

ゲーム理論と最適化手法 第4回: 遺伝的アルゴリズム の基礎

上田 俊

佐賀大学理工学部

Email: sgrueda@cc.saga-u.ac.jp

Web: <https://www.fu.is.saga-u.ac.jp/sgrueda/>

2019年10月28日

最適化問題

- 最適化問題: 解空間から, 与えられた制約条件を満たし, 目的関数を最大化 (最小化) する解を探索する問題
- 例: 2次関数の最小化 ★
 - 解空間: $x \in \mathbb{R}$
 - 制約条件: $1 \leq x \leq 4$
 - 目的関数: $\max x^2 - 4 \cdot x + 5$

人間が解く方法

- $x^2 - 4 \cdot x + 5 = (x - 2)^2 + 1$.
- 下に凸なので、頂点は最小の点となる.
- 最大の点は $x = 1$ か $x = 4$ のどちらか.
- $x = 1$ のとき $(1 - 2)^2 + 1 = 2$,
 $x = 4$ のとき $(4 - 2)^2 + 1 = 5$.
- よって、解は $x = 4$ である.
- コンピュータはこのように解析的に解くのは苦手 (というかできない).

数値計算

- 解析解が得られるとは限らない。★
 - 「ボールを1番遠くに投げた方が勝ちのゲーム」 - 風向き，重力の偏差，磁場とか全部計算に入れる？
 - 「交通渋滞を避けるための赤信号の灯火時間」 - そもそもどうモデル化？
- 強化学習で**数値解**を求める。
- 数値計算は厳密解を得られる保証はないので，解析的に解ける場合は用いない方がよい。

組合せ最適化問題

- 組合せ最適化問題: 解が組合せや順列で表される問題
- 例: 遠足のお菓子選び
~~バナナはおやつに入りますか?~~
 - 解空間: セブンイレブンで買える菓子
 - 制約条件: 合計 300 円以内
 - 目的関数: max 満足度
- ナップサック問題 (knapsack problem) と呼ばれる.

組合せ最適化問題の解き方

- どうやって解きますか? ★
- 全探索
 - 組合せの数が少ないときは有効
 - 組合せの数がすぐ多くなる問題では時間がかかりすぎる.
- 山登り法
 - 少しずついい解が得られて, 満足できる解を得たら終了できる.
 - 局所最適解になり, 永遠に最適解が見つからない時もある.

組合せ最適化問題の難しさ

- 組合せ最適化問題は難しい
→ だから面白い!
- 後半のゲーム理論でも組合せ最適化問題が度々登場する.
- ただし, 多くの問題が解くのが難しいとされている (e.g. NP 完全問題).
 - 詳細は, 第 8 回 組合せ最適化問題 で.
- 解法 (アルゴリズム) がとても重要

非厳密解法

- ヒューリスティックアルゴリズム: 最適解が得られる保証はないが, 比較的短時間で最適解に近い解が得られる解法. 非厳密解法とも呼ばれる.
- 様々な問題に適用可能な汎用的な手法を**メタヒューリスティックアルゴリズム**と呼ぶ.
 - 山登り法もメタヒューリスティックアルゴリズムのひとつ

メタな展開

- ところで、メタ (meta) の意味を説明できますか? ★
- 「高次の」、「上位の」といった意味
- 「作中人物がそれが作品であることを認識していること」だけではないので注意
 - 数学や物理でこの意味だけを知っていると混乱するかもしれない...

遺伝的アルゴリズム (1/2)

- 解を遺伝子として表現し，生物が環境に適用するように，遺伝子 (解) を改良していく様子を模倣したアルゴリズム
- イメージは山登り法に近い．局所最適解に陥らない工夫がある．(遺伝子の突然変異)
- 進化計算アルゴリズム (evolutionary computation algorithm) とも呼ばれる．

遺伝的アルゴリズム (2/2)

- 遺伝的アルゴリズム (genetic algorithm) のアイデア
 - 環境に適応できる個体ほど、自分の遺伝子を残せる。→ 良い解の遺伝子をより多く受け継ぐ。
 - 2個体の交叉により子を作る。→ 2つの解から別の解を作る。
 - ときどき突然変異が起こる。

遺伝子

- 遺伝的アルゴリズムでは、解を遺伝子 (N 次元ベクトル \vec{x}_i) で表す.
- $\vec{x}_i = (x_1^i, x_2^i, \dots, x_N^i)^T$
 - 以降, 転置記号 (T) は省略する.
- N はどう遺伝子として表現するかによって様々.

遺伝子の例

- 仮にお菓子の候補をが 8 つあるとする。
(バナナは除く)
- お菓子を買う場合は 1, そうでない場合は 0 とした 8 次元ベクトル \vec{x}_i で解を表す.
 - お菓子 #3 を買う. $\rightarrow x_3^i = 1$
 - $\vec{x}_i = (1, 1, 1, 0, 0, 0, 1, 1)$ ならば, お菓子 #1, #2, #3, #7, #8 を買い, 他は買わない.

適応度

- 遺伝子の良さを適応度 (fitness value) という.
- 目的関数そのままということも多い.
- 遺伝的アルゴリズムの性能に直結するが設計が難しい.
- ナップサック問題の場合は,
$$v(\vec{x}_i) = \sum_{j \in N} v(j) \cdot x_j^i$$
 とすることが多い. ($v(j)$ はお菓子 #j の満足度)

交叉

- ふたつの遺伝子 (\vec{x}_i と $\vec{x}_{i'}$) から別の遺伝子 ($\vec{x}_{i''}$) を作ること.
- 一点交叉, 二点交叉, 一様交叉といった様々な交叉方法がある.
- 一点交叉の例: 交差点 $i = 3$ より前を親 1 から, 後を親 2 から受け継ぐ.
 - $\vec{x}_i = (1, 1, 1, 0, 0, 0, 1, 1),$
 $\vec{x}_{i'} = (0, 1, 0, 1, 0, 1, 0, 1)$
 $\rightarrow \vec{x}_{i''} = (1, 1, 1, 1, 0, 1, 0, 1)$

突然変異

- 局所最適化にならないための工夫.
- ナップサック問題の場合は, 単純に, いずれかの値を反転させる.
 - $\vec{x}_i = (1, 1, 1, 0, 0, 1, 1)$
→ $\vec{x}_{i'} = (1, 1, 1, 0, 1, 0, 1, 1)$

小レポート

- 進化計算アルゴリズムは生物の進化だけでなく，粘菌の性質を計算に利用する「粘菌コンピューティング」やアリの生態を利用した「アントコロニー最適化」などがある．
- 最適化問題に適用できそうな生物の生態を挙げ，行う最適化を例示してください．(考えても，調べても OK)