

G検定対策講座 第3回

探索・推論

AIが「考える」を実現した古典技術



今日学ぶこと

1

探索・推論とは

探索木という土台

2

基本の探索アルゴリズム

幅優先・深さ優先・ハノイの塔・総当たり

3

ゲームAIの基盤

ミニマックス法・アルファベータ法・モンテカルロ法

4

古典AIの代表システム

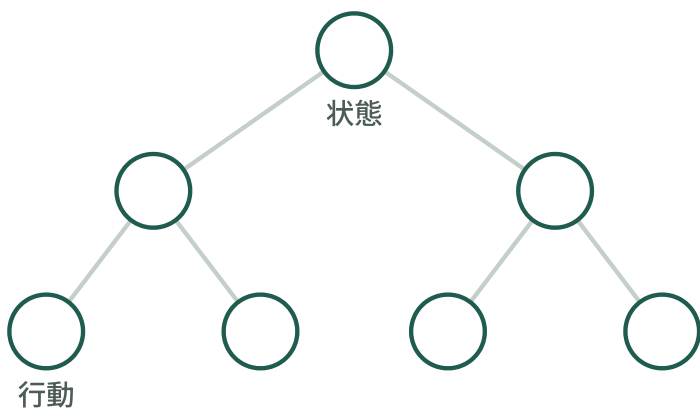
対話のSHRDLU・プランニングのSTRIPS

AIはどうやって 「考えて」いる？

実はコンピュータは、可能な選択肢を順番に試しているだけ

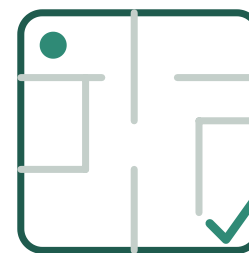
探索 と 推論

古典AIの中核技術。この2つを組み合わせ、チェス・迷路・ロボットの行動計画を解いてきた



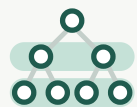
探索

選択肢を**実際に試しながら**目標へたどり着く
丸（ノード）＝状態／枝＝行動



推論

次に**どの枝へ進むか**を論理的に導く
「左は壁 → だから右へ」



基本その1

幅優先探索

同じ深さをまるごと調べてから、次の深さへ



浅いところから順に見るので
最短経路が必ず見つかる

対比

強み / 弱点

● 強み

最短経路が必ず見つかる

カーナビの経路案内などに向く

✕ 弱点

同じ深さの場所を全部覚えておくため

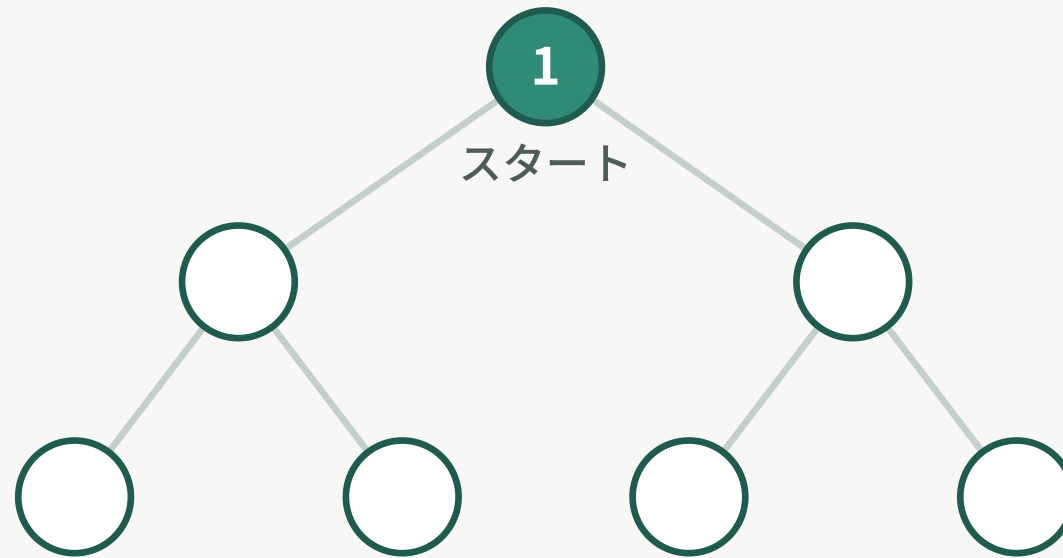
メモリを多く使う

POINT

木が横に広いほどメモリを食う。次のページから、進み方を目で追ってみる

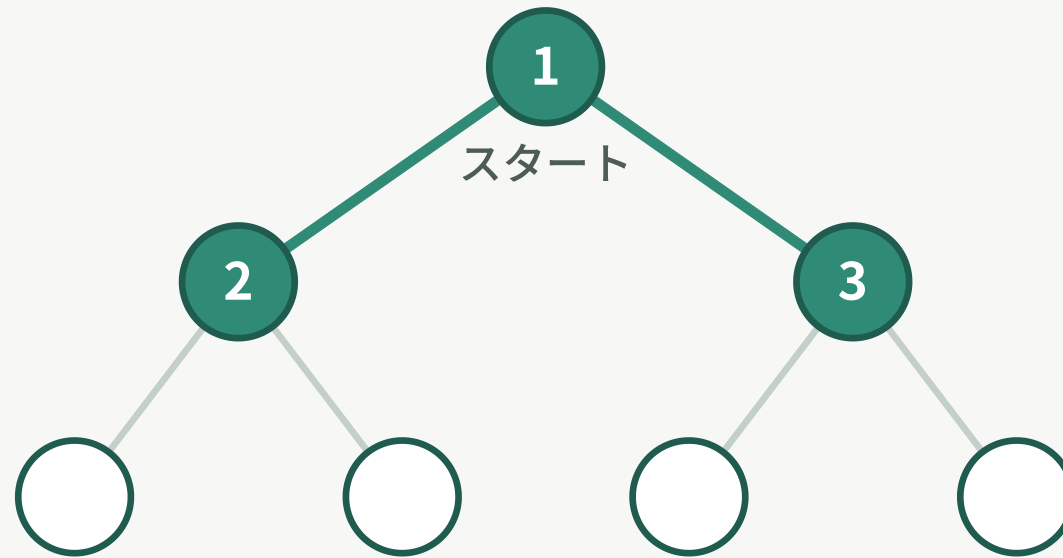
幅優先探索 | 進み方

まずは出発点。ここが探索のスタート



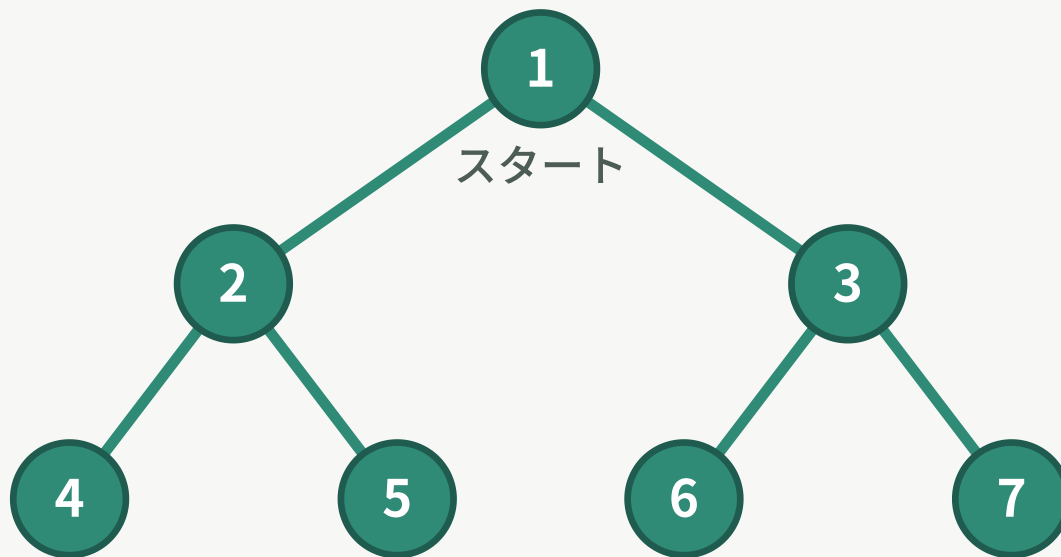
幅優先探索 | 進み方

一歩でいける場所を、先に全部調べる



幅優先探索 | 進み方

同じ深さを調べ終わったら、次の深さへ



POINT 浅いほうから順に見るので、最初に見つけたゴールがそのまま最短になる



基本その2

深さ優先探索

一本の枝を選んだら、行き止まりまで縦に潜る



いま通っている一本の道だけ
覚えればよい

対比

強み / 弱点

● 強み

メモリが少なくて済む

同じ深さを全部覚える必要がない

✕ 弱点

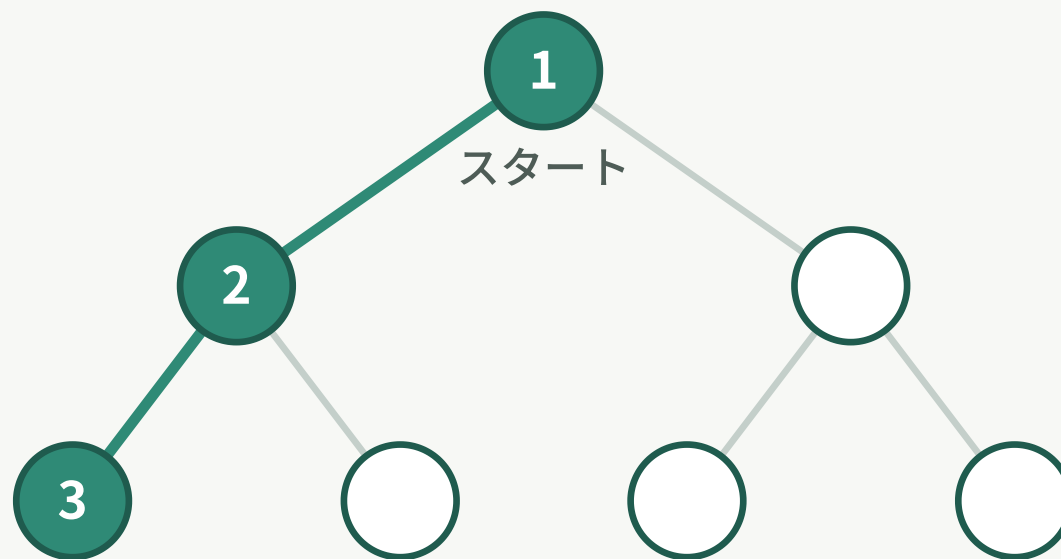
最初に見つけた道が最短とは限らない

解けるかの確認やパズルを解くのに向く

POINT 幅優先とは正反対の進み方。次のページから、目で追ってみる

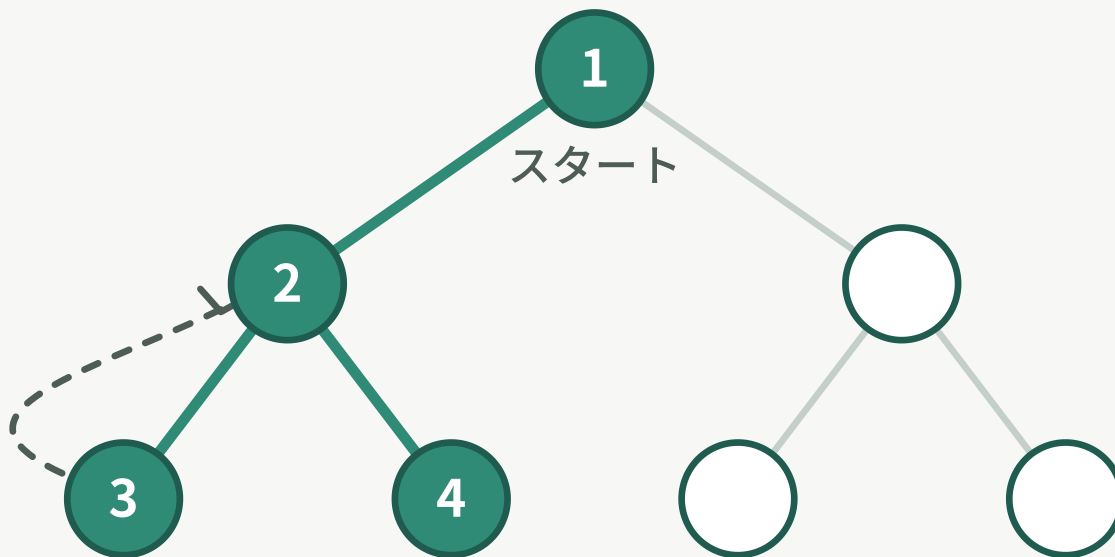
深さ優先探索 | 進み方

一本の道を選んだら、行けるところまで一気に奥へ



深さ優先探索 | 行き止まりに当たったら

一つ前まで戻って、隣の枝へ。これをくり返して全体を回る



1883年



ハノイの塔

3本の柱と円盤で、
探索アルゴリズムの代表的なテスト問題



フランスの数学者
エドゥアール・リュカが考案

具体例 ルールと、AIとの関わり



左端の塔を、まるごと別の柱
へ移す



大きい円盤を
小さい円盤の上に置けない

一度に動かせるのは一枚だけ

円盤が1枚増えるたび
手数はおよそ2倍にふくらむ

POINT 探索の難しさが、この小さなパズルに詰まっている



総当たり

ブルートフォース

考えられる可能性を、片っぴしから全部試す



日本語では力任せ探索、
あるいはしらみつぶし

対比

必ず見つかる / でも爆発する

● 確実

答えは必ず見つかる

パスワード解読・初期のチェスAI

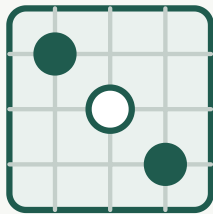
× 限界

可能性が増えると計算時間が爆発的にふくらむ

選択肢の多い問題には向かない

POINT

だから、無駄な枝を切り落とす枝刈りや、あたりをつけて探すヒューリスティックが生まれた



ゲーム木

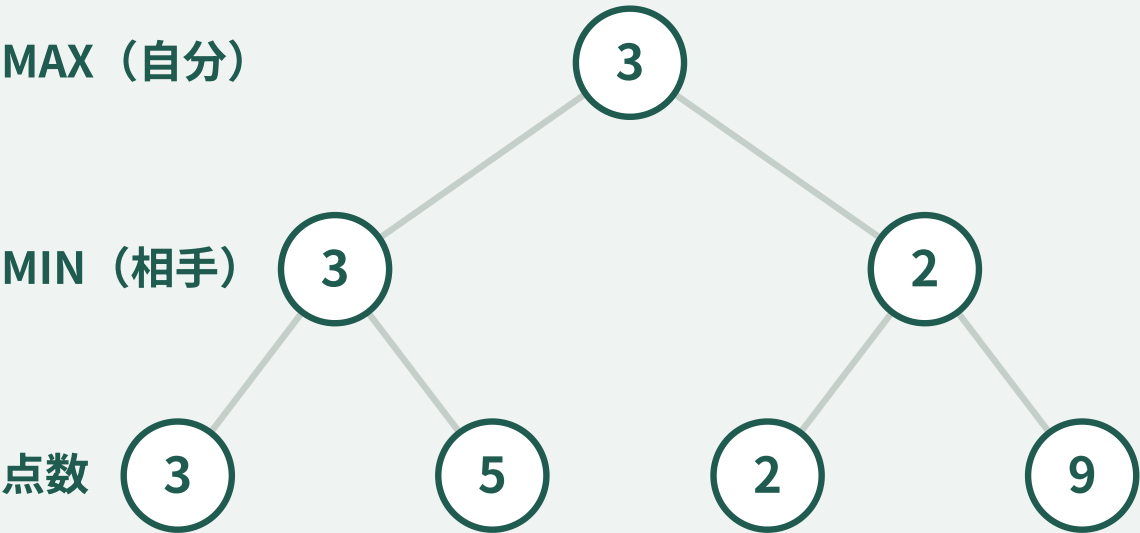
ミニマックス法

二人で対戦するゲームで、**最善手**を計算する方法



1928年 フォン・ノイマンの
ミニマックス定理が土台

具体例 末端の点数を、上へ引き上げる



自分は点数を最大にしたい
(MAX)

相手は点数を最小にしたい
(MIN)

相手も最善を打つ、と読んで手
を決める

POINT 数手先まで読んだ点数を、木の下から上へ引き上げて、いまの最善手を決める



1956年ごろ

アルファベータ法

ミニマックス法を、もっと効率よくした改良版



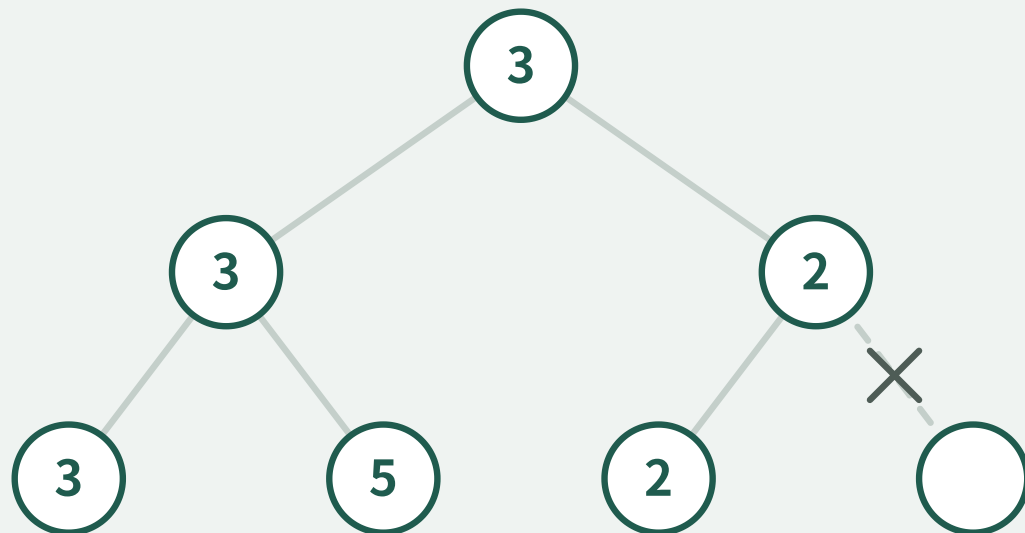
ジョン・マッカーシーらが提唱

具体例

結果に影響しない枝は、途中で切る（枝刈り）

MAX

MIN



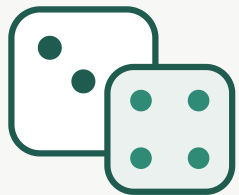
相手がどうせ選ばない枝だと
分かった瞬間、その先は読まない

これを枝刈りとよぶ
英語でプルーニング

読む深さは同じでも
計算量が大きく減る

POINT

チェスや将棋のAIでは、このアルファベータ法が実装の基本になっている



1940年代

モンテカルロ法

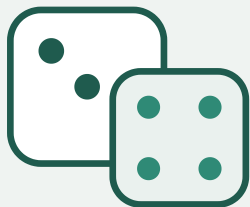
でたらめな試行をとにかく何度も繰り返す



ウラムとフォン・ノイマンが
マンハッタン計画で考え出した

具体例

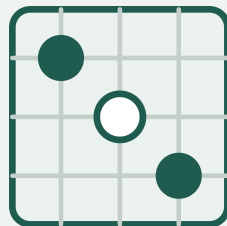
囲碁AIでの使われ方



ランダムに
打ってみる



何度も繰り返す



終局までシミュ
レーション



勝率を数える

評価

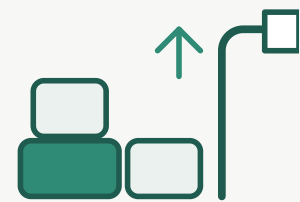
勝ちやすい手を選ぶ

AlphaGo の核のひとつ
モンテカルロ木探索

POINT

組み合わせが天文学的に多い囲碁はミニマックス法では読み切れない。だから発想を変えた

1968～70年



SHRDLU (シュルドウル)

英語で話しかけると積み木を動かせる自然言語処理システム

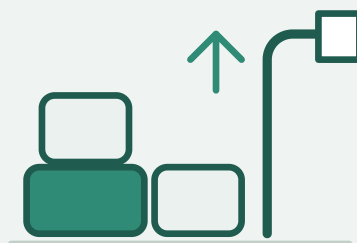


マサチューセッツ工科大学の
テリー・ウィノグランドが開発

具体例 積み木の世界 (ブロックス・ワールド)

「赤いブロック
を取って、
青の上に置いて」

英語で指示する



ロボットアームが
その通りに動く



「さっき持っていたのは何？」
という
文脈をふまえた質問にも答えた

ただし通用するのは積み木の世
界だけ

POINT 自然言語理解の初期の成功例であり、同時に古典AIの限界も浮き彫りにした



1971年

STRIPS (ストリップス)

ロボットに

目標までの行動の順番を自分で考えさせる



スタンフォード研究所の
ファイクスとニルソンが開発

具体例 動作ごとに「前提条件」と「効果」を決めておく

動作 持つ

前提 箱の前にいること

効果 箱を持っている状態になる

→
目標から逆算

目標：箱を別の部屋へ運べ

歩く → 持ち上げる → 運ぶ → 置く
この手順をAIが自動で組み立てる

POINT 古典的なプランニングの基礎。いまのロボットの動作計画にも受け継がれている

