

生物に触発された計算 特に最適化計算

パーセプトロン

ホップフィールドネットワーク・ボルツマンマシン

メタヒューリスティクス・ノーフリーランチ定理

疑似やきなまし・タブー探索

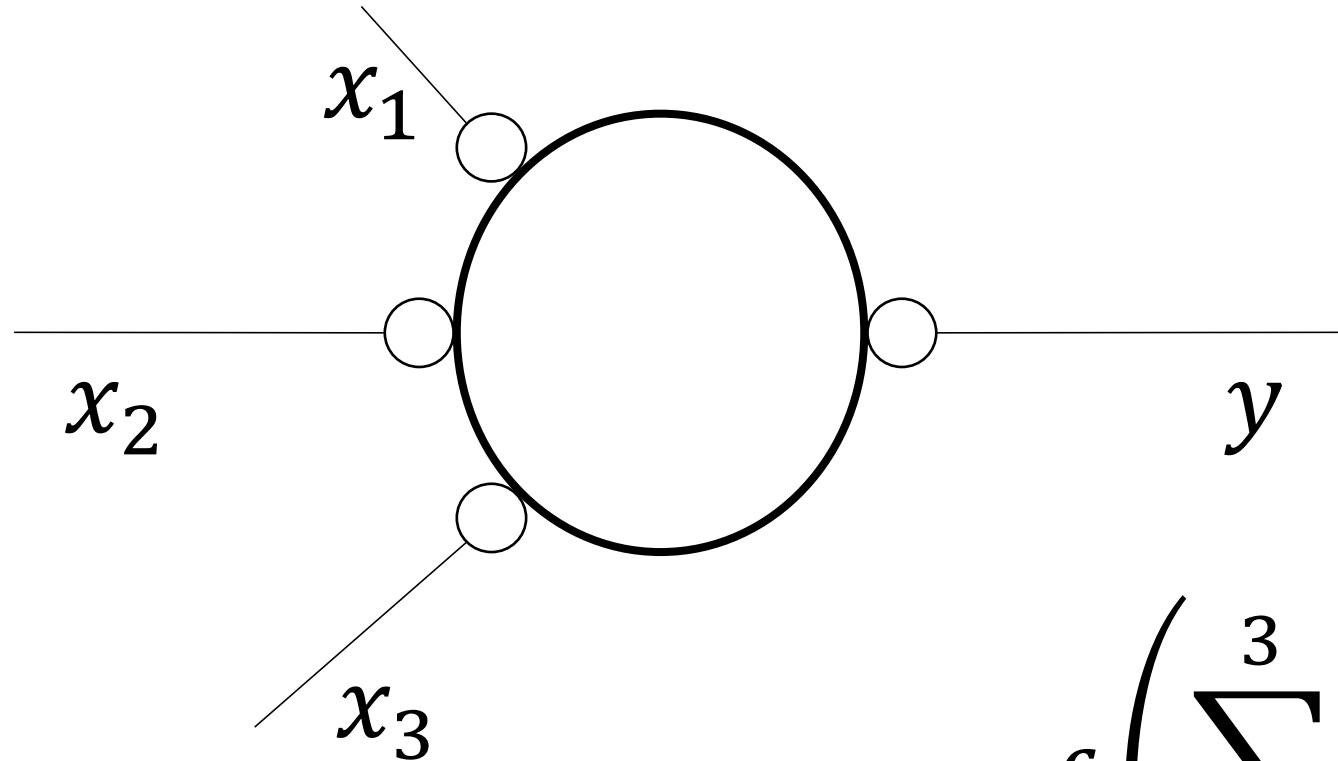
遺伝的アルゴリズム

粒子スウォーム最適化

ニューラルネットワーク

- パーセプトロン
- ホップフィールドネットワーク
- ボルツマンマシン
- ...

ニューロンのモデル



$$y = f \left(\sum_{i=1}^3 w_i x_i - \theta \right)$$

モデルの種類

- x_i, y : 入力と出力
 - 離散 (0 or 1) か、連続 (非負実数)
- w_i : シナプス荷重
 - 通常は負も含む実数
- θ : 閾値
 - 実数
- $f(\dots)$
 - 階段関数
 - ジグモイド関数
 - ...

ヘビサイドの階段関数

$$f(x) = \begin{cases} 0 & (x < 0) \\ ? & (x = 0) \\ 1 & (x > 0) \end{cases}$$

ジグモイド関数

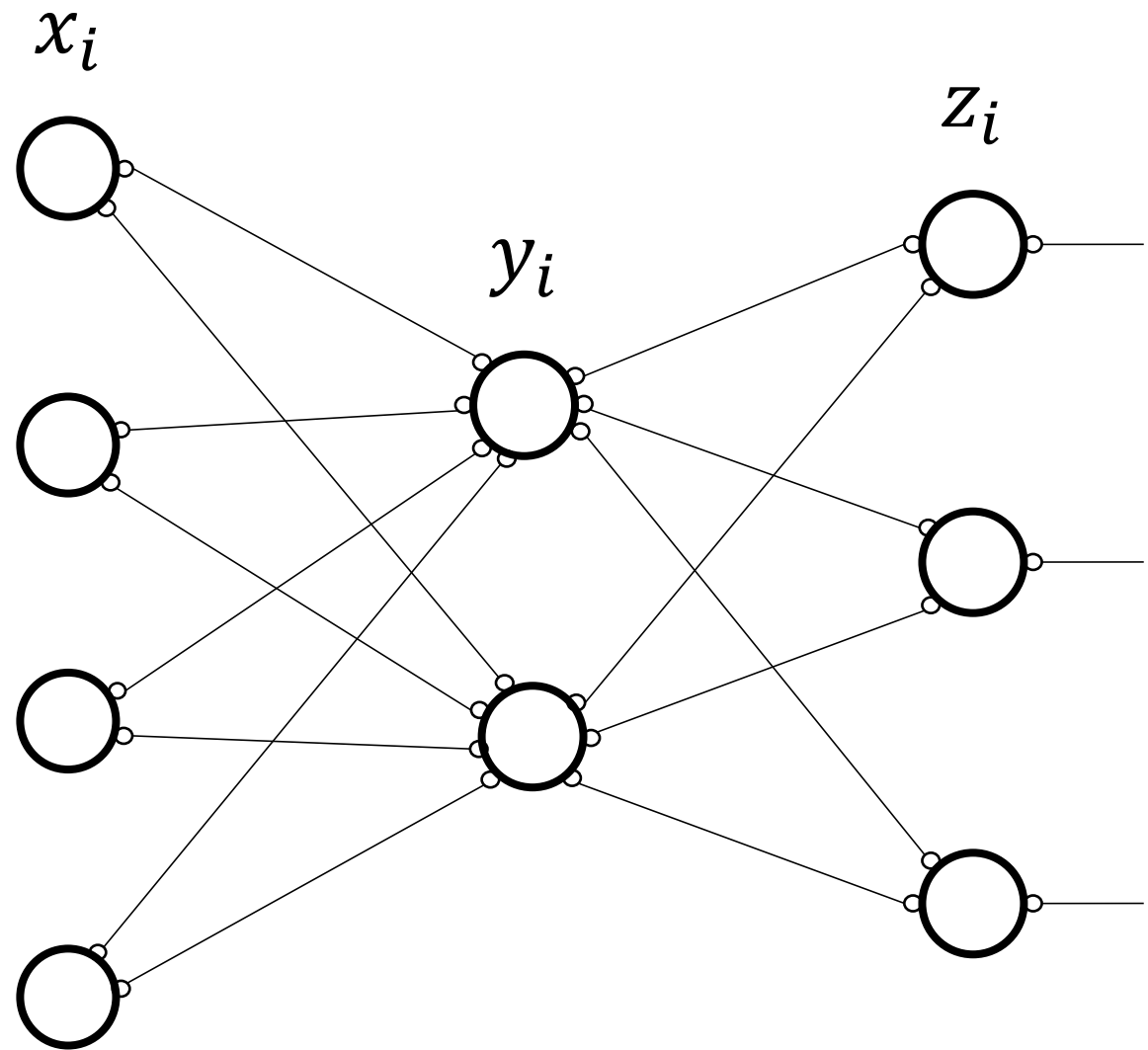
$$f(x) = \frac{1}{1 + e^{-\frac{x}{T}}}$$

ネットワークの種類

- ループも含まれる
 - ホップフィールドネットワーク
 - ...
- ループがない(階層的)
 - パーセプトロン
 - ...

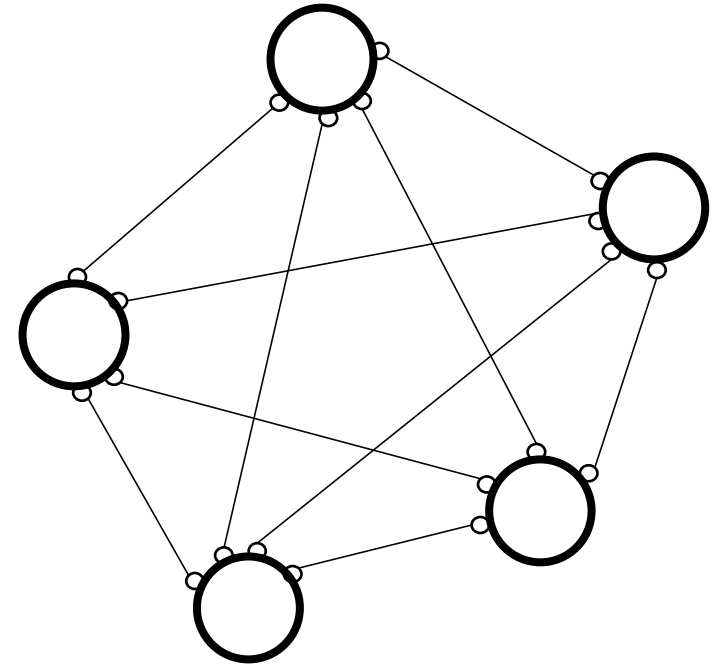
パーセプトロン

- ヘビサイドの階段関数
- 3層の場合
 - x_i : 入力層
 - y_i : 中間層
 - z_i : 出力層
- 各種の分類問題



ホップフィールドネットワーク

- 連想記憶
- 一般のフィードバックを持つネットワーク
- $W_{ij} = W_{ji}$
- $W_{ii} = 0$
- ニューロンの入出力は 0 or 1
 - 連続値にすることもあり



ホップフィールドネットワークの更新

- ランダムにニューロン i を選ぶ
- ニューロン i への入力 $y_i = \sum_{j=1}^N w_{ji} x_j - \theta_i$ を計算
- $y_i < 0$ ならば、ニューロン i の出力 x_i は 0 とする
- $y_i > 0$ ならば、ニューロン i の出力 x_i は 1 とする
- $y_i = 0$ ならば、ニューロン i の出力 x_i は変化させない
- 以上を繰り返す

ホップフィールドネットワークのエネルギー

$$E = -\frac{1}{2} \sum_{i=1}^{i=N} \sum_{j=1}^{j=N} w_{ij} x_i x_j + \sum_{i=1}^{i=N} \theta_i x_i$$

- E は、ネットワークが更新されると、減少する(増加しない)
 - $x_i = 0$ の場合を基準にとると、 $x_i = 1$ の場合のエネルギーは、ちょうど、 $y_i = \sum_{j=1}^{j=N} w_{ji} x_j - \theta_i$ だけ小さい
- したがって、ネットワークの更新を繰り返すと、 E は極小値に到達

ホップフィールドネットワークの応用

- 最適化

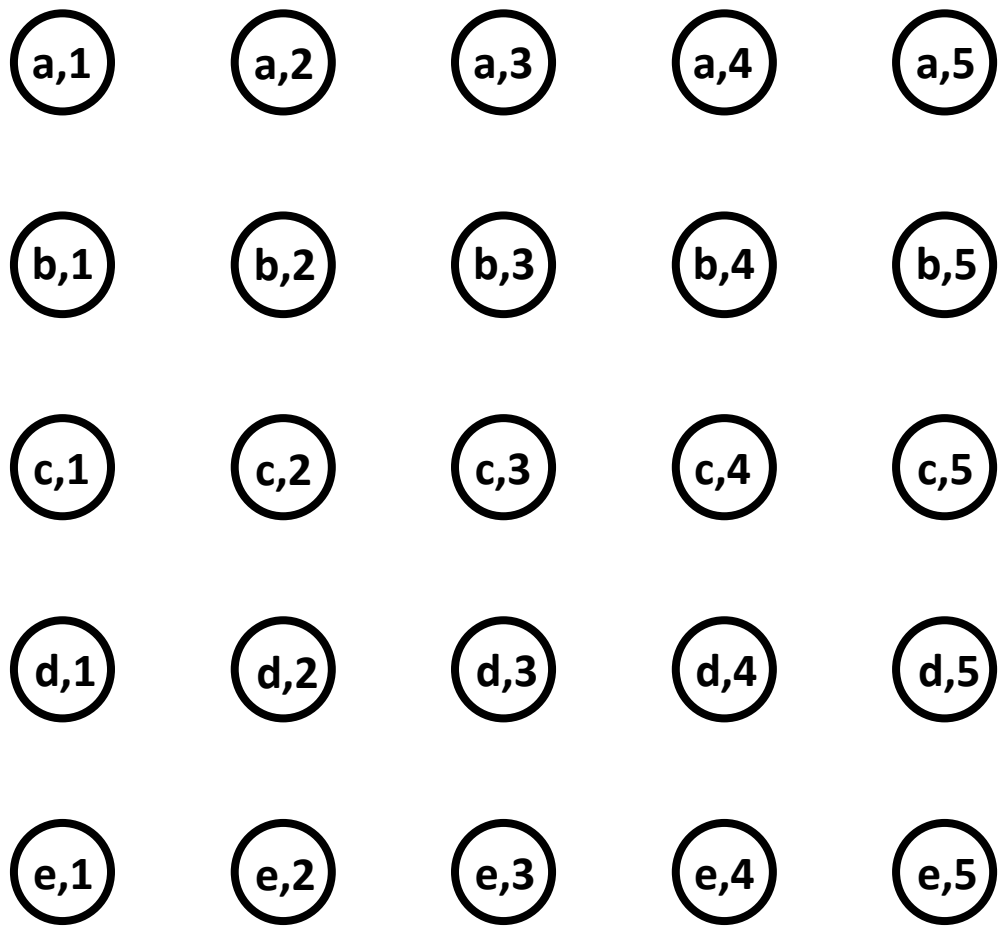
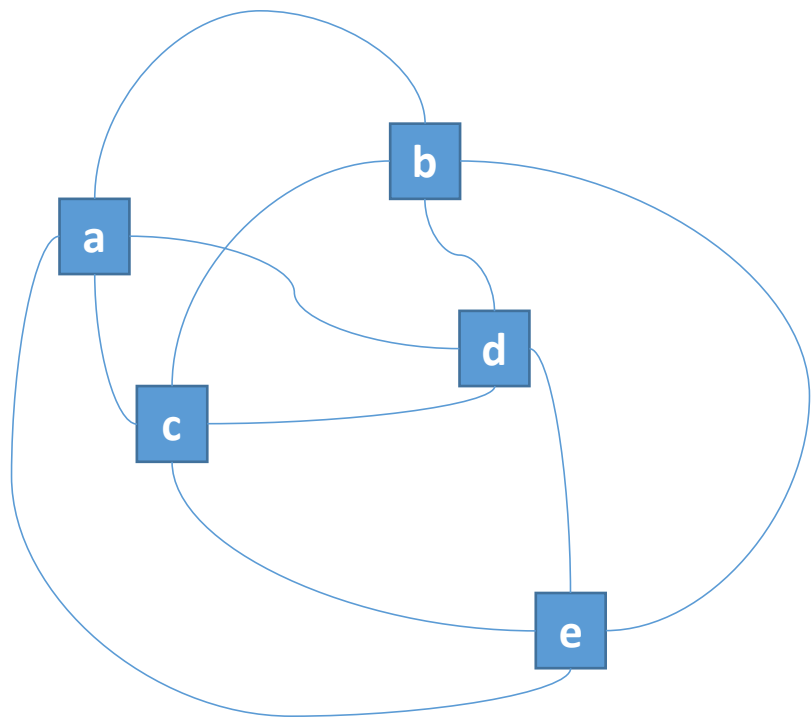
- エネルギーが評価関数と一致するように、ニューロンの種類とシナプス荷重を適切に設計してやる
- 例: 巡回セールスマン問題
 - N 都市に対して、 $N \times N$ 個のニューロン
 - i 番目に訪れる都市が X であるときに、ニューロン (X, i) は 1 とする
 - 都市間の距離と各都市を一度ずつ訪れるという制約をシナプス荷重で表現

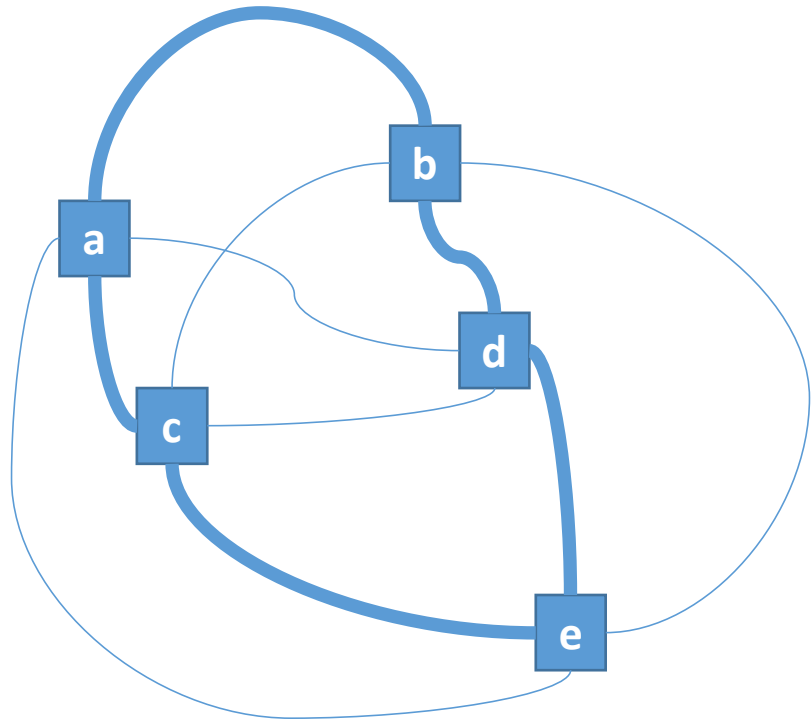
- 連想記憶

- エネルギー極小状態がパターンに相当する
- 入力を与えると、それに近いパターンに収束

巡回セールスマン問題

- X, Y, \dots : 都市
- d_{XY} : 都市 X と Y の間の距離
- すべての都市をちょうど一回ずつ訪れる巡回路で、都市間の距離の合計が最小のもの





a,1	a,2	a,3	a,4	a,5
b,1	b,2	b,3	b,4	b,5
c,1	c,2	c,3	c,4	c,5
d,1	d,2	d,3	d,4	d,5
e,1	e,2	e,3	e,4	e,5

$$(X, i) \neq (Y, j)$$

$$w_{(X,i)(Y,j)} = -A\delta_{XY}(1 - \delta_{ij}) - B\delta_{ij}(1 - \delta_{XY}) - C - Dd_{XY}(\delta_{j i+1} + \delta_{j i-1})$$

$$\theta_{(X,i)} = -C \left(N - \frac{1}{2} \right)$$

同じ都市を2回以上
訪れるか

一度に二つ以上の
都市を訪れるか

連続して訪れる
都市間の距離
(δ の添え字の演
算はmod N)

$$A, B, C, D > 0$$

$x_{(X,i)} = 1$ であるニューロンの数を n とすると

$$C \left(\frac{1}{2} n(n-1) - \left(N - \frac{1}{2} \right) n \right) = \frac{C}{2} n(n-2N) \quad n = N \text{ のとき最小}$$

$$E = -\frac{1}{2} \sum_{i=1}^{i=N} \sum_{j=1}^{j=N} w_{ij} x_i x_j + \sum_{i=1}^{i=N} \theta_i x_i$$

参考：連想記憶

- 連想記憶の場合、ニューロンの値を1と-1とした方が考えやすい
 - 0の代わりに-1を使う
- シナプス荷重 w_{ij} (cf. ヘブ則)

$$w_{ij} = \sum_{p=1}^{p=P} x_i^p x_j^p$$

(x_i^p は、 p 番目のパターンのニューロン i の値)

ただし、 $w_{ii} = 0$

- 閾値 $\theta_i = 0$
- $x_i = x_i^p$ のときに極小となる(傾向)

参考：イジング・モデル（スピングラス）

- 格子点上 i のスピ $\sigma_i = \pm 1$
- エネルギー $E = -\sum_{\langle i,j \rangle} J_{ij} \sigma_i \sigma_j - \sum_i h_i \sigma_i$
 - $\langle i, j \rangle$: 隣り合う格子点
 - J_{ij} : スピン間の相互作用
 - 強磁性体では $J_{ij} = J > 0$
 - 反磁性体では $J_{ij} = J < 0$
 - スピングラスでは J_{ij} はランダム
 - h_i : 外部磁場
- 相転移

ボルツマンマシン

- ホップフィールドネットワークの状態遷移を確率的にしたもの
- 疑似やきなまし (SA---simulated annealing)
 - 極小状態から脱出
- x_i のエネルギーへの寄与を考慮して、 x_i の値を決める
 - $E = - (1/2) \sum_{i=1}^{i=N} \sum_{j=1}^{j=N} w_{ij} x_i x_j + \sum_{i=1}^{i=N} \theta_i x_i$ において、 $x_i = 0$ の場合を基準にとると、 $x_i = 1$ の場合のエネルギーは、ちょうど、 $y_i = \sum_{j=1}^{j=N} w_{ji} x_j - \theta_i$ だけ小さい

ボルツマン分布

- ある分子が、二つの状態をとり、それぞれのエネルギーが E_1 と E_2 であるとき、ボルツマン分布では、それぞれの状態をとる確率を $e^{-E_1/kT} / Z$ と $e^{-E_2/kT} / Z$ とする
- $Z = e^{-E_1/kT} + e^{-E_2/kT}$ は、分配関数

ボルツマンマシン(続き)

- $x_i = 0$ の場合のエネルギーを E_0 とすると、 $x_i = 1$ の場合のエネルギーは、 $E_1 = E_0 - y_i = E_0 - \sum_{j=1}^{j=N} w_{ji}x_j + \theta_i$
- すると、ボルツマン分布にしたがえば、 $x_i = 0$ の場合と $x_i = 1$ の場合の確率は、 $1/(1 + e^{y_i/kT})$ および $e^{y_i/kT}/(1 + e^{y_i/kT})$
- これらの確率にしたがって、 x_i の次の値を定める
- kT は次第に小さくして行く(simulated annealing)
- $kT = 0$ のときは、ホップフィールドネットワークに一致

参考: ボルツマンマシンの学習

- ボルツマンマシンに覚えさせたいパターンの確率分布のもとで、ニューロン i とニューロン j が同時に 1 になる確率を P^+
 - Positive phase
- ボルツマンマシンを自由に走らせたときの平衡状態において、ニューロン i とニューロン j が同時に 1 になる確率を P^-
 - Negative phase
- $P^+ - P^-$ に適当な係数を掛けた値で w_{ij} を増やす

参考：制限付きボルツマンマシン

- RBM---Restricted Boltzmann Machine
- ニューロンは入力ユニットと隠れユニットに分類される
- 入力ユニットの間にシナプス結合はない
- 隠れユニットの間にシナプス結合はない
- 深層学習 (deep learning) において用いられる
 - ボルツマンマシンを重ねる
 - 隠れユニットを次の入力ユニットに
 - 各層のボルツマンマシンを学習によって鍛える

メタヒューリスティクス

- 最適化のための一般化されたヒューリスティクス
- ヒューリスティクスの作り方

- 疑似やきなまし (SA --- simulated annealing)
- タブー探索 (TS --- tabu search)
- 遺伝的アルゴリズム (GA --- genetic algorithm)
- 粒子スウォーム最適化 (PSO --- particle swarm optimization)

- ノーフリーランチ定理

参考文献

- [1] 古川・川上・渡辺・木下・山本・鈴木: メタヒューリスティクスとナチュラルコンピューティング, コロナ社
- [2] David H. Wolpert and William G. Macready: No Free Lunch Theorems for Optimization, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 1, NO. 1, APRIL 1997 67
- [3] 伊庭: 人工知能と人工生命の基礎, オーム社

まず最初に、ノーフリーランチ定理

- X : 探索空間 (巨大だが有限)
- Y : 評価値の空間 (巨大だが有限)
- $F = Y^X$: 評価関数の空間 (巨大だが有限)
- a : アルゴリズム
 - X の要素 n 組 $\{d_1, \dots, d_n\}$ に対して X の要素 d を (確率的に) 返す
 - d は $\{d_1, \dots, d_n\}$ に含まれない
 - a を m 回繰り返すことにより、 $\{d_1, \dots, d_m\}$ が得られる
 - すると、 $e = \{f(d_1), \dots, f(d_m)\}$ が得られる

ノーフリーランチ定理

- 任意のアルゴリズム a_1 と a_2 に対して

$$\sum_f P(e | f, m, a_1) = \sum_f P(e | f, m, a_2)$$

$$\sum_f P(e | f, m, a_1) P(f) = \sum_f P(e | f, m, a_2) P(f)$$

- すべての評価関数を考えると、評価値のばらつきは、アルゴリズムに依らずに一定
 - If an algorithm does particularly well on average for one class of problems then it must do worse on average over the remaining problems. In particular, if an algorithm performs better than random search on some class of problems then it must perform *worse than random search* on the remaining problems. [2]

山登り法 (最大化問題) [1]

- 1) z に初期解を生成しそれを暫定解とする。また、暫定解の評価値 *best* を $f(z)$ とする。
- 2) z の近傍解の集合を $N(z)$ 、 y を空とし、*localbest* を十分小さな値とする。このとき、 $x \in N(z)$ であるすべての x について、もし $f(x) > \textit{localbest}$ なら、 y を x で置き換え、*localbest* を $f(x)$ とする。
- 3) もし $\textit{best} > \textit{localbest}$ なら暫定解 z とその評価値 *best* を出力して終了する。そうでなければ、*best* を *localbest* で、 z を y で置き換え、2)へ戻る。

疑似やきなまし(最大化問題) [1]

- 1) x を探索中の解、 z を暫定解とし x としておく。また、暫定解の評価値 $best$ を $f(z)$ 、 T を温度、 R を反復回数とする。
- 2) 終了条件を満たすまで3)~5)を繰り返す。
- 3) 以下を R 回繰り返す 常に $R=1$ もあり
 - 3-1) $y \in N(x)$ である y をランダムに選ぶ
 - 3-2) $\Delta = f(x) - f(y)$ とする。
 - 3-3) もし $\Delta \leq 0$ ならば探索中の解 x を y で置き換え、3-5)へ
 - 3-4) もし、 $\exp(-\Delta/T) \geq \text{random}(0,1)$ なら探索中の解 x を y で置き換え、3-5)へ
 - 3-5) もし、 $f(x) \geq best$ なら、暫定解 z を x で置き換える。
- 4) T の更新を行う。
- 5) R の更新を行う。
- 6) 暫定解 z とその評価値 $best$ を出力して終了する。

受理率

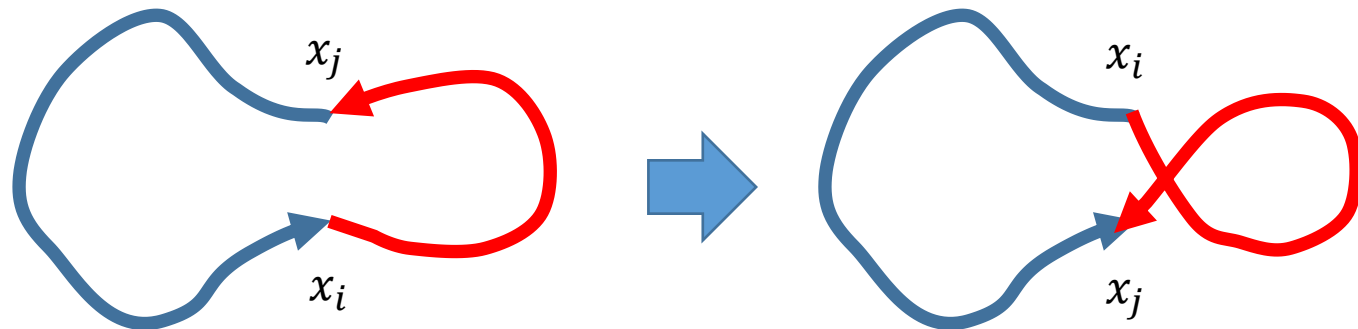
- 解の置き換えを行った割合
- 0.5 付近になるように T と R を調節
 - 問題によっては 0.3~0.5
 - 探索開始時は 0.5~0.8

冷却スケジュール

- T を次第に小さくして行く
- 例: $T = \alpha T$ ($0 < \alpha < 1$)
 - たとえば $\alpha = 0.99$

巡回セールスマン問題の場合

- 探索中の解: 巡回路
- 評価値: 巡回路の総距離
- 巡回路の近傍
 - たとえば、巡回路の二つの都市の入れ換え
 - 2-opt (ツーオプト): 巡回路 x_1, \dots, x_n に対して、 x_i, x_j ($i < j$) を選び、 x_i, x_{i+1}, \dots, x_j の部分を x_j, \dots, x_{i+1}, x_i で置き換える。



参考：量子やきなまし(量子アニーリング)

- 熱揺らぎの代わりに量子揺らぎを利用
- トンネル効果
- SAよりも効率がよい(より短い時間で最適解に収束)

タブー探索 (最大化問題) [1]

- 1) 初期解 $s_0 \in X$ を選び、タブーリスト T を初期化する。
 $s := s_0$, $f^* = f(s_0)$, $s^* = s_0$ とセットする。
通常 s_0 はランダムに選択される。
- 2) 近傍 $N(s)$ の中で、タブーでない、あるいは願望基準 (最良解かどうかの基準) を満たすサブセット $N_s(s)$ をつくる。
 $N_s(s)$ の中で $f(s)$ の最大値を与える解 s を選ぶ。
もし、 $f(s) > f^*$ ならば $f^* := f(s)$, $s^* := s$ と更新を行う。
現在の遷移をタブーとしてタブーリストを更新する。
- 3) 終了条件が満たされれば暫定解を出力して探索は終了。
そうでなければ、2) に戻る。

タブーリストの実装 [1]

- タブーは以前にいたことがある解に探索が戻るサイクリングを防ぐ
- 少し前に行った行動に逆行するような移動を許可しない
- 短期記憶であるタブーリストでは限られた設定情報だけが蓄えられる
- 現在の解に対して直前に行われた数箇所の変更を記録しておき、逆の変換を禁止する
- 長さを固定しておき、それを循環させることによって実装

願望基準

- タブーを解除する基準
- ただし、サイクリングは避ける

GA(遺伝的アルゴリズム) [1]

- 1) 初期個体群の生成(初期化)
- 2) 個体の再生と淘汰
- 3) 交叉、突然変異などの遺伝オペレータ(genetic operation)の適用
- 4) 終了条件を満たせば終了し、満たさなければ2)へ戻り、同じ手順を繰り返す。

個体の再生と淘汰

- ルーレット選択:

個体 j の評価値を f_j としたとき、個体 j を確率 $\frac{f_j}{\sum_i f_i}$ で選択する

($f_i > 0$ と仮定)

エリート保存戦略 [1]

- 集団の中で最も適応度の高い個体を次世代に残す
- エリート個体の遺伝子が集団内に急速に広がる可能性が高いため、局所解 (local minimum) に陥る危険性も存在する

巡回セールスマン問題

- 順序表現

- 都市 a b c d e の場合

- 20200 → caebd

- 21200 → cbead

- 交叉

- 遺伝子座2の場合

- 20200 × 00000 → 20000 × 00200

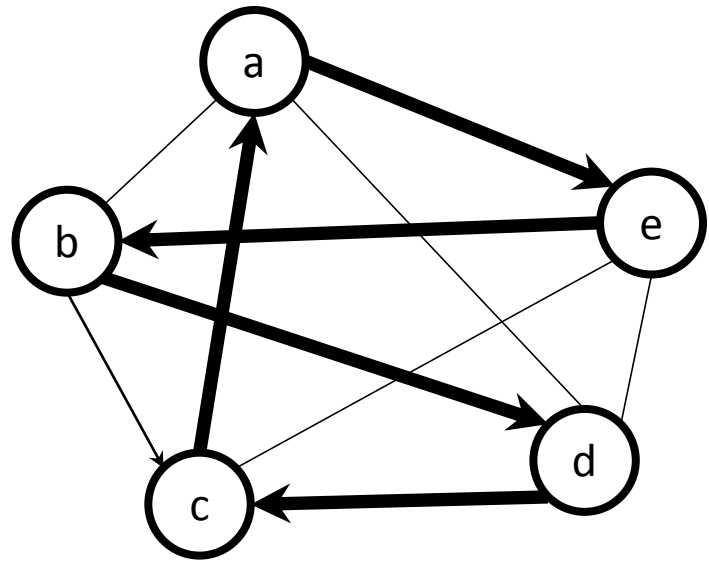
- caebd × abcde → cabde × abecd

- 突然変異

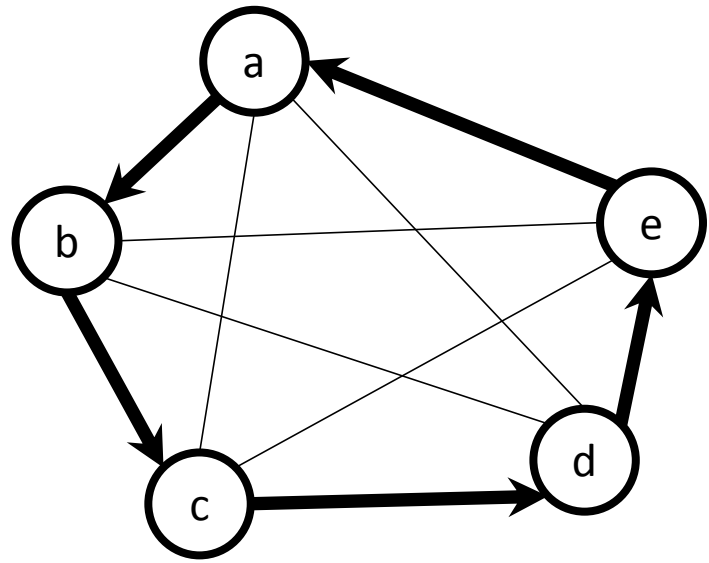
- 遺伝子座1の場合

- 20200 → 21200

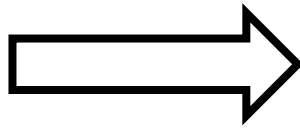
- 逆位



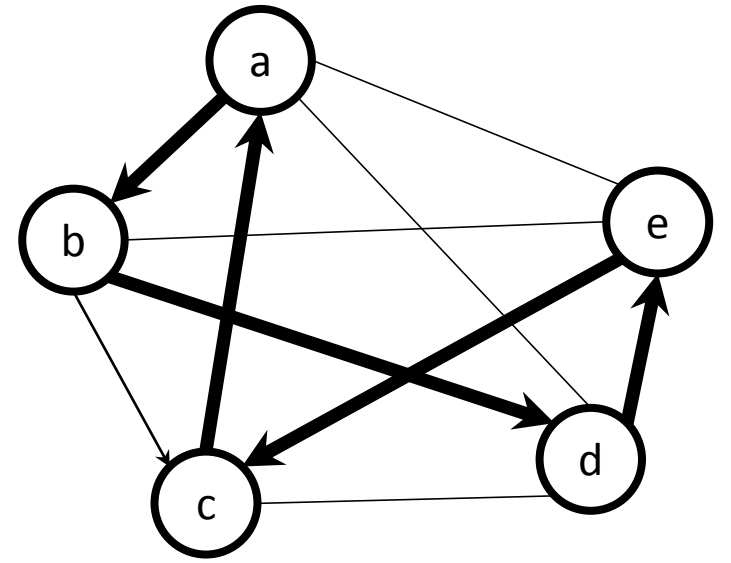
20200



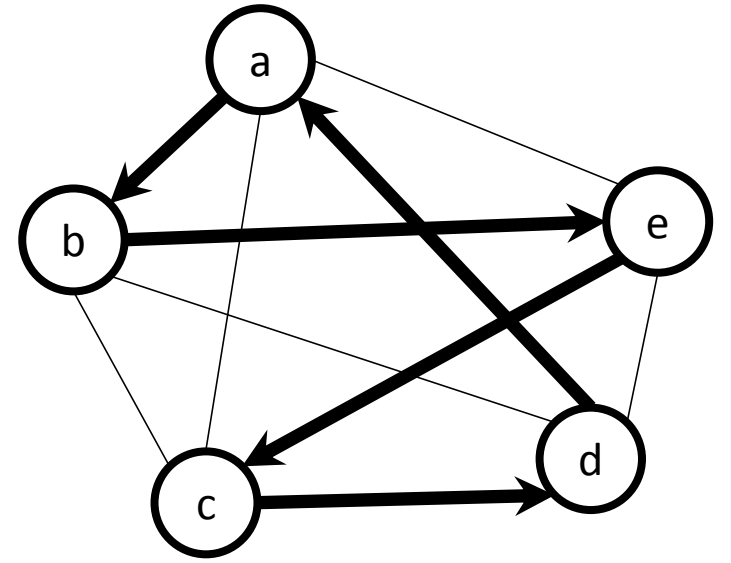
00000



20000



00200



多点探索 [1]

- cf. 単点探索
- GAとPSO
- GAがよい評価値をもつ個体の解表現の一部を用いるのに対して
- PSOでは、よい評価値をもつ解表現すべてを用いてそれらを乱数とともに組み合わせる新しい解を生成する

PSO (粒子スウォーム最適化)

- 多次元空間内の一点を表す位置ベクトルによって解を表現
- 粒子のスウォーム(群れ)を移動させる
- 各粒子は速度ベクトルをもつ
- 各粒子の速度ベクトルは、その粒子のこれまでに発見した最良解と、全粒子がこれまでに発見した最良解を加味して、更新
- 実数値の最適化に限るが、調整すべきパラメータが比較的少ないので、使いやすい

PSO (最大化問題) [1]

- 1) T を最大反復回数、 $x^k(t)$ と $v^k(t)$ をそれぞれ、 t 回目の反復における粒子 k の位置ベクトルと速度ベクトルとする。
- 2) $t = 0$ とし、全粒子がこれまでに発見した暫定解 $g = (g_1, \dots, g_n)$ を空、その評価値 $f(g)$ を十分小さい値とする。
- 3) $k = 1, \dots, N$ のすべての k について、 $x^k(0)$ と $v^k(0)$ をランダムに初期化する。また、粒子 k がこれまでに発見した最良解 $p_k = (p_1^k, \dots, p_n^k)$ を空とし、その評価値 $f(p_k)$ を十分小さい値とする。
- 4) $k = 1, \dots, N$ のすべての粒子 k について以下の4-1)~4-5)を繰り返す。
- 5) $t = t + 1$ として、もし $t > T$ なら、暫定解 g とその評価値 $f(g)$ を出力して終了する。そうでなければ、4)に戻る。

PSO (反復部分) [1]

- 4-1) $i = 1, \dots, n$ のすべてについて、 $x_i^k(t+1) = x_i^k(t) + v_i^k(t)$ を求める。
- 4-2) $x^k(t+1)$ の評価値 $f(x^k(t+1))$ を計算する。
- 4-3) もし $f(x^k(t+1)) > f(p^k)$ であれば、
 p^k を $x^k(t+1)$ で置き換え、 $f(p^k)$ を $f(x^k(t+1))$ とする。
- 4-4) もし $f(x^k(t+1)) > f(g)$ であれば、
 g を $x^k(t+1)$ で置き換え、 $f(g)$ を $f(x^k(t+1))$ とする。
- 4-5) $i = 1, \dots, n$ のすべてについて、 $v_i^k(t+1)$ を
 $wv_i^k(t) + r_{1i}c_1(p_i^k - x_i^k(t+1)) + r_{2i}c_2(g_i - x_i^k(t+1))$ で置き換える。
ただし、 w, c_1, c_2 は定数、 r_{1i}, r_{2i} は区間 $[0,1]$ の一様乱数である。

多目的最適化

- 複数の目的関数(評価関数)を最適化
- 目的関数間にトレードオフ
- パレート最適
 - どの目的関数も、その値を上げるには、他のいずれかの目的関数の値が下がってしまう状況
- 一般に、多目的最適化には多点探索が効果的
 - パレート最適なものの中から選択