

9. モンテカルロ法による最適化 Optimization by Monte Carlo method

乱数を使った数値計算の方法は全て「モンテカルロ法 Monte Carlo method」と呼ばれます。賭博場とぼくぼで有名なモナコ公国のモンテカルロ Monte Carlo 地区にちなんでこのように呼ばれるようです。



Fig. 9.1 モンテカルロ・ルーレット

モンテカルロ法は最適化以外の目的、例えば積分を計算したり方程式を解く目的で用いられる場合もあります。

最適化の目的に限っても、モンテカルロ法的一种である「擬似焼鈍」や「遺伝的アルゴリズム」の人気は高く、実際に良く使われます。

モンテカルロ法には、以下の特徴があります。

- (1) プログラミング（コーディング）が容易である。
- (2) 計算の効率は良くなく、高い精度は期待できない。
- (3) 繰り返しの回数を増やせば増やすほど正解に近づくことが期待できる。
- (4) 多次元（多変数関数）の最適化に特に適している。
- (5) 並列化（分散）処理に適している。

注意は必要と思いますが、将来さらに有望な方法となることは間違いありません。この章では、最適化計算に限定して、モンテカルロ法の基本的な骨格、擬似焼鈍、遺伝的アルゴリズムについて紹介します。

9-1 ランダムサンプリングとインポートランスサンプリング Random sampling and importance sampling

モンテカルロ法による最適化では、「ランダムなパラメータの組み合わせの生成」「結果の評価」を繰り返す手順を取るのが基本です。

ただし、ランダムと言っても、あらゆる可能性を同じ確率で選択する「ランダム・サンプリング」だけでなく、重要そうなところを重点的に探索する「インポートランス・サンプリング」と呼ばれるやり方があります。

インポートランス・サンプリングのために用いられる代表的な方法がランダム・ウォーク（酔歩；酔っぱらい歩き）です。マルコフ連鎖モンテカルロ法 Markov chain Monte Carlo (MCMC) 法とよばれる場合もあります。「マルコフ連鎖」とは、とりうる状態が離散的である「離散状態マルコフ過程」と同じ意味で、位置が離散的に表現される場合も含むのですが、実際には時間が離散的な過程を指すことが多いようです。実際上「離散時間マルコフ過程」のことを「マルコフ連鎖」と呼ぶと構いません。

「マルコフ過程」とは、「未来の状態が現在の状態だけで決まり、過去の状態にはよらない」という性質を持つ確率過程です。

9-2 棄却サンプリング（採択・棄却法） Rejection sampling (accept-reject method)

現在位置からランダムな方向に、ランダムな距離移動すること（ランダム・ウォーク；酔歩）を試み、「現在の値より小さい値になれば進み」、「そうでなければ進まない」という選択を繰り返すことで、最小位置付近の重点的な探索を実現することができます。この方法はリジェクション・サンプリング rejection sampling あるいは acceptance-rejection 法、accept-reject 法などとも呼ばれます。日本語では棄却サンプリング、採択・棄却法と呼ばれますが、どの呼び方も必ずしも一般的ではないかもしれません。この方法では、偽最小 フォールス ミニマ false minima に陥るかどうか、初期位置の影響を強く受けます。

9-3 メトロポリスの方法 Metropolis method

ある温度 T を仮定します。位置（状態）を \mathbf{r} から $\mathbf{r} + \Delta\mathbf{r}$ にランダムに変化させたときにエネルギーが E から $E + \Delta E$ に変化したとします。

メトロポリスの方法では、 $\Delta E \leq 0$ なら確率 1 で、 $0 < \Delta E$ なら確率 $\exp(-\Delta E/k_B T)$ で変化後の位置 $\mathbf{r} + \Delta \mathbf{r}$ を採択します。ここで、 k_B はボルツマン定数です。この方法は「有限な温度での挙動」を模擬するものとも言え、（偽最小ではなく）正しい最小に到達する可能性が高くなると期待できる場合があります。

実際には以下のようなアルゴリズムが用いられます。

【メトロポリスのアルゴリズム】

位置をランダムな方向、ランダムな距離に動かすこと（提案）を考え、新しい位置のポテンシャルと前の位置のポテンシャルの差 ΔE を計算する。

(1) $\Delta E \leq 0$ なら新しい提案を採択する。

(2) $0 < \Delta E$ の場合、0 から 1 の一様乱数 t を発生して

$\exp(-\Delta E/k_B T) < t$ なら新しい提案は棄却し、

$t \leq \exp(-\Delta E/k_B T)$ なら新しい提案を採択する。

このアルゴリズムで「 $\Delta E \leq 0$ なら常に採択し、 $0 < \Delta E$ なら確率 $\exp(-\Delta E/k_B T)$ で採択する」ことになっていることを確認します。

(i) $0 < \Delta E$ のとき、 $0 < \exp(-\Delta E/k_B T) < 1$ の関係が成立します。

(ii) $0 \leq t \leq 1$ の一様乱数 t に対して、 $\exp(-\Delta E/k_B T) < t$ となる（棄却される）確率は $1 - \exp(-\Delta E/k_B T)$ であり、 $t \leq \exp(-\Delta E/k_B T)$ となる（採択される）確率は $\exp(-\Delta E/k_B T)$ で表されます。

「新しい提案を採択する確率（遷移確率）を $\min(1, e^{-\Delta E/k_B T})$ とする」という表現のしかたをする場合もあります。 $\min(a, b)$ は a と b の小さい方の値を返す関数です。

「棄却サンプリング」では、 $0 < \Delta E$ の場合必ず引き返す（棄却する）ので、これをメトロポリスのアルゴリズムにあてはめると、「絶対零度 $T=0$ の場合」に対応します。

メトロポリスの方法で有限の T を用いる場合、 $0 < \Delta E$ の場合でも（少し悪くなくても）一定の割合でそのまま進みます。 T の値が大きいほど進む確率が高くなり、ランダム・サンプリングに近くなります。

メトロポリス・アルゴリズムで十分な試行回数を繰り返すと、エネルギーの分布（確率密度） $P(E)$ が「ボルツマン分布に比例する状態になる」とされています。つまり、

$$P(E) \propto \exp\left(-\frac{E}{k_B T}\right) \quad (9.3.1)$$

という式で表されるような状態になり、これは「詳細釣り合い detailed balance」が成立するからと説明されます。

詳細釣り合いとは、「順過程と逆過程が同じ頻度で起こること」を意味します。エネルギー E の状態から $E' = E + \Delta E$ の状態への変化（順過程）の起こる確率 $P(E \rightarrow E + \Delta E)$ と、 $E' = E + \Delta E$ の状態から E の状態への変化（逆過程）の起こる確率 $P(E + \Delta E \rightarrow E)$ について考えます。

遷移確率がそれぞれ

$$A(E \rightarrow E + \Delta E) = \min(1, e^{-\Delta E/k_B T}) \quad (9.3.2)$$

$$A(E + \Delta E \rightarrow E) = \min(1, e^{\Delta E/k_B T}) \quad (9.3.3)$$

と表されることから、

$$\begin{aligned} \frac{P(E + \Delta E \rightarrow E)}{P(E \rightarrow E + \Delta E)} &= \frac{P(E + \Delta E) A(E + \Delta E \rightarrow E)}{P(E) A(E \rightarrow E + \Delta E)} \\ &= \frac{P(E + \Delta E) \min(1, e^{\Delta E/k_B T})}{P(E) \min(1, e^{-\Delta E/k_B T})} = \frac{P(E + \Delta E)}{P(E)} e^{\Delta E/k_B T} \end{aligned} \quad (9.3.4)$$

の関係が成立します。式 (9.3.4) の最後の式は ΔE の符号によらず成立することを確認してください。「詳細釣り合い」が成立する場合

$$\frac{P(E + \Delta E \rightarrow E)}{P(E \rightarrow E + \Delta E)} = 1 \quad (9.3.5)$$

なので、式 (9.3.4) から

$$\frac{P(E + \Delta E)}{P(E)} = e^{-\Delta E/k_B T} \quad (9.3.6)$$

となります。この関係が成立するのは、式 (9.3.1) が成立する場合に限られます。

9-4 擬似焼鈍 Simulated annealing

前節のメトロポリスのアルゴリズムを用いれば、「一定の温度 T でのボルツマン分布に従うような状態」を模擬できます。

そこで、高い温度設定から始めて、徐々に温度を下げていくと、「はじめは広い範囲・ランダムな配置をくまなく探索する状態であったのが、温度を下げるにつれて、最小位置付近を重点的に探索するような挙動になる」ことが期待できます。

この挙動は現実の金属材料の焼鈍（しょうどん・やきなまし）のプロセスで、「高い温度で保持することにより歪み^{ひずみ}を解消し、ゆっくり温度を下げることで自然に安定な原子配置となる」のと似ていて、「**擬似焼鈍**」「**シミュレーテッド・アニーリング simulated annealing**」と呼ばれます。略して SA 法と呼ばれる場合もあります。

「現実の『焼きなまし』と似ているのでうまく働くに違いない」と必要以上に思われが

ちな面がある印象を受けますが、確かに実際にうまく働くように見える場合も少なくないようです。

9-5 遺伝的アルゴリズム Genetic algorithm

遺伝的アルゴリズム genetic algorithm は、**ダーウィン Darwin の進化論** をもとにした方法と言われ、特に**自然選択** (自然淘汰 とうた ナチュラル セレクション) と**有性生殖** (sexual reproduction せいしよく セクシュアル) (遺伝子交差あるいは交叉 こうさ, **クロスオーバー** chromosomal crossover クロモソマル クロスオーバー) を真似する性格を持つものです。また普通は**突然変異** mutation ミューテイション が取り入れられます。略して GA 法と呼ばれる場合もあります。

この方法は「現実の有性生殖による『生物の進化』と似ているのでうまく働くに違いない」と(前節の疑似焼鈍と比較してさらに)必要以上に思われがちな面が強いように思いますが、確かに実際にうまく働くように見えることも少なくありません。

セントラル・ドグマ central dogma によれば、遺伝情報はデオキシリボ核酸 Deoxyribonucleic acid; DNA 高分子の中の塩基 (アデニン adenine, グアニン guanine, シトシン cytosine, チミン thymine) 配列として記録されています。これはコンピュータで「数値」あるいは「数値の組み合わせ」が、連続する“1”か“0”かの配列 (ビット列) で表されることと似ています。遺伝的アルゴリズムと呼ばれる方法では「パラメータの組み合わせ」を遺伝子と同じように扱います。

また、遺伝的アルゴリズムでは、異なる遺伝子を持った個体 (異なるパラメータの組み合わせ) を複数保持し、適応度の低い個体を排除するプロセス (自然選択 natural selection) を適用します。

また、生存した個体のうち一定の割合でペアを作り、有性生殖と同じように遺伝子を交叉させ、子を作らせます。このプロセスは交叉 crossover と呼ばれます。交配 mating と呼ばれる場合もあります。

また、それとは別に、一定の割合で、親から受け継いだ遺伝子と無関係に遺伝子の一部をランダムに変更する (突然変異 mutation) を施すこともあります。

ビット列の交叉の仕方にはいくつかの方法が提案されていますが、もっとも単純な「一点交叉」と呼ばれる方法では、交叉点の位置をランダムに選択し、例えば交叉点より上位のビット配列に親 A のビット配列、下位のビット配列に親 B のビット配列を用いて、つなぎ合わせたビット配列を持つ子 C を作ります。結果的に C のビット配列は、親 A と B の中間的な性格を持つものになります。

ビット配列を直接処理することが面倒であれば、例えば、親 A の持つパラメータ x_A, y_A, z_A, \dots と、親 B の持つパラメータ x_B, y_B, z_B, \dots とから、 $0 < r < 1, 0 < r' < 1,$

$0 < r'' < 1, \dots$ の一様乱数を使って, $x_C = (1 - r)x_A + rx_B, y_C = (1 - r')y_A + r'y_B, z_C = (1 - r'')z_A + r''z_B, \dots$ として子 C のパラメータを決めれば, 同じような結果が得られます。

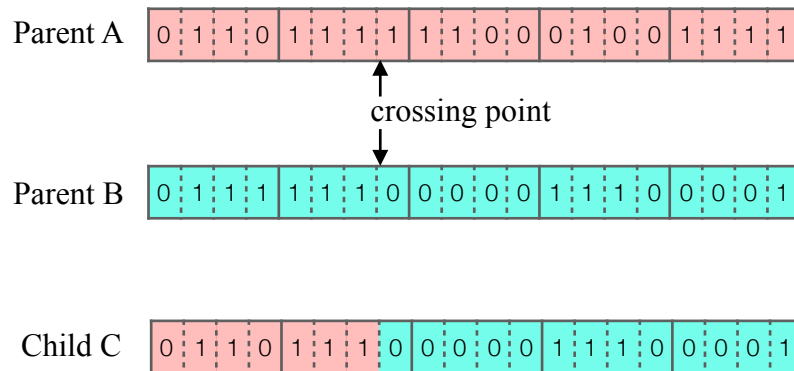


Fig. 9.5.1 遺伝的アルゴリズムで有性生殖（遺伝子交叉）を模擬する例。親 A, B から子 C を生成するとき, ランダムな交叉点でビット列を乗り換える。

遺伝的アルゴリズムを使って複数の変数を含む関数 $f(x, y, z, \dots)$ の最小値を求めるためには, 例えば以下のようにします。

- (I) 初期世代の生成: 100 体の個体 $(x, y, z, \dots) = (x_0, y_0, z_0, \dots), \dots, (x_{99}, y_{99}, z_{99}, \dots)$ をランダムに生成します。
- (II) 自然選択: 関数 $f(x, y, z, \dots)$ の値の大きい (適応度の低い) 21 体の個体を排除します。
- (III) 有性生殖: 関数 $f(x, y, z, \dots)$ の値の小さい (適応度の高い) 40 体の個体を選択します。40 個体から 20 ペアを作成し, 遺伝子交叉により 20 個体を生成します。
- (IV) 突然変異: 新しく 1 個体をランダムに生成し, 個体数を 100 体とします。
- (V) 世代更新: (II) の自然選択に戻ります。

世代交代を繰り返すうちに適応度の高い個体の割合が増えてきて, 最小位置近傍に濃集する挙動が現れます。

遺伝的アルゴリズムでは, 突然変異をまったく入れなくても最適化はできますが, その場合, 棄却サンプリングと同じように偽最小に陥る危険が高くなります。一方で突然変異の割合を大きくしすぎると, ランダム・サンプリングに近くなり, 最適化の効率が低下します。