

## 探索

### 状態空間と探索木

#### コスト無しグラフの探索

幅優先探索, 深さ優先探索, 反復深化探索

#### コスト付きグラフの探索

ブラインド探索

貪欲探索, 最適探索

ヒューリスティック探索 (最良優先探索)

貪欲最良優先探索, A\*探索

-----

## 探索

15 パズルのようなパズルを解く問題や、迷路におけるロボットの経路計画の問題について考える。そのような問題を解く過程は、一つの状態から次の状態へと移る操作を繰り返して、初期状態から最終状態にたどり着く過程としてみることができる。このような状態の全体からなる空間のことを状態空間と呼び、初期状態から最終状態へ遷移する過程を見つけることを探索と呼ぶ。ここでは、状態と遷移は離散的に定義されており、また遷移は確定的に起こるものとする。このような探索を効率よく行うアルゴリズムについて紹介する。なお、本書では扱わないが、より発展的な内容として、連続的な状態を扱う探索や、将棋やチェスなど対戦相手に勝つための手を探すゲーム木探索などがある。

### コスト無しグラフの探索

まず、遷移にかかわるコストを考慮しない場合、すなわち、経路の良し悪しを区別することなく、とにかく最終状態に至る経路が見つかればよいという問題の場合について考える。大きく2つの方法がある。

一つは深さ優先探索であり、行き止まりになるまで進み、最終状態にたどり着かなかったら、直近の分岐に戻って別の経路を探索する、というものである。メモリ使用量が少ないという利点があるが、ゴールが近くにあってもなかなか見つからない、あるいは複数のゴールがあるときに遠くのゴールを先に見つけてしまう、という欠点がある。実装する際には、スタックを使う、あるいは再帰呼び出しを使う方法が一般的である。

もう一つは幅優先探索であり、まず直近の状態をすべて探索して、最終状態にたどり着かなかったら、それぞれの状態の次の状態をすべて探索する、というものである。欠点として、メモリの使用量が多いという欠点があるが、最終状態が近くにある場合に効率よくみつかることができるという利点がある。実装する際には、待ち行列を使うことが一般的である。

深さ優先探索と幅優先探索を組み合わせた方法として、反復深化探索と呼ばれる方法がある。これは、深さに制限をつけて深さ優先探索を行い、徐々に深さを深くしていく、というものである。単純な深さ優先

探索では、深さが無限ある場合やループがある場合に解を見つけることができない、という問題があるが、反復深化探索ではこれを回避することができる。逆に、幅優先探索に比較して、反復深化探索には、メモリ使用量が節約できるという利点がある。

### コスト付きグラフの探索

次に、遷移にかかわるコストを考慮して、なるべく少ないコストで最終状態にたどり着く経路を見つけようとする場合について考える。将来のコストに関する推定値を利用しない方法（ブラインド探索）と、利用する方法（ヒューリスティック探索）の大きく2つにわけることができる。表にこれから紹介する方法の関係を示す。

### コスト付きグラフの探索

	未来のみ考慮	履歴も考慮
ブラインド探索 (推定値を利用しない)	貪欲探索 (現在→次が最小)	最適探索 (初期→次が最小)
ヒューリスティック探索 (最良優先探索) (推定値を利用する)	貪欲最良優先探索 (次→最終が最小)	A*探索 (初期→次→最終が最小)

将来のコストに関する推定値を利用しない方法は、ブラインド探索と呼ばれる。大きく分けて、貪欲探索と最適探索の2つがある。貪欲探索は、次の遷移コストが最小になる遷移を選ぶというものである。最適探索は、初期状態からの遷移コストの和が最小になるような遷移を選ぶというものであり、グラフ探索におけるダイクストラ法と同じものである。最適探索は、最小コストの経路を見つけることが保証されている反面、全探索方向をすべて網羅しようとするので、非常に探索範囲が広くなりいつまでたっても探索が終わらない可能性がある。

このような欠点を解決し、より効率よく探索を行う方法として、将来のコストを推定し、その推定値が小さくなる方向へ探索を進める、というヒューリスティック探索（最良優先探索）と呼ばれる方法がある。より具体的には、現在の状態から最終状態に至るまでのコストの和を推定し、それを次の遷移を選ぶ際の基準として利用する。推定値の計算方法としては、ユーザが事前知識により明示的に計算方法を指定する方法と、データから機械学習により自動構築する方法などがある。

ヒューリスティック探索には、大きく分けて、貪欲最良優先探索とA\*探索がある。貪欲最良優先探索は、遷移先の状態から最終状態に至るまでのコストの和の推定値を最小化するような遷移を選ぶ、というものである。一方、A\*探索は、遷移先の状態から最終状態に至るまでのコストの和の推定値と、初期状態から遷移先の状態までのコストの和の合計を最小にするような遷移を選ぶ、というものである。推定値が真のコストを超えないことが保証されているとき、A\*探索は最適性を持つことが知られている（真のコストに近いほど効率が良くなる）。最適探索（ダイクストラ法）は、推定値をゼロとしたA\*探索ということができる。