

# Emergent Computation(EC)

指導院生：渡邊 真也 奥田 環

チーフ：青井 桂子 サブチーフ：伏見 俊彦 小池 政輝

2001年4月17日

本来 EC とは (Evolutionary Computation:EC) 進化型計算のことであるが、それでは SA(Simulated Annealing:SA) やニューラルネットワークが EC の範囲に含まれないので、これより以下 EC とは Emergent Computation(創発的計算)として扱っていくものとする。

## 1 Emergent Computation(EC)

EC(Emergent Computation) は最適化の方法の一種である。最適化の手法としては、大きく分けて 2 つに分けられる。一つは「数理計画法」と呼ばれるもので、これは計算負荷は少ないが、制約条件が厳しく、適用できる問題に限られているので実際問題では使えないことが多い。それに対して、今回紹介する Emergent Computation(創発的計算)は、計算量はかなり膨大だが、制限がなくどのような問題にも対応できるのが特徴である。つまり、計算パワーに任せて問題を解くというような解決法である。その EC には以下のような種類が挙げられる。

- EC(Evolutionary Computation)

進化型計算 (Evolutionary Computation: EC) とは、生物進化を計算機上で模して、適合、学習、最適化などの機能を実現しようという手法である。進化型計算の研究は、いずれも 1960 年代に独立して始まった以下の 3 つのグループの活動を起源としている。

- 遺伝アルゴリズム (Genetic Algorithm:GA) 問題の解を記号列として表現した個体を用い、生物の遺伝的機構と自然選択を比較的忠実に模擬して、個体集合による離散的な探索空間での適応や最適化を扱う。
- 進化プログラミング (Evolutionary Programming:EP) 個体集合と選択を用いることについては GA と共通しているが、新しい探索点の生成にはもっぱら突然変異を用いる。
- 進化戦略 (Evolution Strategy:ES) 実数関数の最適化を対象とし、個体表現には実数値をそのまま用いる。EP と同様、個体集合の選択と突然変異を主とする探索手法である。

- SA(Simulated Annealing)

シミュレーテッドアニーリング (Simulated Annealing:SA) は、最適化問題、特に組み合わせ最適化問題のための汎用的近似解法のひとつである。

- 強化学習

強化学習とは、試行錯誤を繰り返すことによって問題解決の方法を自律的に獲得する機械学習の手法である。

- ニューラルネットワーク

生物の脳細胞や神経回路網の構造、情報処理機能をシステムに取り入れ、高度な情報処理を実現させようというものである。

## 2 Simulated Annealing(SA)

### 2.1 SA の概要

シミュレーテッドアニーリング (Simulated Annealing:SA) は、最適化問題、特に組み合わせ最適化問題のための汎用的近似解法のひとつである。システムの計画や運用などの効率化を考える場合、多くの問題が『組み合わせ最適化問題』として定式化できるが、実際的な問題の多くは厳密な最適解を求めるのは困難である。そこで、この種の問題は満足できる解を求める近似解法が適用されている。SA は Kirkpatrick らによって提案された焼きなまし法を計算機上に模倣した手法である。焼きなまし法とは、(Fig.1 参照) 高温で熔融状態にある物質を徐々に冷却し、整った分子結晶などの低エネルギーの状態を得る方法である。例えば大きなガラスの結晶は非常にゆっくり冷却することで冷やしたりするとガラスの結晶は亀裂が生じて不完全なものとなると思われる。アニーリングのプロセスは物質を多数の粒子の集合体と見なすことで模倣できる。SA は構成が簡単で理解しやすく、適用範囲の広い組み合わせ最適化手法として、人工知能分野や OR(Operation Research) 分野でも研究が進められている。

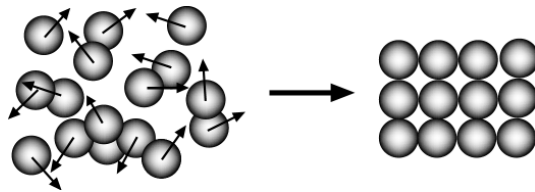


Fig. 1 原子の動き

#### 2.1.1 ランダム法と局所探査法

ランダム法は問題の性質をまったく考慮しない完全な確率的手法である。記述した最適化の事象システムに関して、定義域から無作為に解の候補を選択し、その中で目的関数が最大(最小)となるものを解とする方法である。この方法では論理的には探索回数を無限大に増やせば、最適解を得ることが可能である。しかし、問題空間の解の候補を全て調べ上げると言うのは、あまりにも非現実的である。

SA は実行可能解の部分集合を求めるのに、現在解から近傍解へと繰り返し移動して検索する方法として知られる局所探査法の変形と見なすことができる。従来の局所探査法は最小化問題に対して常に改善方向に解が移動するような降下法が用いられるが、この方法はときとして大域的最適解よりも局所的最適解へ収束する場合がある。降下法で得られた解は全て初期値に依存する。

ある最適化問題の暫定的な解  $x \in F$  に摂動を加えた解  $x' \in F$  を生成する。この解  $x$  に対して、摂動解  $x'$  を生成する範囲を  $N(x)$  と書き『 $x$  の近傍』と呼ぶ。LS では  $x'$  が  $x$  より良い目的関数値を与えるなら暫定解を  $x'$  に移動する、という逐次的な解の改善操作を繰り返す最適化手法である。このような解の挙動を Fig.2 に示す。局所探査法の探査

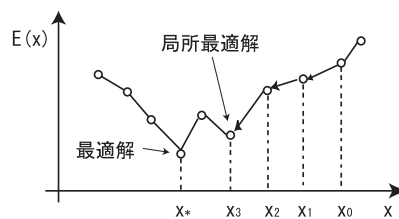


Fig. 2 局所探査法

能力を強化する方法として、次のようなことが挙げられる。

1. 近傍の定義を見直す
2. 初期値を取り直して局所探査法を繰り返す。
3. 局所探査法での暫定解の更新法として目的関数を改善するものだけでなく、改悪となるものも一定の制限のもとで許し、局所最適解からの脱出を可能にする。

SA は 3 番目のアプローチにより局所探査法を拡張したものと考えられる。

## 2.2 SA のアルゴリズム

最適化問題とは有限集合  $F$  を解の探索範囲として、この  $F$  の中で問題関数  $E$  を最小化する要素  $x \in F$  を探し出すことである。

$$\min_{x \in F} E(x) \quad (1)$$

SA は局所探索法で示したように、与えられた初期状態から出発して、エネルギーが確率的に小さくなるように次々と状態を変化させ、最終的には最適な状態になることが期待されるアルゴリズムである。SA では各状態  $x$  に対してエネルギー  $E(x)$  が定義された有限の状態空間が適用可能な対象と考える。この状態空間で SA が局所最適解に収束しにくい理由としては、エネルギー変化を確率的に行っているためである。

### 2.2.1 SA の基本となるアルゴリズム

SA の基本的アルゴリズムは生成関数  $GENERATE(x)$ 、受理関数  $ACCEPT(E, E', T)$ 、徐冷関数  $REDUCE(T_k)$  から成り立つ。SA の基本的なアルゴリズムを Fig.3 に示す。

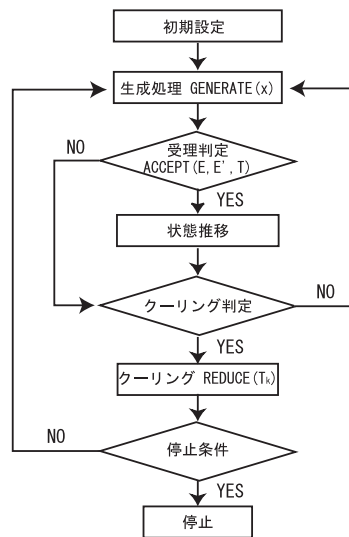


Fig. 3 SA のアルゴリズム

Fig.3 を文字にしてあらわすと次のようになる。

1. 初期設定初期の温度を  $T_k = T_0$  と設定し、初期状態  $x_0$  を与える。
2. 停止条件が満足するまで繰り返す (Loop1)
  - (a) 温度  $T_k$  で十分に探索。つまりここでクーリング判定をする (Loop2)
    - i. 現在の状態から  $GENERATE(x)$  を用いて  $x'$  を生成する。
    - ii. 次の状態  $x'$  のエネルギー  $E' = f(x')$  を計算
    - iii. エネルギーの差分  $\Delta E (= E' - E)$  と温度  $T_k$  より  $ACCEPT$  関数を用いて受理するかを判断。受理する場合は状態推移する ( $x \rightarrow x', E \rightarrow E'$ )
 End Loop2
  - (b) クーリングを行う。 ( $T_{k+1} = REDUCE(T_k)$ )
  - (c) クーリングステップ数をそろえる ( $k = k + 1$ )
 End Loop1
3.  $x$  が最適状態、 $E$  が最適値として出力する。

### 2.2.2 生成処理

生成処理では現在の状態  $x$  を入力とし、次に推移すべき状態  $x'$  を返す。この関数  $GENERATE(x)$  では、状態  $x$  から出発して状態  $x'$  が生起する確率分布  $G(x, x')$  を用いる。組み合わせ最適化問題の場合は、状態  $x'$  は状態  $x$  の近傍にあたり、推移に優先性を与えない場合は 2 式のように等確率推移となる。

$$G(x, x') = \frac{1}{n(x)} \quad (2)$$

### 2.2.3 受理判定

受理判定は、次の状態  $x'$  のエネルギー  $E'$  と現在の状態  $x$  のエネルギー差分  $\Delta E (= E' - E)$ 、および温度パラメータ  $T$  によって、次の状態への推移を受理するか否かの判定を行う。通常は 3 式の Metropolis 基準が採用される。

$$ACCEPT(E, E', T) = \begin{cases} 1 & \text{if } \Delta E < 0 \\ \exp(-\frac{\Delta E}{T}) & \text{otherwise} \end{cases} \quad (3)$$

この基準が返す値は受理する確率で、'1' なら次状態に推移し、'0' なら推移しない。この値は温度  $T$  によって変化し、を見てもわかるように、温度  $T$  は、エネルギーが増大する方向への推移確率に重大な影響を与えるパラメータである。このように温度が高い場合は悪い状態への推移確率も大きくなり、反対に温度が低い場合は良い方向に推移することになる。しかし、どんな温度でも悪い方向への推移確率がゼロになるわけではない。

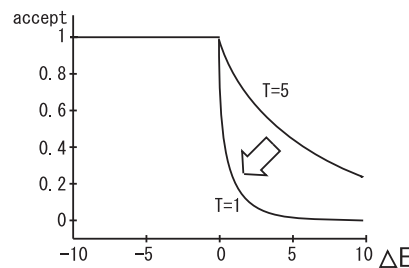


Fig. 4 状態遷移の受理確率

### 2.2.4 クーリング

クーリングでは第  $k$  ステップの温度  $T_k$  を与えて、次のステップの温度  $T_{k+1}$  を返す。最適解の漸近収束性を保証するためには 4 式に示す対数型アニーリング以上に急速に冷やしてはいけない。

$$T_{k+1} = \frac{T_1}{\log k} \quad (4)$$

しかし、それでは現実の応用はあまりにも計算スピードが遅いため、運用上は真の最適解への収束を犠牲にして 5 式に示す指数型アニーリングが用いられる。

$$T_{k+1} = \gamma T_k (0.8 \leq \gamma < 1) \quad (5)$$

クーリングは、各温度で状態  $x$  が平衡状態が実現するまで“十分な探索”を行った後に実行する。この“十分な探索”としては

1. 定めた回数だけ繰り返す。
2. 定めた受理 (または棄却) 回数になるまで繰り返す。
3. 温度幅を十分小さくする場合は 1 回受理するまで繰り返す。

などの実装方法がある。

### 2.2.5 停止条件

SA の処理が停止するための条件には、次のような実装方法がある。

1. アニーリングを定めた回数  $k_f$  だけ繰り返して停止
2. 受理がほとんど起こらなくなったら停止
3. 同じ状態が何度も生成されるようになったら停止
4. 温度が十分低くなったら停止
5. エネルギーの変化またはエネルギーそのものが十分小さくなったら停止

### 2.3 SA の特徴

SA の特徴としては、以下のものが挙げられる。

#### 2.3.1 長所

- (1) 頑強性：SA は用意に局所最適解に捕捉 (トラップ) されず、理論上は真の最適解に、実際には準最適解に到達できる。これは、解品質が改良される方向のみに探索を進めるのではなく、時折改悪する方向も選ぶ仕組みによる。
- (2) 汎用性：枠組み自体が極めて汎用にできているため、広範囲の問題に適用できる。
- (3) 柔軟性：目的関数に対する制約がほとんどない。目的関数は微分可能でなくても、複雑な式でもとまるものであっても、確率的であっても良い。
- (4) 簡易性：実装が容易である。

#### 2.3.2 短所

- (1) 非効率性：SA を用いて最適解を得るには非常に多くの計算量を要する。そのため逐次処理のまま高速なアニーリングを導入する高速化の研究、および並列化して高速化を図る並列化の研究が近年見られる。
- (2) 操作性：パラメータをチューニングする努力が必要となる。特に温度を制御するパラメータのチューニングは非常に困難である。

## 3 温度並列シミュレーテッドアニーリング (TPSA)

### 3.1 SA の並列化

SA は最適化問題を解く有効な手段であるが、最適解を得るためには非常に多くの計算量を必要とする。そのため並列計算機が利用可能になるに伴い、処理不可の重い SA も並列処理の研究対象となった。並列 SA では「同期型」と「非同期型」の2つに分類できる。SA はマルコフ連鎖を次々とたどる処理であるため、本来強い逐次性があり、並列化は概して容易ではない。そのためこれらの並列化では、計算の効率化と解品質の向上のどちらかを犠牲にすることが多い。「同期型」はアニーリングの収束を保証しているがあまり速度向上は得られず、「非同期型」は速度向上のために解の精度を犠牲にしている。

そこで、並列処理による速度向上が得やすく、解の品質の劣化が少ない並列 SA が望まれる。我々の研究室ではこれに対するアプローチとして、温度並列シミュレーテッドアニーリング (Temperature Parallel Simulated Annealing : TPSA) が研究されている。

### 3.2 TPSA の基本アルゴリズム

逐次 SA では、温度  $T$  を温度  $T'$  に減少させるのにたいして、TPSA のアルゴリズムでは温度  $T$  のプロセスから温度  $T'$  のプロセスへ解を渡す。このときの解交換は確率的に行う。TPSA のアルゴリズムを Fig.5 に示す。

逐次 SA のアルゴリズムで温度スケジュールを設定することは、TPSA のアルゴリズムではプロセッサ間の解の交換をいつ行うかを指定することに相当する。TPSA では、プロセッサ間の解交換を確率的に行わせることによって、温度スケジュールを自動化している。すなわち、確率的な解の交換によって、解自身が自分に適した温度スケジュールを選び出してくれることを期待するのである。二つのプロセッサ間ではある確率  $p$  でお互いの解を交換し、確率  $1 - p$

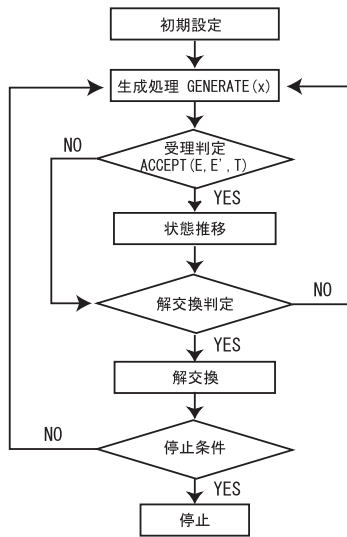


Fig. 5 TPSA のアルゴリズム

で何もしないことを定義する．この確率のことを交換確率という．解交換は買い好感確率に従い，お互いに異なる一定温度で同時並列的にアニーリングをし，ある一定のアニーリングごとに行う最終的に最低温度のプロセッサで得られる状態とエネルギーの各々最適状態と最適値を出力する．逐次 SA と TPSA のクーリングスケジュールを Fig.6 及び Fig.7 に示す．

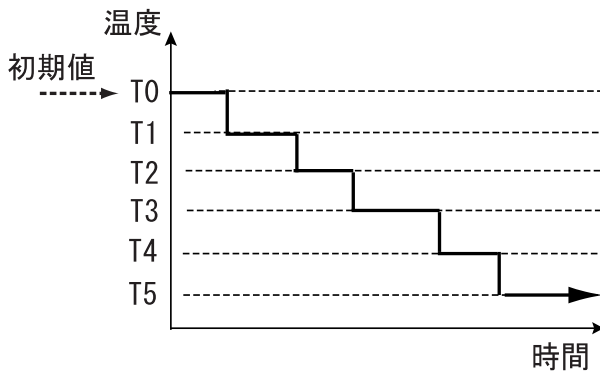


Fig. 6 逐次 SA のクーリングスケジュール

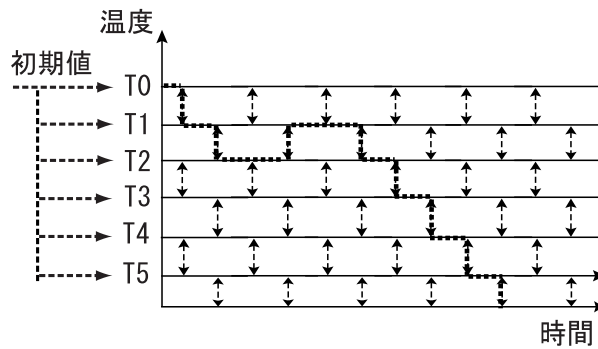


Fig. 7 TPSA のクーリングスケジュール

### 3.3 解交換確率

TPSA を行うにあたり，各プロセッサごとに互いに相異なる温度を与えてアニーリングを行った場合，高温のプロセッサでは比較的自由に解空間が検索されて最適化はあまり進まないのに対し，低温のプロセッサではエネルギー E の減少する方向ばかりに検索が進むので逆に局所最適解に陥りやすい．このことは，単にプロセッサごとに独立にアニーリングを行っただけでは，最低温度のプロセッサで最良の解が得られにくいということを意味している．そこで，プロセッサ間で解の交換を行う必要が出てくる．この解の交換には式 6 のような解交換確率が用いられている．

$$P_{EX}(T, E, T', E') = \begin{cases} 1 & \text{if } \Delta T \Delta E < 0 \\ \exp\left(-\frac{\Delta T \Delta E}{T T'}\right) & \text{otherwise} \end{cases} \quad (6)$$

式 6 は，交換相手が低温（または高温）であるが自分より悪い解（または良い解）を持っていれば全て交換し，交換相手が低温（または高温）で自分よりも良い解（または悪い解）の場合でもある確率で交換するという意味している．

### 3.4 TPSA の長所

TPSA は通常の逐次 SA と比較して、次のような長所を持つ。

- (1) 温度スケジュールの自動化
- (2) アルゴリズムの時間一様性
- (3) 並列処理との高い親和性

プロセッサ間通信が必要となるのは解交換の瞬間のみである。

## 4 研究紹介 (2000 年度の卒業論文より)

### 4.1 SA の連続最適化問題に適用した場合の近傍設計

SA は組み合わせ最適化問題に有効な汎用アルゴリズムである。連続最適化問題に SA を適用する場合、解摂動に用いる近傍の設計が解の精度に影響を与え、目的関数ごとに近傍を調節する必要がある。

Corana は連続最適化問題を SA を用いて解く場合、近傍幅が大きくなりすぎると受理率が低くなりすぎることや、近傍幅が小さくなりすぎると受理率が高くなりすぎること指摘した。そして効率よい探索を行うため解摂動に用いる近傍幅を受理率が 0.5 になるように調節することで、近傍設計を自動化することを提案した。しかしながら、Corana が提案した手法の近傍は局所解から脱出可能程度の近傍幅より小さくなってしまふ。局所解から脱出可能な近傍幅を保つために、一定に保つ受理率を小さくする近傍設計について研究している。

### 4.2 温度並列シミュレーテッドアニーリング

3 節参照

### 4.3 遺伝的交叉を用いた並列シミュレーテッドアニーリング

第 2 回の EC ゼミで行う。

## 5 SA の研究応用例

- 磁気シールド付き MRI 超電導コイルの設計
- 音源パラメータの推定
- たんぱく質の折りたたみと人工設計 (構造解析)
- VLSI とコンピュータ設計
- 画像判別の応用：テクスチャ画像の判別における最適な構造
- オペレーションズリサーチ分野でスケジューリング問題 (ジョブスケジューリング)
- 建物のレイアウト設計に対する商用のパッケージにおけるアニーリング
- ヨーロッパにおける最適な大気汚染除去装置の配置決定：異なる配置にした場合の効果影響 (装置の数が一定)
- DNA マッピング
- 進化ツリー作成問題
- 生産スケジューリング

## 6 課題

SA のソースプログラムを配布するので、各パラメータを変化させることにより結果がどのように変化するか考察する。