5 節で述べたとおり、ここでの最短経路探索は、一度ゴールに到達してから実行されることを想定しているため、この前提を満たすことができる。

探索中に行う、暫定迷路最短経路を用いた枝刈りは、次のとおりである。

(1) 暫定迷路最短経路のコストを計算する。
(2) ある区画を経由する、スタート-ゴール間の経路のコストを計算する。
(3) (2)のコストが(1)のコストより大きいとき、(2)の経由区画は不要探索区画である。

なお、(2)の経路は、「未知」壁を「なし」として扱い導出されるものとする。

4.3.3　検証

不要探索区画の枝刈りを行った場合の最短経路探索について、シミュレーションによる検証を行った。

4.3.3.1　方法

基本的に、4.1.1 節で示した要領でシミュレーションを行った。シナリオとしては、マウスはスタート後に足立法（4.1 節で用いたものと同様）によってゴールまで探索し、その後、各種探索アルゴリズムによって、最短経路探索を行うものとした。検証に用いた探索アルゴリズムは、

次のとおりである。

・ A1： 近傍巡回（従来の全面探索）
・ A2： 近傍巡回＋枝刈り

なお、A1、A2 いずれにおいても、探索が必要な区画をすべて訪問すると、現在地からスタート区画までの最短経路を通りスタートまで戻ってくるようにした。

4.3.3.2 結果

シミュレーション結果を、表 4.3、表 4.4 に示す。評価値である「移動区画数」と、ロボットに訪問された区画の数を表す「進入区画数」について示した。なお、ここでの値は、マウスがスタートをしてから、再びスタートに戻ってくるまでのものであり、最短経路探索に要したコストだけでなく、目的地探索に要したコストも含んでいる。マウスが迷路最短経路を見つけるまでの一連の探索行動という形で評価を行うようにした。

表 4.3　A1（近傍巡回）の探索結果

	公式迷路群		ランダム迷路群	
	移動区画数	進入区画数	移動区画数	進入区画数
平均値	370.4	219.0	455.3	222.7
最大値	514	252	634	242
中央値	374.0	226.0	448.0	224.0
最小値	242	140	228	122
標準偏差	50.2	23.5	41.0	8.0

表 4.4　A2（近傍巡回＋枝刈り）の探索結果

	公式迷路群		ランダム迷路群	
	移動区画数	進入区画数	移動区画数	進入区画数
平均値	290.0	189.6	310.2	170.0
最大値	420	252	484	233
中央値	296.0	192.0	314.0	175.0
最小値	92	66	52	27
標準偏差	57.7	34.1	58.0	30.1

4.3.3.3 考察

表 4.3、表 4.4 より、迷路群の違いによらず、枝刈りによって進入区画数が減少していることから、不要探索区画の枝刈りが機能していることが確認できる。移動区画数も減少しており、不要探索区画の枝刈りによって、探索を効率化できることが確認できた。

移動区画数の値のばらつきを見ると、枝刈りによって最小値や最大値は減少しているが、標準偏差はわずかに増加していることが分かる。これは、枝刈り自体は多くの迷路で起こるが、どの程度の数を枝刈りできるかは迷路によって異なるということを表している。このことは、進入区画数の標準偏差の違いからも確認できる。

4.4　巡回路の最適化

前節のような枝刈りによって探索が必要な区画（以下、要探索区画）が分かれば、それらを順に訪問することが必要となる。ここでは、その巡回路を大域的な情報を用いて導出し、見過ごしを抑えることによって、探索を効率化する要素アルゴリズムを提案する。

4.4.1　要探索区画の最適巡回路

要探索区画は、壁の有無が分からない未知領域に含まれるため、巡回路を考えるためにはそのことを考慮する必要がある。ここでは、足立法で用いられる考え方を取り入れる。足立法は、未知の領域には壁が存在しない理想的な状態を仮定し、状態が更新されるたびに最短経路を求め、次動作を決定する。最短経路探索においても同様に、未知領域に壁は存在しないとして扱い、逐次的に要探索区画の最適巡回路を導出するようにする。これによって探索の効率化を図る。

4.4.2　準最適巡回

近傍巡回は、直近の区画を評価するだけで済むため巡回路の導出を行っていないが、あえて巡回路を考えるとすると、それは貪欲法（Greedy Algorithm）によって得られるものと同等と見なせる。つまり、マウスの現在地を始点に、直近の区画へ移動するということを繰り返したときに得られる巡回路と等しい。貪欲法は、巡回セールスマン問題（Traveling Salesman Problem: TSP）においてしばしば用いられる。しかし、得られる巡回路は最適解とは限らない。したがって、近傍巡回には改善の余地があるという見方もできる。

最適巡回路を用いれば、探索の効率化を最も期待できるが、TSP の最適解を得ることは現実的でない。そこで、代わりに準最適解を求めるようにする。ここでは、準最適な巡回路を「準最適巡回路」と呼ぶことにし、これを利用した巡回アルゴリズムを「準最適巡回」と呼ぶことにする。

準最適解を得るための最適化手法としては、逐次改善法に類するものを用いることが望まれる。実際の競技での利用を想定すると、準最適巡回路の導出が動作計画の所要時間内に完了しないことが考えられる。逐次改善法であれば、処理を途中で打ち切ったとしても、解の改善が見込める。

4.4.3　検証

巡回路の最適化を行った場合の最短経路探索について、シミュレーションによる検証を行った。

4.4.3.1　方法

基本的には、4.3.3.1 節と同じ要領でシミュレーションを行った。検証に用いた探索アルゴリズムは、次のとおりである。

- A1：近傍巡回
- A3：準最適巡回

A1 は、4.3.3 節と同じものである。また、A3 で行っている最適化を伴う巡回路の導出手順は、次のとおりである。

(1) 迷路中の未知区画（四方の壁のいずれかが「未知」である区画）をすべて取得する。
(2) 貪欲法を用いて、マウスの現在地を始点、スタート区画を終点とするように、取得した未知区画群を巡回する巡回路を導出する。
(3) or-opt 法を用いて、巡回路を最適化し、準最適巡回路を導出する。

ここで巡回路の最適化手法として用いている or-opt 法は、巡回路の一部を切り取り、別の箇所に仮挿入したときに巡回路のコストが小さくなるようなら、その移し替えを実行するというものである。巡回路のコストが大きくなることはないため、逐次改善法に分類される。or-opt 法としては、移し替えるノード（区画）に応じてさまざまパターンを考えることができるが、今回は図 4.1 に示した 2 つを用いた。

巡回路の導出を内包した、A1、A3 に共通の動作計画の手順は、次のとおりである。なお、動作計画は、マウスが未知区画に到達して保持する迷路の壁情報が更新されたときに実行する。

(1) 巡回路を導出する。
(2) 巡回路の 2 つ目の区画までの理想最短経路を導出する。(巡回路の 1 つ目は現在地である)
(3) 導出した理想最短経路の方向に進行する。

なお、理想最短経路とは、「未知」の壁は「なし」と仮定した迷路上における、任意 2 地点間の最短経路を表すとする。

(a) 1 つのノードを移し替える　　(b) 連続する 2 つのノードを移し替える

図 4.1　or-opt 法（巡回路がより短くなるようならノードの順番を入れ替える）

4.4.3.2　結果

新たにシミュレーションを行った A3 の結果を表 4.5 に示す。形式は、4.3.3.2 節のときと同様である。

表 4.5　A3（準最適巡回）の探索結果

	公式迷路群		ランダム迷路群	
	移動区画数	進入区画数	移動区画数	進入区画数
平均値	369.1	219.2	452.6	222.9
最大値	502	252	634	242
中央値	372.0	227.0	446.0	224.0
最小値	250	140	220	122
標準偏差	49.8	23.6	39.4	8.0

4.4.3.3　考察

表 4.3（近傍巡回）、表 4.5（準最適巡回）より、迷路群の違いによらず、巡回路の最適化による移動区画数はわずかに減少した。微々たるものではあるが、公式迷路群だけでなく、10,000 種類のランダム迷路群においても同様の結果になっていることから、ここではわずかながら効果があったと見ることにする。しかし、効果が小さかったことは確かである。巡回路が長ければコストが大きく付くのは当然のように思える。今回の結果から察するに、or-opt 法が巡回路の最適化にあまり効果的でなかったと考えられる。しかし、これは貪欲法が導出している初期巡回路に、改善の余地がほとんどないと見ることもできる。ここでは、コストが小さい巡回路ほど探索を効率化できるということを示せていないが、貪欲法による巡回路を用いるような近傍巡回でも、ある程度効率的に探索を行えていると考えられる。

4.4.4　枝刈りとの併用

不要探索区画の枝刈りと巡回路の最適化を同時に行う場合について検証する。

4.4.4.1　方法

基本的には、4.3.3.1 節と同じ要領でシミュレーションを行った。検証に用いた探索アルゴリズムは、次のとおりである。

- A1：近傍巡回
- A2：近傍巡回＋枝刈り
- A3：準最適巡回
- A4：準最適巡回＋枝刈り

A1 と A2 は 4.3.3 節、A3 は 4.4.3 節と同じものである。なお、A4 における巡回路の導出方法は、A3 と同様である。

表 4.6　A4（準最適巡回＋枝刈り）の探索結果

	公式迷路群		ランダム迷路群	
	移動区画数	進入区画数	移動区画数	進入区画数
平均値	292.0	189.6	314.3	170.7
最大値	430	252	512	236
中央値	294.0	193.0	318.0	175.0
最小値	92	66	52	27
標準偏差	58.3	33.5	59.3	30.3

表 4.7　公式迷路群における A1～A4 の探索結果のまとめ

	A1 近傍巡回	A2 近傍巡回 ＋枝刈り	A3 準最適巡回	A4 準最適巡回 ＋枝刈り
平均値	370.4	290.0	369.1	292.0
最大値	514	420	502	430
中央値	374.0	296.0	372.0	294.0
最小値	242	92	250	92
標準偏差	50.2	57.7	49.8	58.3

表 4.8　ランダム迷路群における A1～A4 の探索結果のまとめ

	A1 近傍巡回	A2 近傍巡回 ＋枝刈り	A3 準最適巡回	A4 準最適巡回 ＋枝刈り
平均値	455.3	310.2	452.6	314.3
最大値	634	484	634	512
中央値	448.0	314.0	446.0	318.0
最小値	228	52	220	52
標準偏差	41.0	58.0	39.4	59.3

4.4.4.2　結果

新たにシミュレーションを行った A4 の結果を表 4.6 に示す。形式は、4.3.3.2 節のときと同様である。また、A1～A4 の探索家結果を移動区画数についてまとめ直したものを、迷路群別に表 4.7、表 4.8 に示しておく。

4.4.4.3　考察

表 4.7、表 4.8 の A3、A4 より、迷路群の違いによらず、巡回路の最適化を行う場合であっても、枝刈りは探索を効率化できることが分かる。一方で、A2、A4 を見ると値はわずかに増加している。このことから、枝刈りと巡回路の最適化は、それぞれ独立的に作用するものではないことが分かる。枝刈りと巡回路の最適化の併用に関する検証は、4.5 節でも行う。

4.5　要探索区画の見過ごし

巡回路の最適化は、要探索区画の見過ごしを抑えることが目的であった。しかし、枝刈りを行う場合、見過ごした区画が探索中に除外されることがある。つまり、見過ごしをなくし、既知領域を通るような無駄な行動を少なくすることと、あえて見過ごして、枝刈りによって訪問する必要がなくなることは、二律背反の関係にあると考えられる。このとき、探索が最も良くなるような、適度な「見過ごし」があると推測できる。そこで、枝刈りを行うとき、要探索区画をどの程度に見過ごすべきであるか検証する。

4.5.1　方法

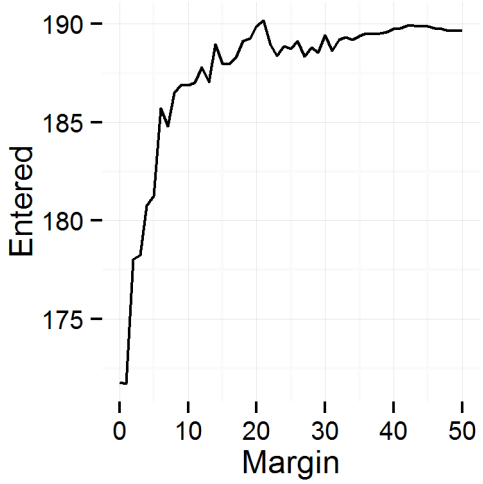
要探索区画を見過ごす方法としては、未知領域には壁が存在しないとしたスタート-ゴール間の最短経路（以下、理想迷路最短経路）を基準に、そのコストからいくらかのマージンを設け、それよりも経由コストが大きい区画を訪問しないようにする。この方法を「見過ごし枝刈り」と呼ぶことにする。見過ごし枝刈りでは、「暫定迷路最短経路のコスト」と「理想迷路最短経路のコスト＋マージン」の小さいほうを上限とし、それより大きな経由コストの区画を枝刈りする。なお、見過ごし枝刈りによって除外した区画は、理想迷路最短経路の更新によって巡回しなくてはならなくなることがある。見過ごし枝刈りは通常の枝刈りも含むため、暫定迷路最短経路によって探索から除外された場合は再度巡回路に含められることはないが、理想迷路最短経路（＋マージン）に由来する除外は一時的なものであることに注意が必要である。

検証においては、理想迷路最短経路からのマージンを変化させて（0～50）、どの程度離れた要探索区画を見過ごすべきかを調べた。基本的には、4.3.3.1 節と同じ要領でシミュレーションを行った。検証に用いた探索アルゴリズムは、次のとおりである。

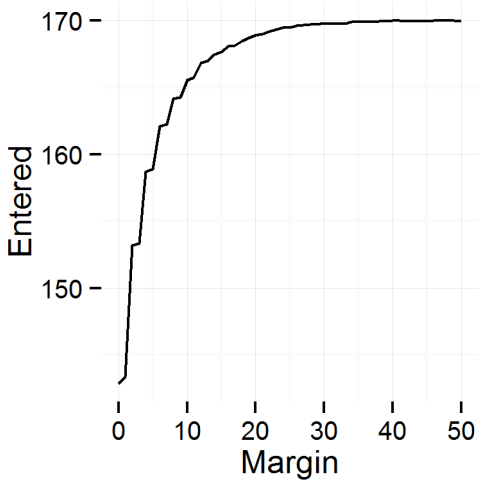
- A5：　近傍巡回＋見過ごし枝刈り
- A6：　準最適巡回＋見過ごし枝刈り

4.5.2　結果

A5 と A6 について、理想迷路最短経路からのマージンと進入区画数の平均値の関係を図 4.2、図 4.3 に、マージンと移動区画数の平均値の関係を図 4.4、図 4.5 に、マージンと移動区画数の標準偏差の関係を図 4.6、図 4.7 に示す。また、マージンを 0 としたとき、つまり、経由コストが理想迷路最短経路のコストと等しい区画のみを巡回するようにしたときの結果を表 4.9、表 4.10 に示す。

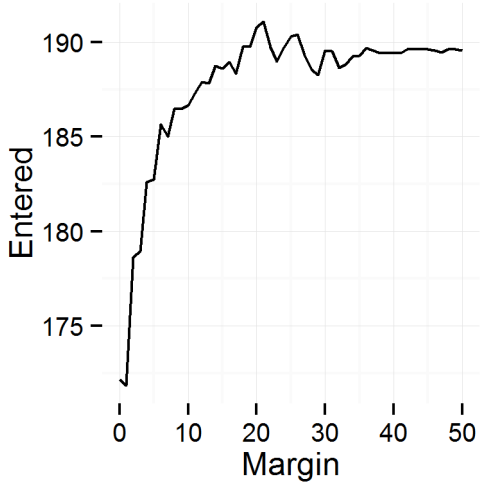


(a) 公式迷路群

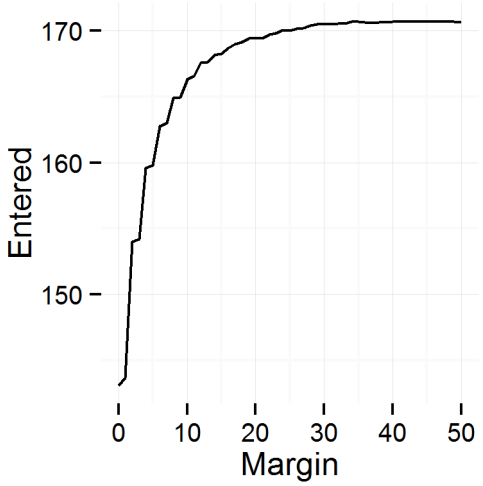


(b) ランダム迷路群

図 4.2　A5（近傍巡回＋見過ごし枝刈り）におけるマージンと進入区画数の平均値の関係

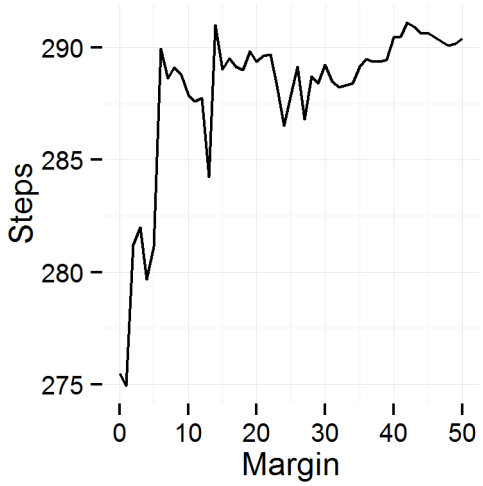


(a) 公式迷路群

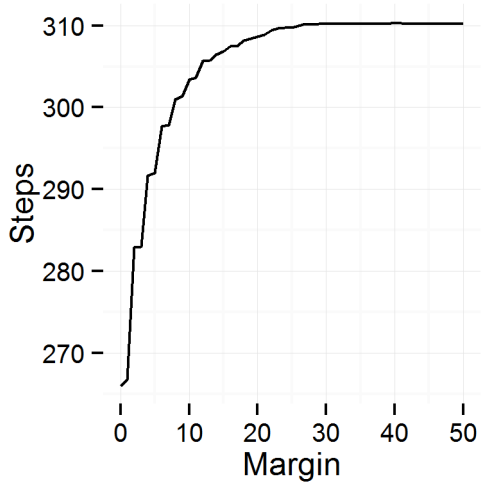


(b) ランダム迷路群

図 4.3　A6（準最適巡回＋見過ごし枝刈り）におけるマージンと進入区画数の平均値の関係

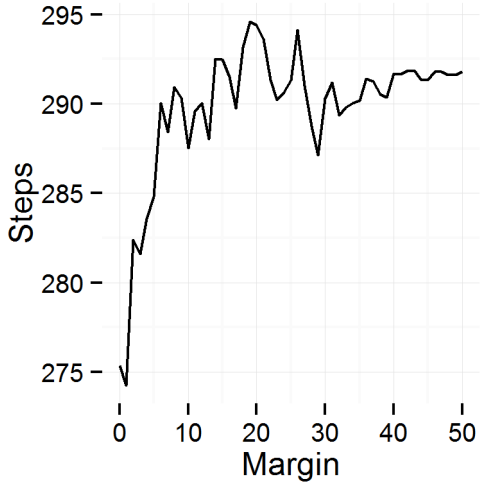


(a) 公式迷路群

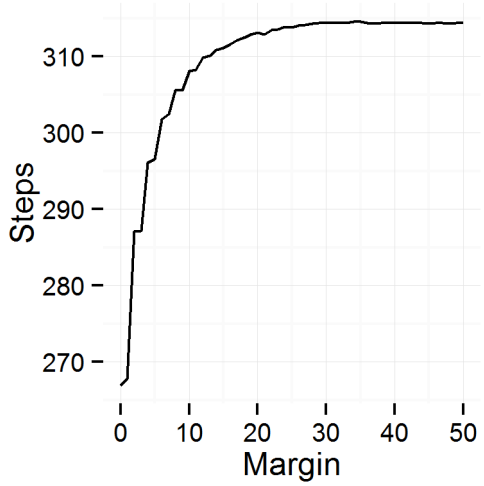


(b) ランダム迷路群

図 4.4　A5（近傍巡回＋見過ごし枝刈り）におけるマージンと移動区画数の平均値の関係

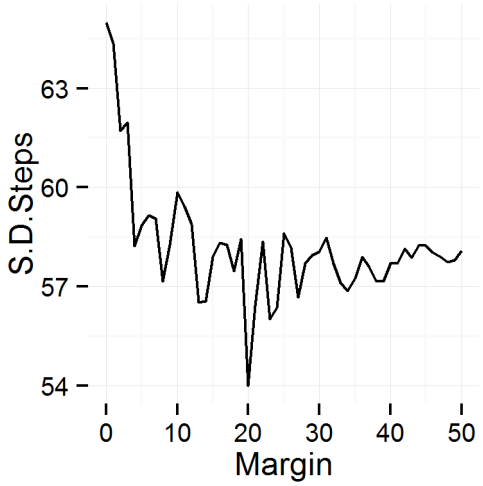


(a) 公式迷路群

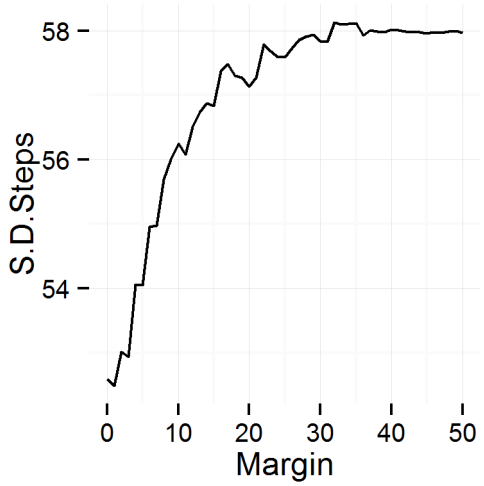


(b) ランダム迷路群

図 4.5　A6（準最適巡回＋見過ごし枝刈り）におけるマージンと移動区画数の平均値の関係

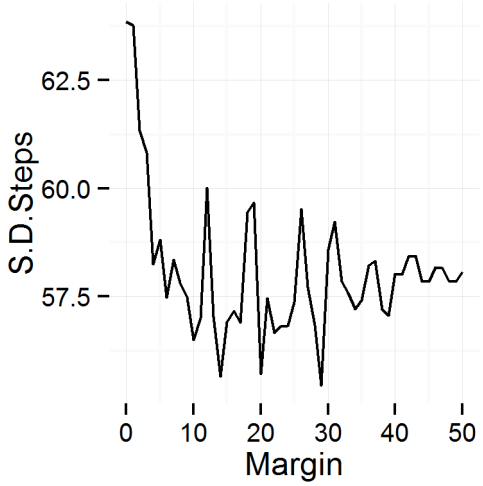


(a) 公式迷路群

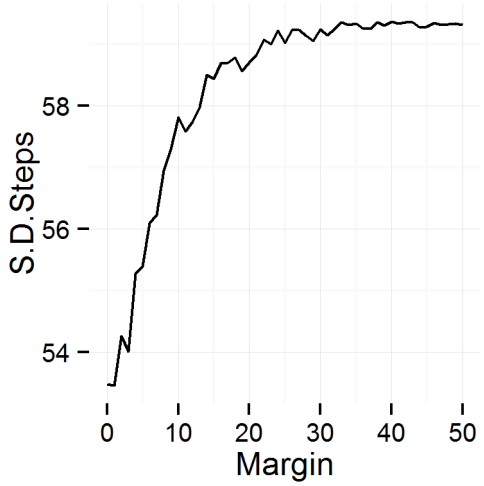


(b) ランダム迷路群

図 4.6　A5（近傍巡回＋見過ごし枝刈り）におけるマージンと移動区画数の標準偏差の関係



(a) 公式迷路群



(b) ランダム迷路群

図 4.7　A6(準最適巡回＋見過ごし枝刈り)におけるマージンと移動区画数の標準偏差の関係

表 4.9　A5（近傍巡回＋見過ごし枝刈り）の探索結果（マージン=0）

	公式迷路群		ランダム迷路群	
	移動区画数	進入区画数	移動区画数	進入区画数
平均値	**275.5**	171.8	**266.0**	142.3
最大値	484	248	448	214
中央値	272.0	174.0	266.0	145.0
最小値	52	43	52	27
標準偏差	65.0	33.3	52.6	25.9

表 4.10　A6（準最適巡回＋見過ごし枝刈り）の探索結果（マージン=0）

	公式迷路群		ランダム迷路群	
	移動区画数	進入区画数	移動区画数	進入区画数
平均値	**275.3**	172.1	**267.0**	143.1
最大値	442	248	480	213
中央値	276.0	173.0	266.0	145.0
最小値	60	46	52	27
標準偏差	63.9	33.6	53.5	26.0

4.5.3　考察

図 4.2、図 4.3 より、探索アルゴリズムおよび迷路群によらず、理想迷路最短経路からのマージンが小さくなるにつれて、進入区画数が減少していることが分かる。このことから、見過ごし枝刈りが機能していることが確認できる。なお、マージンが十分に大きくなると見過ごし枝刈りは通常の枝刈りになるため、ここでの折れ線はマージンの増加にしたがって一定値に漸近していっている。このことは、他に示した同形状の折れ線のグラフにおいても言えることである。

図 4.4、図 4.5 を見ると、マージンが小さくなるにつれて、移動区画数が減少していることが分かる。枝刈りと要探索区画の見過ごしは背反するものであると予想していたが、下に凸な二次曲線のようにはならず、マージンが 0 のときに最も効率が良いという結果になった。マージンが 0 のときは、見過ごしが最も多く生じている状態であり、同一の迷路においては巡回しようとする区画は最も少ない。探索が進むと理想迷路最短経路はよりコストの大きなものに更新され、見過ごしていた区画を巡回しなくてはならなくなる。それにもかかわらず結果が良いということは、見過ごしていた区画が探索から除外されたということである。また、見過ごしによる再探索がいずれの迷路でも起こらないとは考えづらいため、再探索のコストを帳消しにするほど多くの枝刈りが起こったと見ることもできる。単に巡回する区画を減らせば、枝刈りが起こりやすいとは限らない。枝刈りが発生するには、暫定迷路最短経路が更新されコストがより小さいものになる必要がある。これは、スタートとゴールを結ぶような経路の区画を通るようにしたほうが良いということである。ここでの結果を考慮すると、そのような経路の中でも巡回する区画が理想迷路最短経路に近いほど、枝刈りが発生しやすいと考えることができる。

以上より、理想迷路最短経路以外の区画は見過ごすようにしたとき、探索を最も効率化できることが分かった。このときの探索結果（表 4.9、表 4.10）は、先述の探索アルゴリズム（表 4.7）よりも効率的であった。また、近傍巡回と準最適巡回の結果の差を、通常の枝刈りの場合（公式

迷路群：$|290.0 - 292.0| = 2.0$、ランダム迷路群：$|310.2 - 314.3| = 4.1$）とマージンが 0 の見過ごし枝刈りの場合（公式迷路群：$|275.5 - 275.3| = 0.2$、ランダム迷路群：$|266.0 - 267.0| = 1.0$）とで比較すると、迷路群の違いによらず、マージンが 0 の見過ごし枝刈りのほうがわずかながら小さい。通常の枝刈りは、見過ごし枝刈りにおける理想迷路最短経路からのマージンが十分に大きいものであると考えることができる。マージンが小さくなると、見過ごす区画が多くなる反面、巡回する区画が少なくなるため、巡回方法の違いを受けづらくなったと考えられる。

移動区画数の標準偏差について示した図 4.6、図 4.7 では、探索アルゴリズムによる差異はないように見えるが、迷路群によって折れ線の形状が異なることが分かる。ランダム迷路群では、平均値のときと同じように、移動区画数の標準偏差は、マージンが小さくなるにつれて減少している。一方の公式迷路群では、値の振れが大きいため確かなことは言い難いが、マージンが小さくなるにつれて増加しているように見える。ランダム迷路群では、理想迷路最短経路からのマージンが小さくなるにつれて探索の安定性も高くなるが、公式迷路群は逆に低下するということが分かった。このような違いが表れたのは、4.1.4 節で憂慮事項として述べた、公式迷路群とランダム迷路群のどちらかに何かしらの偏った特徴があるということが原因と考えられる。ただし、この影響は移動区区画数の標準偏差に現れているものであり、すでに見たとおり、移動区画数の平均値には見られなかった。効率的に探索を行えることは間違いないが、迷路の特徴によっては安定性が低くなるようである。

4.5.4 まとめ

ここでの検証によって、次のことが分かった。

- 最短経路探索の効率化には、枝刈りによる効果が大きい。
- 枝刈りは、巡回路が理想迷路最短経路に近いほど起こりやすい。
- 巡回する区画は理想迷路最短経路に近いほど、探索の効率性は高くなる。
- 公式迷路においては、巡回する区画は理想迷路最短経路に近いほど、探索の安定性は低い。
- 巡回する区画が少ないほど、巡回順の違いによる効率の差は小さくなる。

当初は、見過ごしを多く行うことと少なく行うことは相反するものとして、適度な見過ごしを行ったときに最も効率的になると予想していたが、見過ごしは積極的に行ったほうが良いという結果になった。ただし、これは枝刈りが起こることを期待した上でのことであると考えられる。枝刈りを行う場合は、理想迷路最短経路の近くを通るように動作を計画することが望まれる。

4.6　最短経路探索に関する提案のまとめ

既存の探索アルゴリズムの検証によって、探索の効率化に関して次のような知見を確認した。

(a) ランダムであるより、目的達成に関する情報を考慮したほうが良い。
(b) 探索中に得た情報を利用したほうが良い。
(c) 未知情報の取得より、目的の達成を優先したほうが良い。

また、最短経路探索に関する提案の検証によって、探索の効率化について次のような結果が得られた。

・ 枝刈りは、探索の効率化に有効である。
・ 巡回路の最適化は、探索の効率化に効果が小さい。
・ 理想迷路最短経路から外れる区画は、極力見過ごしたほうが良い。

枝刈りは、知見の(a)と(b)を満たすものである。しかし、巡回路の最適化は(a)と(b)には合致するものの、(c)には反していた。そして、理想迷路最短経路から外れる区画は極力見過ごしたほうが良いという結果は、枝刈りに(c)を加えたものと見ることができる。

前述の知見は探索の研究においては当然のことのように見えるが、先行研究がほぼないような最短経路探索問題においても有効であることが確認できた。

第5章　結言

5.1　本研究のまとめ

マイクロマウス競技における迷路探索、特に最短経路探索に関して効率化に有効な提案を行った。その検証においては、実際の競技会で使用された公式の迷路では数が不十分であると考えたため、公式迷路の作成方法を参考にした迷路生成方法を提案し迷路数を補った。また、生成した迷路と公式迷路との類似性を判断するために、公式迷路の作成方法を考慮した指標である局所経路パターンによる評価を提案し、生成迷路が公式迷路と類似性があることを確認した。最短経路探索に関する提案を行うにあたっては、既存の探索アルゴリズムについて検証を行い、公式迷路 253 種類およびランダム迷路 10,000 種類を用いることによって、各アルゴリズムの性能比較において、先行研究に対し信頼性の高い検証結果を示した。

最短経路探索に関する提案においては、問題を「どこを探索するか」、「どの順番で探索するか」という 2 つの要素に分割して考え、探索が不要な区画を判定し除外する方法（枝刈り）と、探索が必要な区画の巡回路を想定し最適化する方法（巡回路の最適化）を示した。移動区画数評価における検証の結果、枝刈りはその有無によって約 68 %にまで効率化できることを確認したが、巡回路の最適化はわずかに効率化される程度であり、枝刈り比べて効果は小さかった。

上記 2 つの提案に関連して、通常は探索が必要な区画を見過ごすと再探索のために余計な動作を行うことになるが、枝刈りを行うようにしておくと見過ごした区画が枝刈りされ、結果的に探索しなくても良くなる場合がある。枝刈りを行う場合は見過ごすことと見過ごさないことは二律背反の関係にあると予想し、どの程度の見過ごしによって最も効率的に探索できるかを調べた。その結果、予想に反して見過ごしは極力行ったほうが良いということが分かった。ただし、枝刈りを多数発生させる必要があり、この検証では、「未知」の壁は「なし」として扱った迷路におけるスタート-ゴール間の最短経路（理想迷路最短経路）の区画を巡回するようにすると、枝刈りが多く起こることを見出した。なお、このときの探索アルゴリズムを用いると、現在の競技会で迷路の最短経路を見つけるために行われる全面探索に対して、約 58 %にまで探索を効率化できることを確認した。

5.2　今後の課題

本研究では、マウスの動作の評価関数は基本的にどのようなものでも良いという考え方を用いていた。これによって、競技者ごとの機体に適した効率的な最短経路探索を行うことができる。しかし、この評価関数について、実際にどのようなものまで用いることができるか示していない。少なくとも本研究で用いた移動区画数評価はその使用に耐え得ると考えられるが、本研究成果の価値が実際の競技会で見出されるためには、評価関数に関する検討が必要である。

マイクロマウス競技では、様々な知識や技術、ノウハウが求められるが、ソフトウェアに関連する議論がなされることは少ない。特に迷路探索に関しては、1984 年に提案された足立法以来ほとんど進展がない。この原因としては、思い付いた探索アルゴリズムを検証する環境を用意することが難しいということが考えられる。3 m 四方になる実際の競技迷路台やシミュレータを確保することもそうであるが、公式迷路を入手することが困難であるということがある。本研究では、公式迷路を用意するにあたり大会運営団体の事務局に問い合わせたが、迷路一覧というものは存在しないとのことであったため、過去 35 年分の大会資料を借りることによって入手した。得られたのは全体の 8 割程度であるが、これほどの数の公式迷路が揃ったことは今までにないと思われる。多くの競技者がこれらを容易に入手できるようになれば、探索に関する議論が活発になることが期待できる。また、マイクロマウス迷路のように、35 年の歴史を持ち実際のロボットの走行を想定して人の手によって作られた問題空間は、どこを探しても見つからないはずである。マイクロマウスに限らず、探索やその他の分野において、マイクロマウス迷路は有意義な研究資料になり得ると考えている。今後は、大会運営側に協力する形で、多くの迷路を手軽に利用できるようにする取り組みを行うつもりである。

最後に、本研究において確認できた範囲で最も効率の良い最短経路探索アルゴリズムを示す。

(1) 理想迷路最短経路を導出する。
(2) 導出した理想迷路最短経路上の区画のうち、「未探索である」かつ「現在地から最も近い」区画を取得する。取得できない場合は、迷路のスタートまで戻るように動作を決定する。
(3) 現在地から、(2)で取得した区画までの理想最短経路を導出する。
(4) 導出した理想最短経路にしたがって動作を決定する。

今後は、これ以上に効率的な探索アルゴリズムが見つかることを期待する。

謝辞

本研究にあたり、ご意見、ご指導を頂きました、知能システム学講座　末廣尚士教授　工藤俊亮准教授　冨沢哲雄助教に感謝いたします。

また、過去に競技会で使用された迷路図につきまして、収集へのご協力と、本論文への掲載許可を頂きました、公益財団法人ニューテクノロジー振興財団　田代泰典事務局長　マイクロマウス大会運営関係者の皆様に感謝いたします。

そして、学生生活を送るにあたり、お世話になった、知能システム学講座の皆様　金森研究室の皆様に感謝いたします。

参考文献

[1] 井谷 優, “マイクロマウスの歩んだ路,” 日本ロボット学会誌, Vol. 27, No. 9, pp. 979-982, 2009.

[2] 小島 宏一, 加藤 雄資, 福井 善朗, 中村 文一, “開かれた環境が育むマイクロマウスの技術進化,” システム制御情報学会, Vol. 55, No. 7, 2011.

[3] S. Mishra, P. Bande, “Maze Solving Algorithms for Micro Mouse,” IEEE International Conference on Signal Image Technology and Internet Based Systems, pp. 86-93, 2008.

[4] J. Cai, X. Wan, M, Huo, J. Wu, “An Algorithm of micromouse Maze Solving,” 10th IEEE International Conference on Computer and Information Technology(CIT), pp. 1995-2000, 2010.

[5] H. Dang, J. Song, Q. Guo, “An Efficient Algorithm for Robot Maze-Solving,” Second International Conference on Intelligent Human-Machine Systems and Cybernetics(IHMSC), pp. 79-82, 2010.

[6] Z. Cai, L. Ye, A. yang, “FloodFill MazeSolving with Expected Toll of Penetrating Unknown Walls for Micromouse,” IEEE 14th International Conference on High Performance Computing and Communications(HPCC), pp. 1428-1433, 2012.

[7] 石田 亨, 新保 仁, “実時間探索による経路学習,” 人工知能学会誌, Vol. 11, No. 3, pp. 411-419, 1996.

[8] R. E. Korf, “Real-Time Heuristic Search,” Artificial Intelligence, Vol. 42, pp. 189-211, 1990.

[9] Toru Ishida, “Real-Time Bidirectional Search: Coordinated Problem Solving in Uncertain Situations,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 6, 1996.

[10] K. Knight, “Are Many Reactive Agents Better Than a Few Deliberative Ones?,” International Joint Conference on Artificial Intelligence(IJCAI), 1993.

[11] T. Ishida, R. E. Korf, “Moving Target Search,” International Joint Conference on Artificial Intelligence(IJCAI), pp. 204-210, 1991.

[12] T. Ishida, “Moving Target Search with Intelligence,” Artificial Intelligence(AAAI), Proceedings of the 10th National Conference, pp. 525-532, 1992.

[13] 石田 亨, “移動目標探索アルゴリズムとその性能改善,” 人工知能学会誌, Vol. 8, No. 6, pp. 760-769, 1994.

[14] T. Ishida, R. E. Korf, “Moving-Target Search: a Real-Time Search for Changing Goals,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, No. 6, pp. 609-619, 1995.

[15] 公益財団法人ニューテクノロジー振興財団, “マイクロマウスクラシック競技規定,” http://www.ntf.or.jp/mouse/micromouse2014/kitei_classic_since2014.html, 2014.

[16] S. Koenig, “A Comparison of Fast Search Methods for Real-Time Situated Agents,” Autonomous Agents and Multi Agent Systems(AAMAS), Proceedings of the 3rd International Joint Conference, pp. 864-871, 2004.

[17] C. Hernandez, P. Meseguer, “LRTA*(k),” International Joint Conference on Artificial Intelligence(IJCAI), Proceedings of the 19th, pp. 1238-1243, 2005.

[18] V. Bulitko, G. Lee, “Learning in Real Time Search: a Unifying Framework,” Journal of Artificial Intelligence Research, Vol. 25, pp. 119-157, 2006.

[19] S. Koenig, M. Likhachev, “Real-Time Adaptive A*,” Autonomous Agents and Multi Agent systems(AAMAS), Proceedings of the 5th International Joint Conference, pp.281-288, 2006.

[20] D. C. Rayner, K. Davison, V. Bulitko, K. Anderson, J. Lu, “Real-Time Heuristic Search with a Priority Queue,” International Joint Conference on Artificial Intelligence(IJCAI), Proceedings of the 20th, pp. 2372-2377, 2007.

[21] C. Hernandez, P. Meseguer, “Improving LRTA*(k),” International Joint Conference on Artificial Intelligence(IJCAI), Proceedings of the 20th, pp. 2312-2317, 2007.

[22] Y. Björnsson, V. Bulitko, N. R. Sturtevant, “TBA*: Time-Bounded A*,” International Joint Conference on Artificial Intelligence(IJCAI), Proceedings of the 21st, pp. 431-436, 2009.

[23] 水澤 雅高, 栗原 正仁, “障害物密度に応じた迷路探索問題の何度指標と実時間探索アルゴリズムの性能解析,” 人工知能学会論文誌, Vol. 21, pp. 266-275, 2006.

[24] 新保 仁, 石田 亨, “実時間探索の収束性について,” 人工知能学会誌, Vol. 13, No. 4, pp. 631-638, 1998.

[25] 浅野 健一, 高速マイクロマウスの作り方―勝てるロボコン, 東京電機大学出版局, 2000.