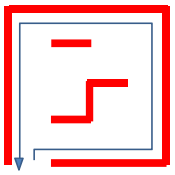
迷路を短時間で全探索するアルゴリズムの研究

玉川学園中学部　1年　國吉　仁志

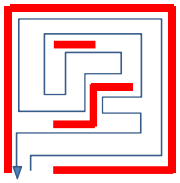
【要旨】私は被災したビルに見立てた広さ不定の迷路を自律走行で探索し、被災者をより多く見つけ、スタート地点に戻るロボットについて研究している。災害時、多くの被災者を短時間で見つけることは重要である。本研究の目的は、ロボットが走行して迷路の全ての場所を探索するとき、必要な実走行時間を最適化することである。広さが既知の迷路を探索する速さを競うマイクロマウスでは迷路の概要を調べるために全探索アルゴリズムが使われる。本研究では、このアルゴリズムを広さ不定の迷路に利用できるように改良した、不定形迷路対応全探索アルゴリズムを提案し検証する。

不定形迷路対応全探索アゴリズム

【右手法】
右側の壁に沿って迷路を解く方法
長所：プログラミングが簡単
短所：右側の壁に繋がっていない場所へは進めない



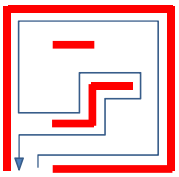
【拡張右手法】
最初は右手法で迷路を進むが、同じマスに進まないように仮想の壁を作りながら進んでいく方法
長所：全ての場所を進める
短所：同じ道を2度通るため、無駄が多い

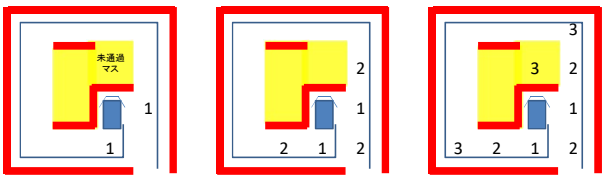


【マイクロマウスで使われるアルゴリズム】

1. 全てのマスにフラグを立てる（迷路の広さが既知であることより可能）
2. フラグの立っている適当なマスを次に進むマスとして決める
3. 2で決めたマスに通過したマスのフラグは消しながら足立法で進む
   足立法:壁情報などがわからなくてもスタートのマスから目標のマスに到達できるアルゴリズム
4. 全てのフラグが消えるまで2～3を繰り返す
5. スタート地点に最短経路で戻る

【広さ不定の迷路への適用】 ⇨ 不定形迷路対応全探索アルゴリズム
本研究では広さは不定なため、フラグを立てることができない。
➡フラグは立てず、2の動作でロボットから最も道程の短いマスを選ぶようにする（既に通過したマスに繋がっているマスのみ選ぶ）
（2の動作でロボットから最も道程の短いマスを選ぶ方法）
コスト：あるルールに基づいて、各マスをランク付けした値。ここでは、ロボットのいるマスからそのマスへの移動にかかるマス数





①ロボットのいるマスから1回で移動できるマスのコストを決める

②最もコストの小さいマスの隣のマスのコストを決める

③コストの決まった未通過のマスを最も道程の短い未通過のマスとして決定。現れない場合は②を繰り返す

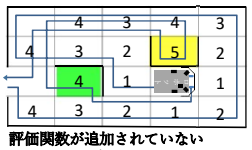
④未通過のマスがなかったなら、スタート地点に決定する

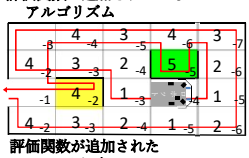
【アルゴリズムの改良】
改良前:今の場所から「最も近い未通過マス」を優先して行く
改良後:なるべくゴールに遠い未通過マスを優先して行く
⇨コスト決定のルールを下の式のように変更

コスト ＝ ロボットのいるマスからそのマスへの移動にかかるマス数 － スタート地点からの距離

・スタート地点からの距離による影響度を変えて検証するため、スタート地点からの距離には比例定数をかけて実験も行う。

コスト ＝ ロボットのいるマスからそのマスへの移動にかかるマス数 － スタート地点からの距離*比例定数


評価関数が追加されていないアルゴリズム

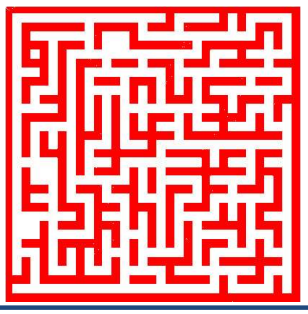

評価関数が追加されたアルゴリズム

検証方法

本研究では環境条件（電池残量や複数迷路の作成など）の問題を考慮させないで、探索アルゴリズムのみを比較検証するため、コンピュータ上でシミュレーションすることとした。

1. 実際にロボットが前進や右回転、左回転にかかる時間を計測する
2. 右図のような広さ30*30の迷路を自動生成する
3. 作成した迷路をそれぞれのアルゴリズムで解き、探索に要した右回転回数と左回転回数、前進回数を調べる
4. 調べた回数と計測した走行時間から迷路の探索にかかる走行時間を算出する

2～4を500回繰り返し、シミュレーションした走行時間、前進回数、回転回数の平均で比較し検証する。



結果・考察

【結果】

アルゴリズム	全探索成功回数	走行時間の平均（秒）
右手法	0	713.62
拡張右手法	500	2944
不定形迷路対応アルゴリズム	500	2444.97
改良版 比例定数:1	500	2461.47
比例定数:0.1	500	2439.67
比例定数:0.2	500	2431.9
比例定数:0.3	500	2420.11
比例定数:0.4	500	**2418.2**
比例定数:0.5	500	2418.65
比例定数:0.6	500	2433.15

【考察】

17%の短縮に成功

最短時間で走行できた

- 右手法以外のアルゴリズムで全探索ができた
- 拡張右手法に比べ、適用したアルゴリズムは約17%走行時間が短縮した。拡張右手法で道を往復する無駄がなくなったからだと考えられる
- 改良したアルゴリズムでは走行時間が適用したアルゴリズムより少し長くなった。スタート地点からの距離が優先されすぎ、ロボットからの道程の長さが無視されたからだと考えられる。
- 改良に比例定数をかけた場合は0.1～0.6の全てで適用したアルゴリズムの走行時間より短縮されていた
- その中で最も走行時間が短いのは比例定数が0.4の時で、適用したアルゴリズムより約1.1%走行時間が短縮された
- 改良したアルゴリズムの比例定数が0.4の時が最も走行時間が短かった

【まとめ】

今回はマイクロマウスで使われる全探索アルゴリズムを広さの不定な迷路に適用し、更に改良することで拡張右手法より約18%走行時間を短縮することができた。

【今後の課題】

実際の迷路には障害物などロボットの走行の邪魔になる物があるため、障害物も考慮したコストの計算方法、経路の導出方法を考えていきたい。