

カオスと組み合わせ最適化

非線形システム特論

池口 徹

埼玉大学 大学院 理工学研究科 情報数理科学専攻

338-8570 さいたま市 桜区 下大久保 255

Tel : 048-858-3577, Fax : 048-858-3716

Email : tohru@ics.saitama-u.ac.jp

URL : <http://www.nls.ics.saitama-u.ac.jp/~tohru>

はじめに

組み合わせ最適化問題

1. スケジューリング, 基盤配線, 配送計画
多様な応用分野を有する
2. NP-困難 な組み合わせ最適化問題
 - ◀ 巡回セールスマン問題
 - ◀ 2次割当問題

厳密な最適解を求めることは非常に難しい



出来るだけ良好な解を, 出来るだけ短時間 → 近似解法

カオスダイナミクスを用いた
組み合わせ最適化問題へのアプローチ

組み合わせ最適化問題の解法

◀ 厳密解法

膨大な計算量を要する．実際の問題には適用困難．

組み合わせ最適化問題の解法

◀ 厳密解法

膨大な計算量を要する．実際の問題には適用困難．

◀ 近似解法

組み合わせ最適化問題の解法

- ◀ 厳密解法
膨大な計算量を要する．実際の問題には適用困難．
- ◀ 近似解法
 - 構成法
Nearest Neighbor 法など

組み合わせ最適化問題の解法

◀ 厳密解法

膨大な計算量を要する．実際の問題には適用困難．

◀ 近似解法

□ 構成法

Nearest Neighbor 法など

□ 逐次的改善法

組み合わせ最適化問題の解法

◀ 厳密解法

膨大な計算量を要する．実際の問題には適用困難．

◀ 近似解法

□ 構成法

Nearest Neighbor 法など

□ 逐次的改善法

◆ 最急降下法 ⇒ ローカルミニマム問題

組み合わせ最適化問題の解法

◀ 厳密解法

膨大な計算量を要する．実際の問題には適用困難．

◀ 近似解法

□ 構成法

Nearest Neighbor 法など

□ 逐次的改善法

◆ 最急降下法 ⇒ ローカルミニマム問題

◆ ローカルミニマムからの脱出

SA, GA, タブーサーチ

カオスダイナミクス

⇒ カオスをノイズ源として用いるのではない

⇒ 過去の探索履歴情報 + 忘却効果が，結果としてカオスダイナミクスを生み出している

相互結合型ニューラルネットを用いた手法

相互結合型ニューラルネットの山下りダイナミクス
(Hopfield – Tank ニューラルネットワーク, 1980)

- ◀ Good news
 - 問題が与えられるとシナプス結合係数が決まる
 - ⇔ ニューラルネットワークのダイナミクスに従って解(もしかすると最適?) を得ることが出来る

相互結合型ニューラルネットを用いた手法

相互結合型ニューラルネットの山下りダイナミクス
(Hopfield – Tank ニューラルネットワーク, 1980)

◀ Good news

問題が与えられるとシナプス結合係数が決まる

⇔ ニューラルネットワークのダイナミクスに従って解
(もしかすると最適?) を得ることが出来る

◀ Bad news

1. 大きい問題への適用困難

サイズ $N \rightarrow \begin{cases} \text{ニューロン } N^2 \text{ 個} \\ \text{シナプス結合係数 } N^4 \text{ 本} \end{cases}$

2. 実行可能解が出にくい

3. ローカルミニマム問題

従来のカオスを用いた探索法

アイディア

1. ホップフィールドニューラルネットワークを基盤
2. ローカルミニマムをカオスダイナミクスを用いて脱出

文献

- ◀ Nozawa: Chaos, Vol.2, No.3, pp.377–386, 1992 等
カオスニューラルネットワークを用いた TSP の解法 .
でも , 10 都市 .
- ◀ Yamada & Aihara: J. Intelligent and Fuzzy System, Vol.5,
No.1, pp.53–68, 1997.
どのようなカオスが性能が良いか? \Leftrightarrow edge of chaos
- ◀ Chen & Aihara: Neural Networks, Vol.8, No.6,
pp.915–930, 1995 等
カオスアニーリング , 最適解の埋め込み証明

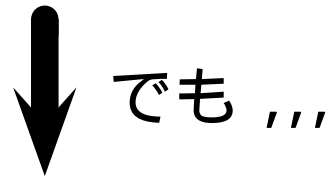
従来のカオスを用いた探索法

アイディア

1. ホップフィールドニューラルネットワークを基盤
2. ローカルミニマムをカオスダイナミクスを用いて脱出

問題点

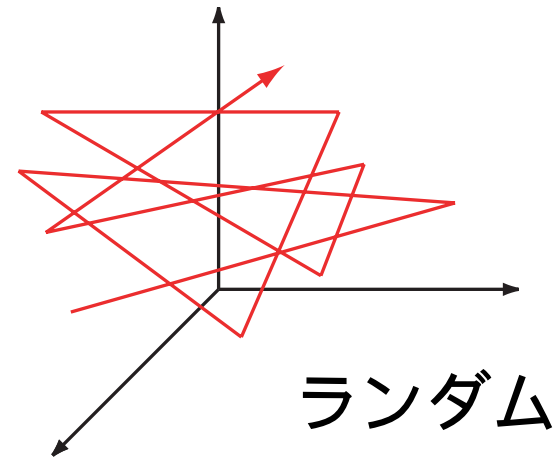
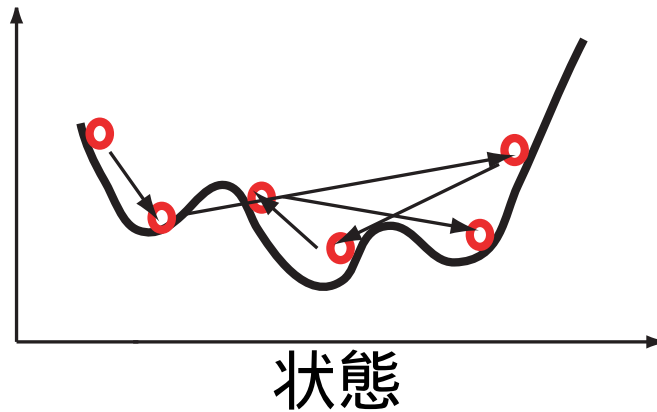
- ◀ 大きい問題への適用は依然困難 .
10 都市 (Nozawa, Yamada) ~ 48 都市 (Chen)
- ◀ 実行可能解が出にくい
∴ $N \times N$ 個のニューロンの発火パターンで巡回路を表現



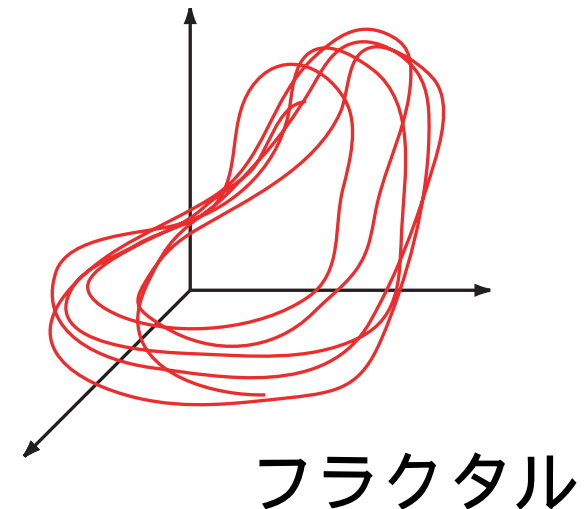
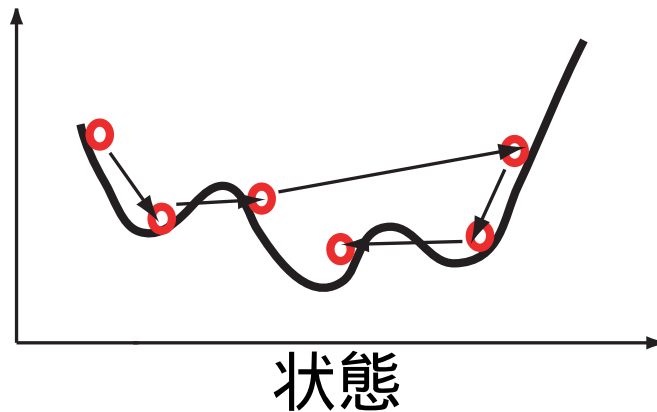
ローカルミニマム問題を解決するには、
確率的な手法によりも効率的？

カオスの探索と確率的探索

◀ 確率的なダイナミクスによるローカルミニマムの回避



◀ カオスダイナミクスを用いたローカルミニマムの回避



カオスダイナミクスを創り出す

不応性を有するニューロンモデル

発火した直後は、発火しにくくなる性質

◀ 南雲-佐藤ニューロンモデル (Nagumo & Sato, 1972)

$$x(t+1) = u \left[A(t) - \alpha \sum_{d=0}^t k^d x(t-d) - \theta \right]$$

k : 不応性の減衰定数

$\alpha(>0)$: 不応性の強度

$A(t)$: ニューロンへの入力

θ : 閾値

$u(\cdot)$: 出力 (活性化) 関数 \Rightarrow ステップ関数

南雲-佐藤ニューロンの不応性

$$x(t+1) = u \left[A(t) - \alpha \sum_{d=0}^t k^d x(t-d) - \theta \right] = u[y(t+1)]$$

$$\begin{aligned} y(t+1) &= A(t) - \alpha \{ k^0 x(t) + k^1 x(t-1) + \cdots + k^{t-1} x(1) + k^t x(0) \} - \theta \\ &= A(t) - \alpha \{ x(t) + k^1 x(t-1) + \cdots + k^{t-1} x(1) + k^t x(0) \} - \theta \end{aligned}$$

南雲-佐藤ニューロンの不応性

$$x(t+1) = u \left[A(t) - \alpha \sum_{d=0}^t k^d x(t-d) - \theta \right] = u[y(t+1)]$$

$$\begin{aligned} y(t+1) &= A(t) - \alpha \{ k^0 x(t) + k^1 x(t-1) + \cdots + k^{t-1} x(1) + k^t x(0) \} - \theta \\ &= A(t) - \alpha \{ x(t) + k^1 x(t-1) + \cdots + k^{t-1} x(1) + k^t x(0) \} - \theta \end{aligned}$$



過去の発火情報が次時刻の発火に影響を与える

南雲-佐藤ニューロンの不応性

$$x(t+1) = u \left[A(t) - \alpha \sum_{d=0}^t k^d x(t-d) - \theta \right] = u[y(t+1)]$$

$$\begin{aligned} y(t+1) &= A(t) - \alpha \{ k^0 x(t) + k^1 x(t-1) + \cdots + k^{t-1} x(1) + k^t x(0) \} - \theta \\ &= A(t) - \alpha \{ x(t) + k^1 x(t-1) + \cdots + k^{t-1} x(1) + k^t x(0) \} - \theta \end{aligned}$$



過去の発火情報が次時刻の発火に影響を与える



$0 < k < 1, \alpha > 0$ とすれば,

過去に発火すれば, 次時刻は発火しにくい \iff 不応性

南雲-佐藤ニューロンと1次元写像

$$x(t+1) = u \left[A(t) - \alpha \sum_{d=0}^t k^d x(t-d) - \theta \right]$$

において,

$$y(t+1) = A(t) - \alpha \sum_{d=0}^t k^d x(t-d) - \theta$$

とおくと,

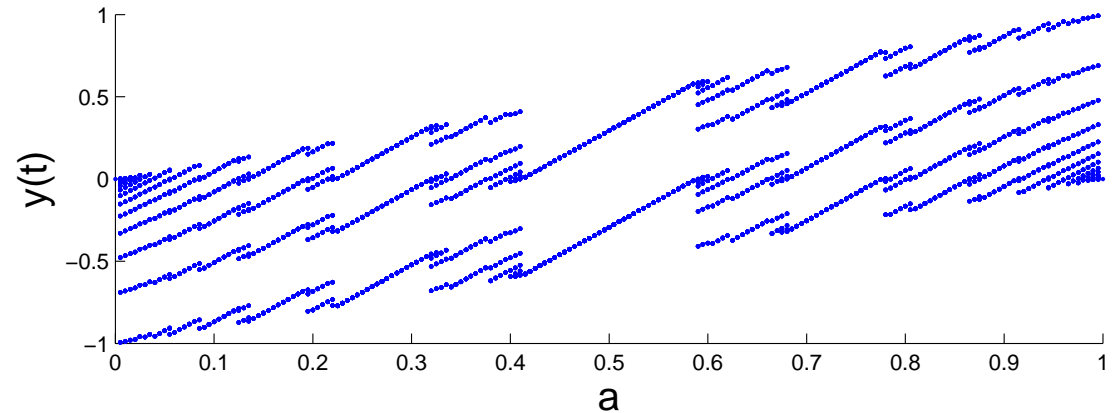
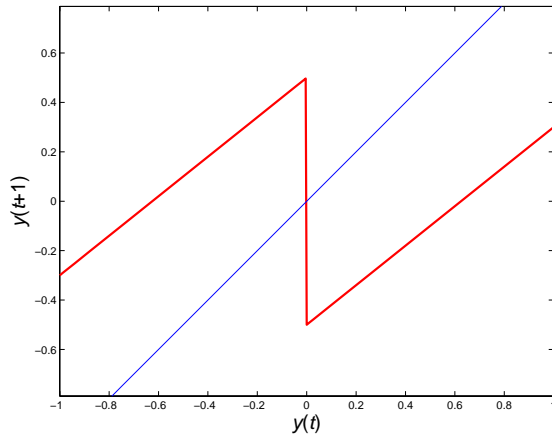
$$y(t+1) = ky(t) - \alpha u[y(t)] + (A(t) - \theta)(1 - k)$$

となる. $A(t) = A(\text{一定})$ とすると $a = (A - \theta)(1 - k)$

$$y(t+1) = ky(t) - \alpha u[y(t)] + a \quad \text{1次元写像}$$

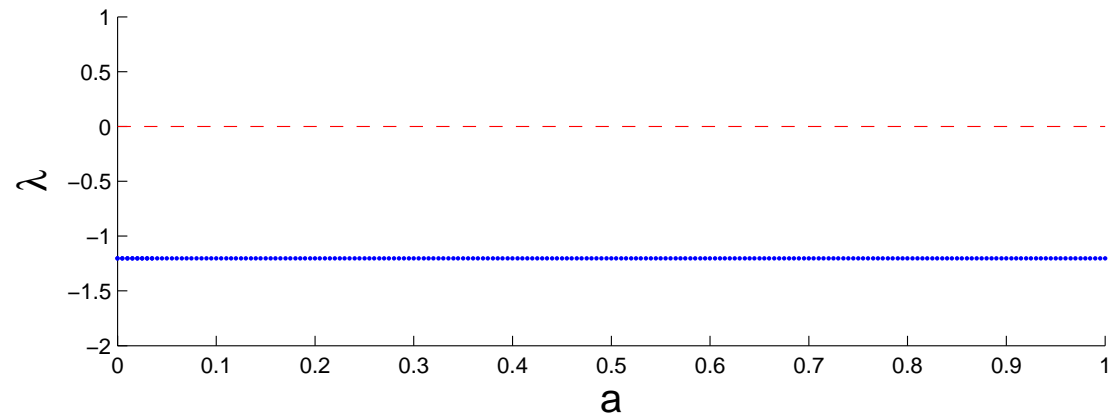
南雲-佐藤ニューロンの応答特性

$$y(t+1) = ky(t) - \alpha u[y(t)] + a = F(y(t)), t \in \mathbb{Z}$$



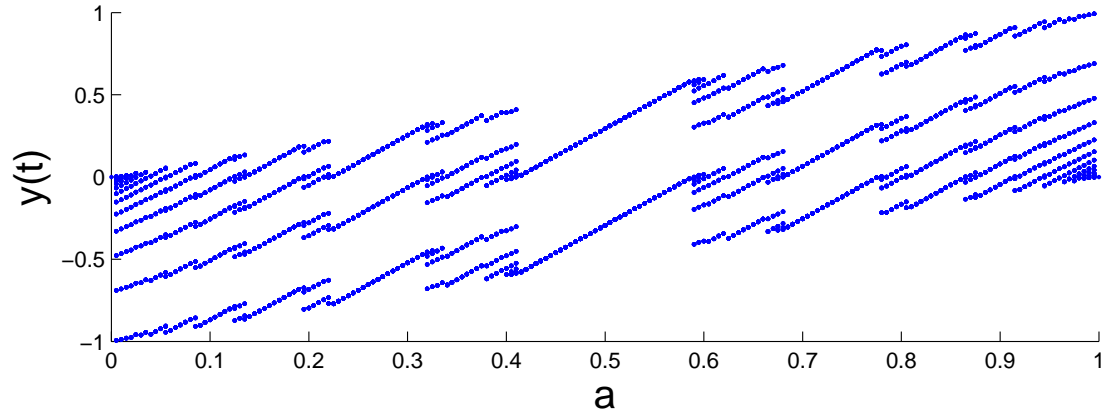
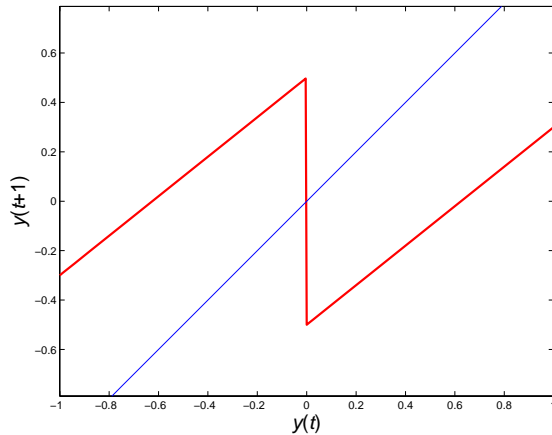
リアプノフ指数

$$\lambda = \lim_{L \rightarrow \infty} \frac{1}{L} \sum_{t=0}^{L-1} \log \left| \frac{dF(y(t))}{dy(t)} \right|$$



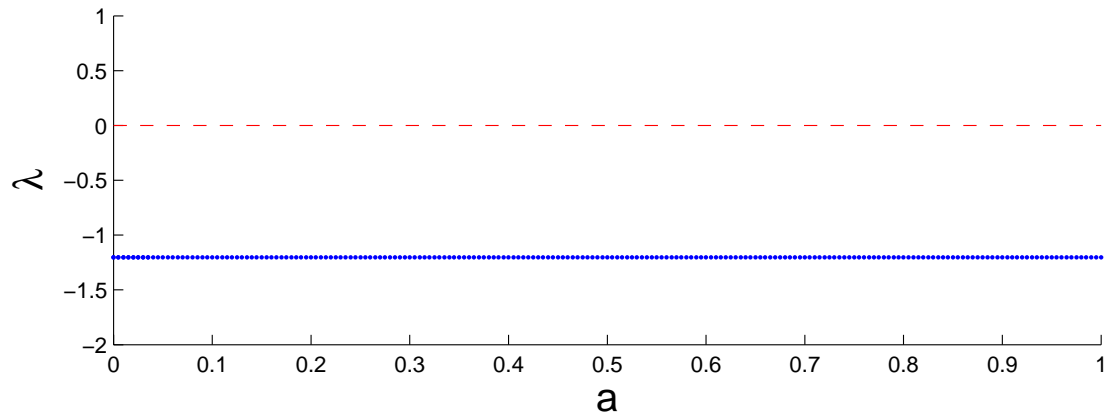
南雲-佐藤ニューロンの応答特性

$$y(t+1) = ky(t) - \alpha u[y(t)] + a = F(y(t)), t \in \mathbb{Z}$$



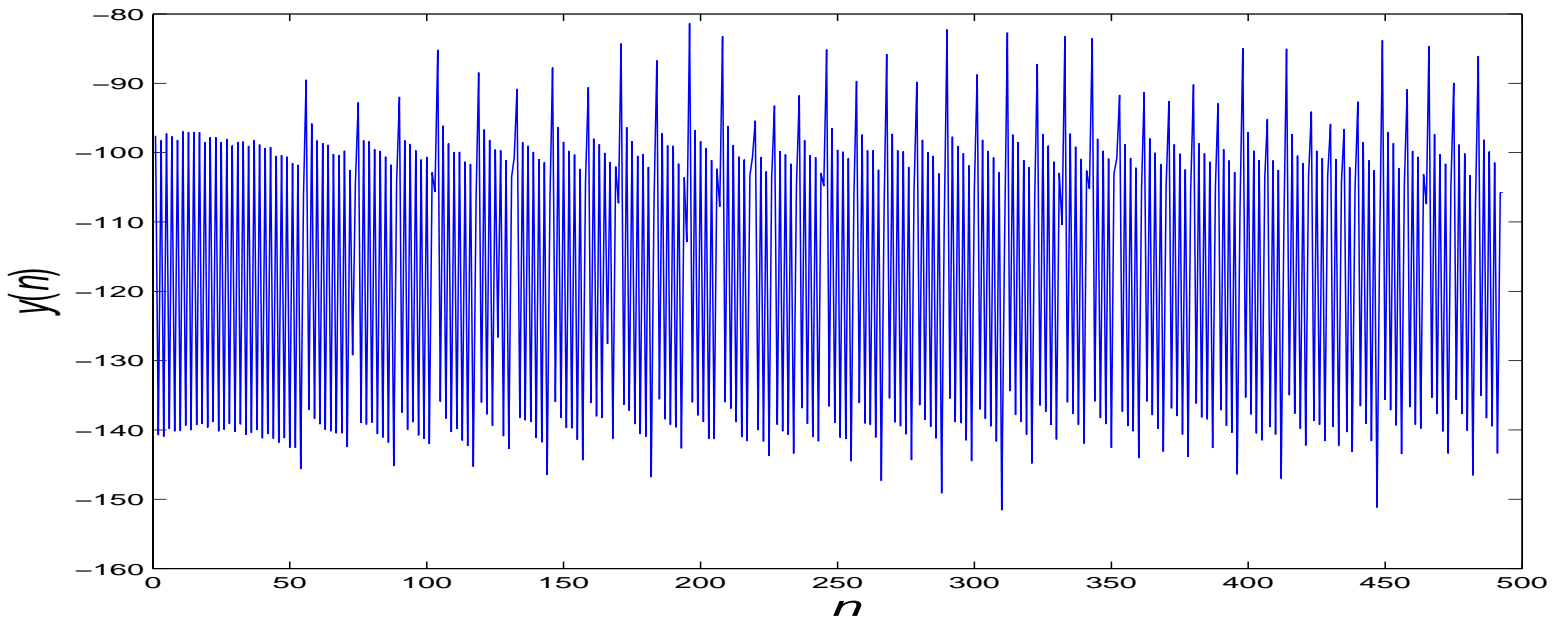
リアプノフ指数

$$\lambda = \lim_{L \rightarrow \infty} \frac{1}{L} \sum_{t=0}^{L-1} \log \left| \frac{dF(y(t))}{dy(t)} \right|$$



南雲-佐藤ニューロンモデルではカオス応答を再現できない

ヤリイカ巨大軸索におけるカオス



- ◀ 南雲-佐藤ニューロンモデルではカオス応答を再現できない
- ◀ 実際のニューロンでは，カオス応答が観測されている．

カオスダイナミクスを創り出す

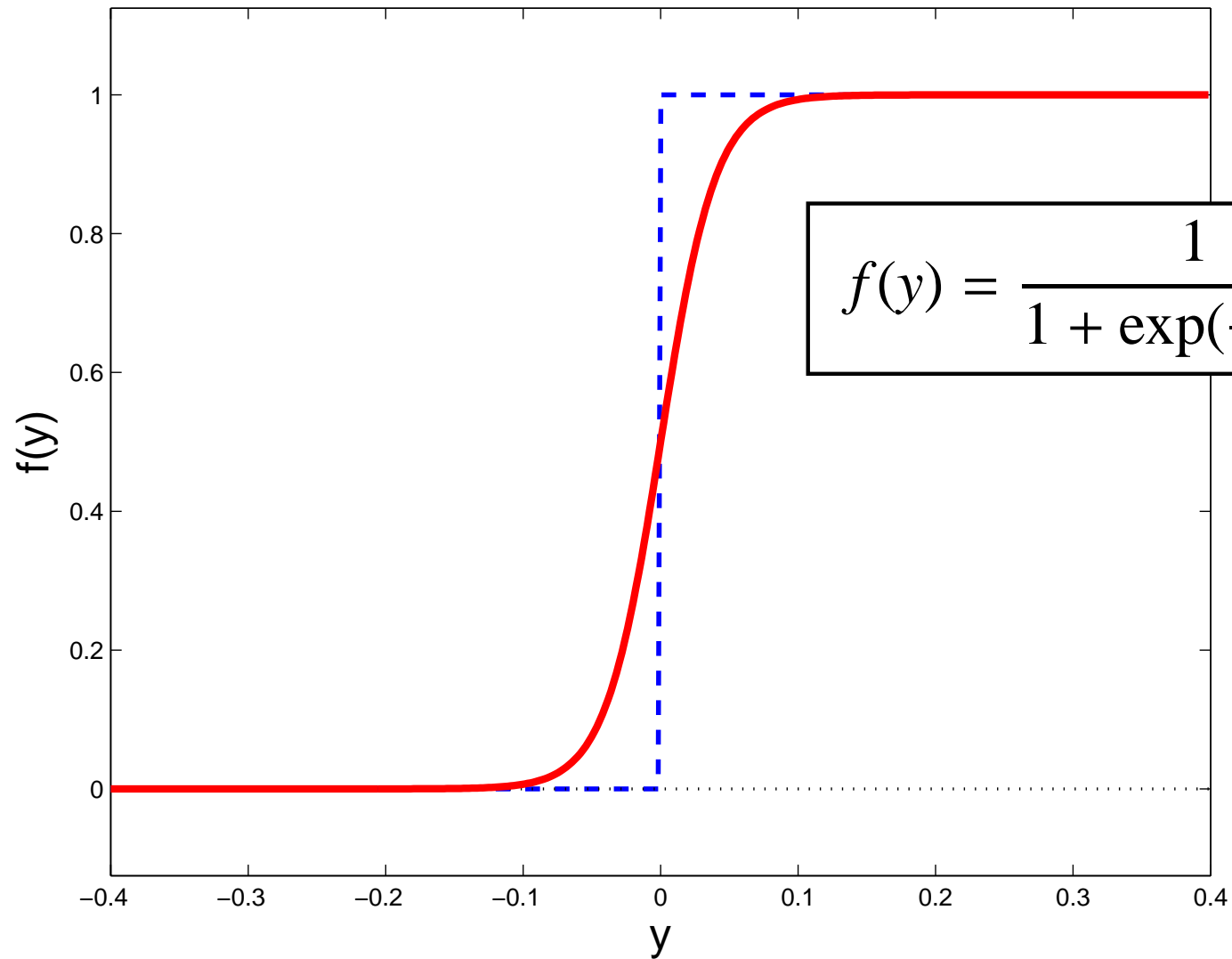
不応性を有するニューロンモデル

- ◀ 南雲-佐藤ニューロンモデル \Rightarrow カオス応答再現不能
- ◀ カオスニューロン (Aihara, Takabe, Toyoda, 1990)

$$x(t+1) = f \left[A(t) - \alpha \sum_{d=0}^t k^d g\{x(t-d)\} - \theta \right]$$

- k : 不応性の減衰定数
- $\alpha(>0)$: 不応性の強度
- $A(t)$: ニューロンへの入力
- θ : 閾値
- $f(\cdot)$: 出力 (活性化) 関数 \Rightarrow シグモイド関数

ステップ関数とシグモイド関数



カオスニューロンと1次元写像

$$x(t+1) = f \left[A(t) - \alpha \sum_{d=0}^t k^d x(t-d) - \theta \right]$$

において ,

$$y(t+1) = A(t) - \alpha \sum_{d=0}^t k^d x(t-d) - \theta$$

とおくと ,

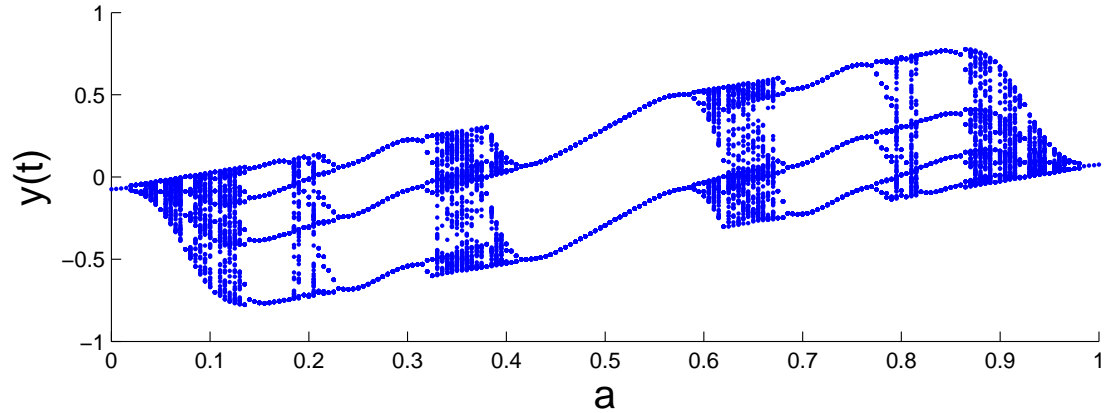
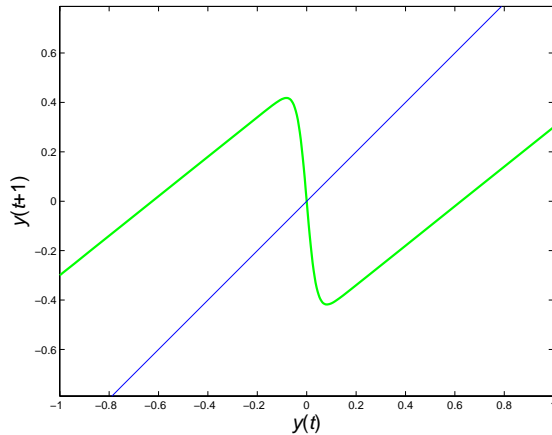
$$y(t+1) = ky(t) - \alpha u[y(t)] + (A(t) - \theta)(1 - k)$$

となる . $A(t) = A(\text{一定})$ とすると $a = (A - \theta)(1 - k)$

$$y(t+1) = ky(t) - \alpha f[y(t)] + a \quad \text{1次元写像}$$

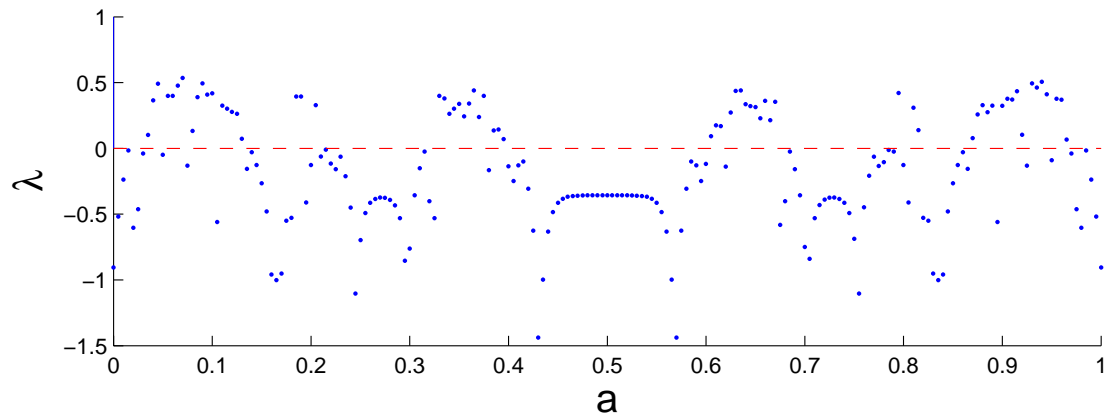
カオスニューロンの応答特性

$$y(t+1) = ky(t) - \alpha f[y(t)] + a = F(y(t)), t \in \mathbb{Z}$$

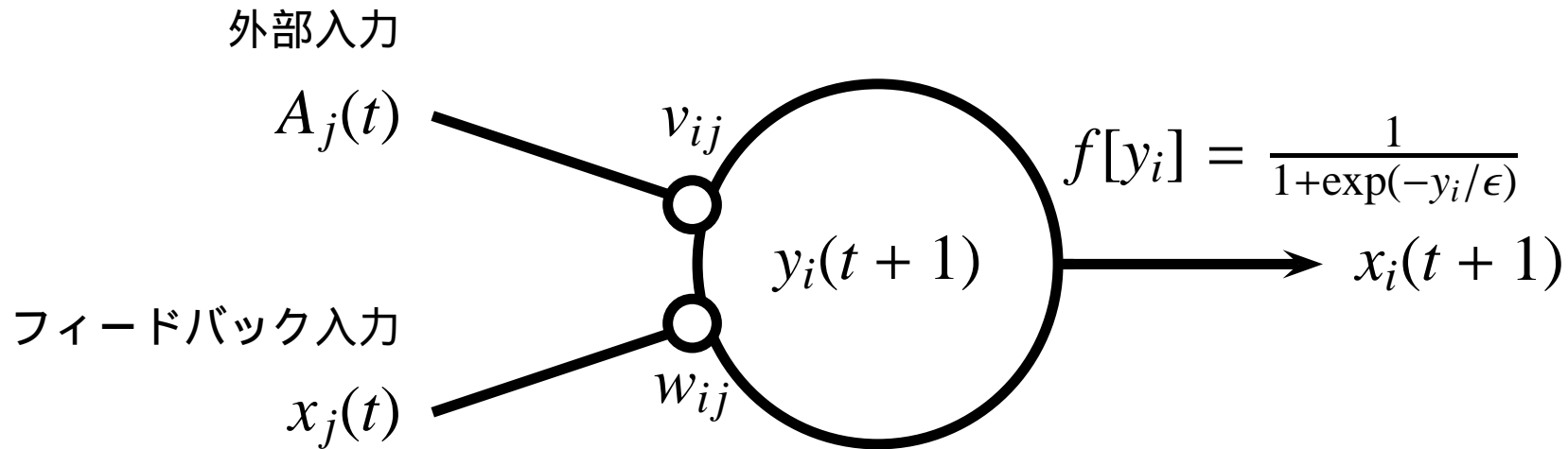


リアプノフ指数

$$\lambda = \lim_{L \rightarrow \infty} \frac{1}{L} \sum_{t=0}^{L-1} \log \left| \frac{dF(y(t))}{dy(t)} \right|$$



カオスニューラルネットワーク



$$\begin{aligned}
 x_i(t+1) = & f \left[\sum_j^M v_{ij} \sum_{d=0}^t k_s^d A_j(t-d) \right. \\
 & \left. + \sum_j^N w_{ij} \sum_{d=0}^t k_m^d x_j(t-d) - \alpha \sum_{d=0}^t k_r^d x_i(t-d) - \theta_i \right]
 \end{aligned}$$

カオスニューラルネットワーク

$$\left\{ \begin{array}{l} \xi_i(t+1) = \sum_{j=1}^M v_{ij} \sum_{d=0}^t k_s^d A_j(t-d) \\ \eta_i(t+1) = \sum_{j=1}^N w_{ij} \sum_{d=0}^t k_m^d x_j(t-d) \\ \zeta_i(t+1) = -\alpha \sum_{d=0}^t k_r^d x_i(t-d) - \theta_i \end{array} \right.$$

$$\Rightarrow \left\{ \begin{array}{l} \xi_i(t+1) = k_s \xi_i(t) + \sum_{j=1}^M v_{ij} A_j(t) \\ \eta_i(t+1) = k_m \eta_i(t) + \sum_{j=1}^N w_{ij} x_j(t) \\ \zeta_i(t+1) = k_r \zeta_i(t) - \alpha x_i(t) - \theta_i \end{array} \right.$$

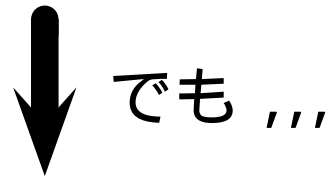
従来のカオスを用いた探索法

アイデア

1. ホップフィールドニューラルネットワークを基盤
2. ローカルミニマムをカオスダイナミクスを用いて脱出

問題点

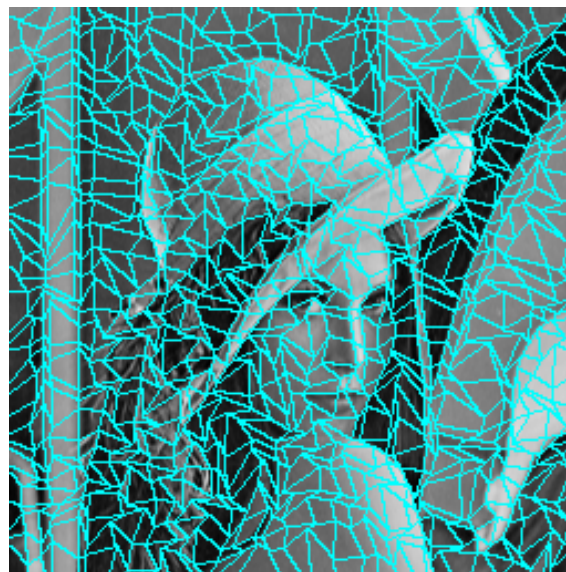
- ◀ 大きい問題への適用は依然困難 .
10 都市 (Nozawa, Yamada) ~ 48 都市 (Chen)
- ◀ 実行可能解が出にくい
∴ $N \times N$ 個のニューロンの発火パターンで巡回路を表現



ローカルミニマム問題を解決するには、
確率的な手法によりも効率的？

画像の領域分割

◀ 可変形状ブロック法 (松田, 伊東, 宇都宮, 1994)



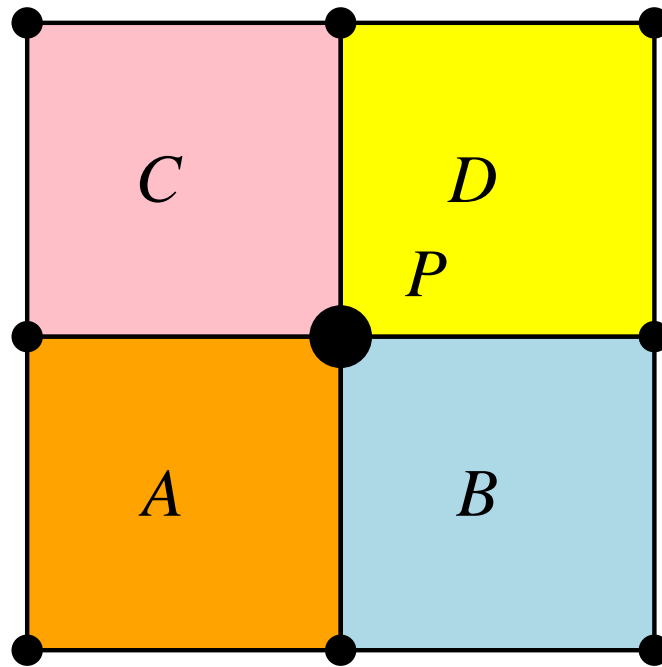
◀ 画像解析, 画像圧縮に応用可能な領域分割法

◀ ローカルミニマム問題

→ カオスダイナミクスによる性能改善が可能では?

画像の領域分割

◀ 可変形状ブロック法 (松田, 伊東, 宇都宮, 1994)



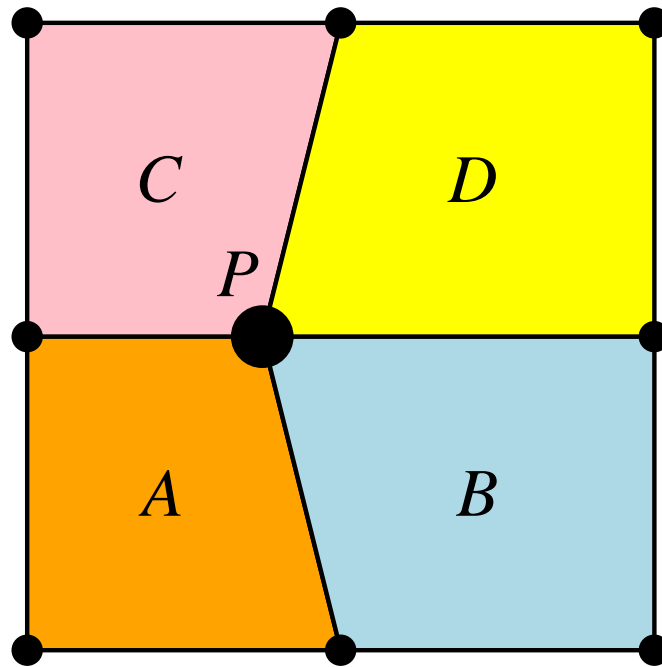
◀ 画像解析, 画像圧縮に応用可能な領域分割法

◀ ローカルミニマム問題

→ カオスダイナミクスによる性能改善が可能では?

画像の領域分割

◀ 可変形状ブロック法 (松田, 伊東, 宇都宮, 1994)



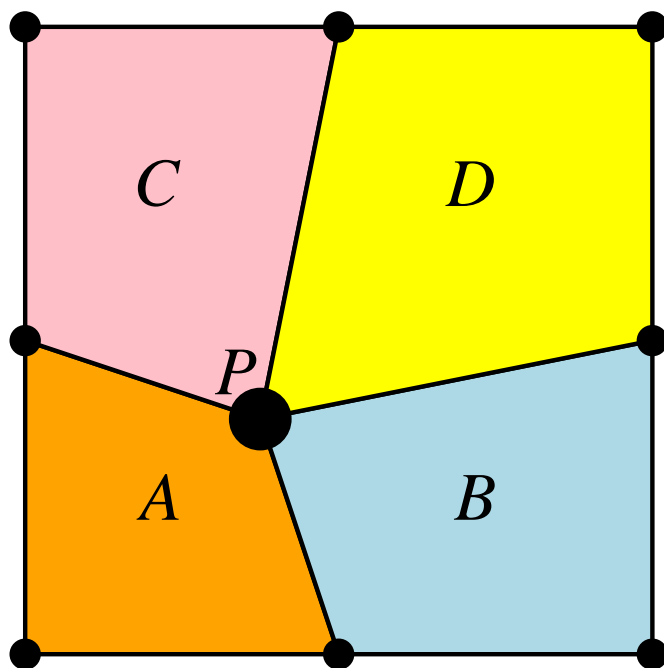
◀ 画像解析, 画像圧縮に応用可能な領域分割法

◀ ローカルミニマム問題

→ カオスダイナミクスによる性能改善が可能では?

画像の領域分割

◀ 可変形状ブロック法 (松田, 伊東, 宇都宮, 1994)



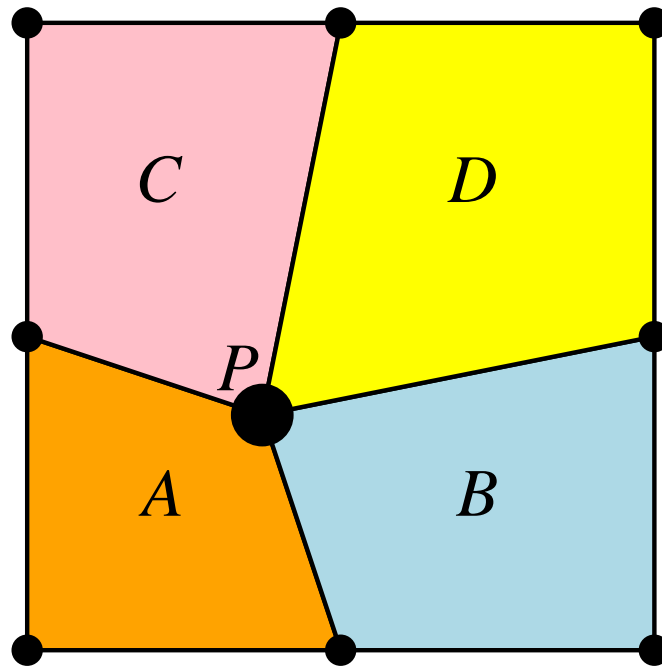
◀ 画像解析, 画像圧縮に応用可能な領域分割法

◀ ローカルミニマム問題

→ カオスダイナミクスによる性能改善が可能では?

画像の領域分割

◀ 可変形状ブロック法 (松田, 伊東, 宇都宮, 1994)



$$E_P = S_A + S_B + S_C + S_D$$

S_A : 領域 A 分散

S_A : 領域 B 分散

S_A : 領域 C 分散

S_A : 領域 D 分散

◀ 画像解析, 画像圧縮に応用可能な領域分割法

◀ ローカルミニマム問題

→ カオスダイナミクスによる性能改善が可能では?

可変形状ブロック法のための CNN

$$\xi_{P_i}(t+1) = k_s \xi_{P_i}(t) + h(E_0 - E_i)$$

$$\zeta_P(t+1) = k_r \zeta_P(t) - \alpha \left[\sum_{j=1}^4 M_{P_j}(t) \right] + a_P$$

$$M_{P_i}(t+1) = f[\xi_{P_i}(t+1) + \zeta_P(t+1)]$$

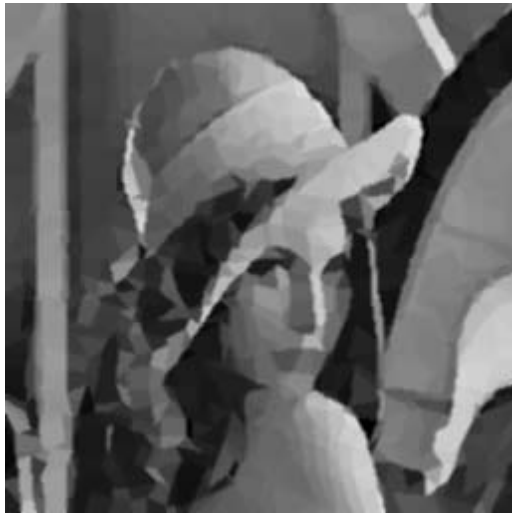
- ◀ $P_i (i = 1, 2, 3, 4)$ 頂点 P 四方向
- ◀ E_0 : 頂点 P に関する現在の分散和
- ◀ E_i : 頂点 P を i 方向に移動させた場合の分散和

$M_{P_i}(t+1) > 0.5$ で最大の i 方向に頂点 P を移動

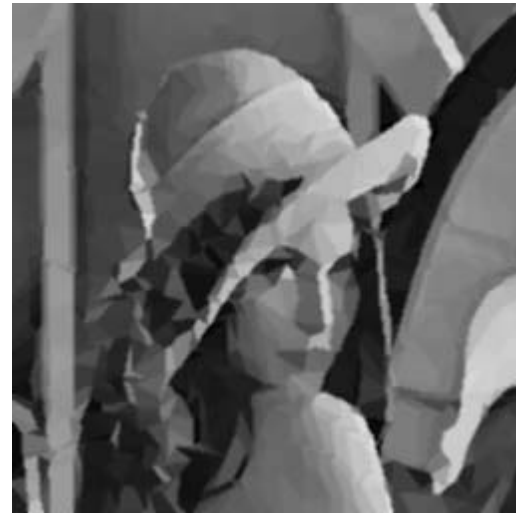
結果



原画像

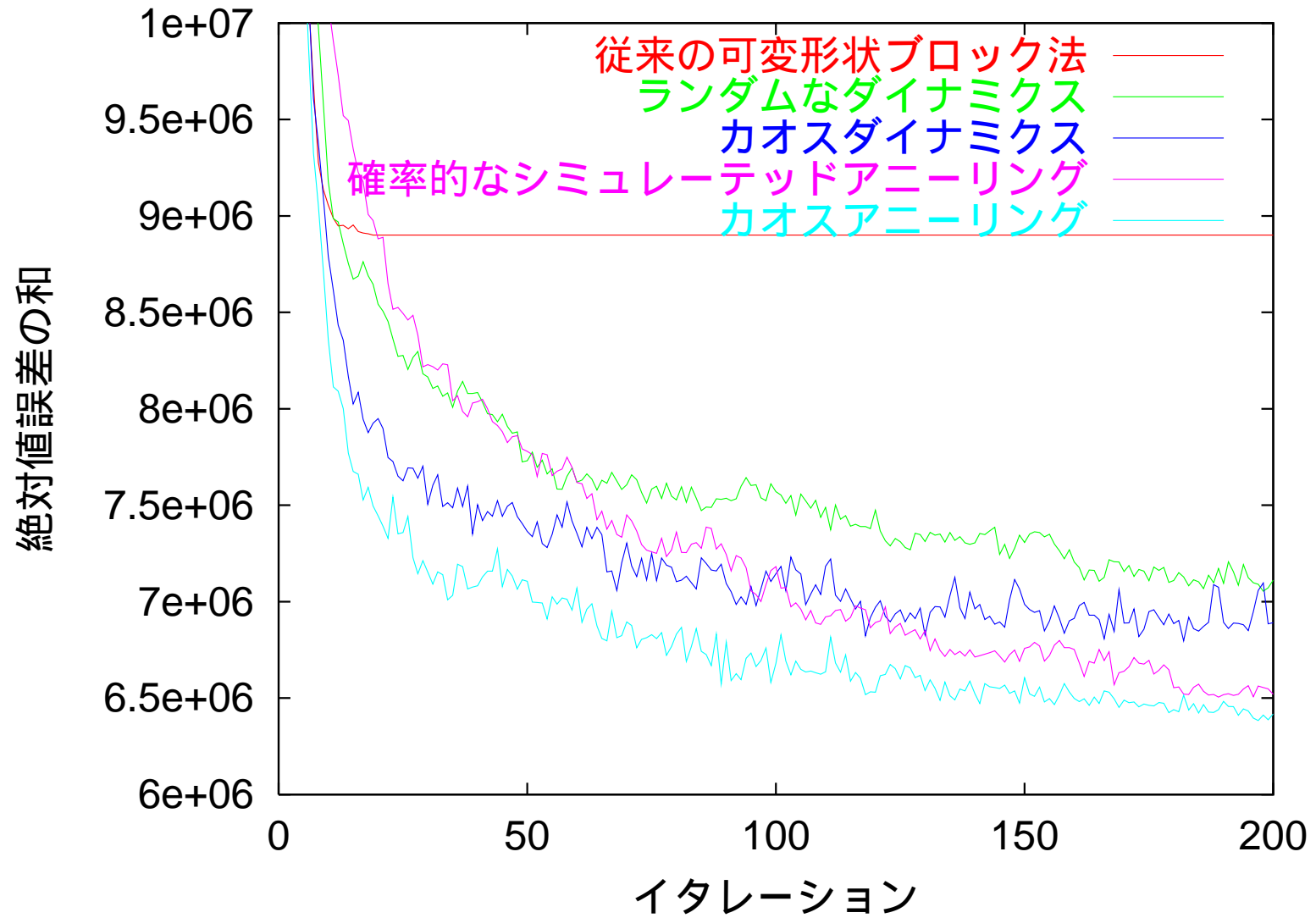


最急降下法 (25.98dB)

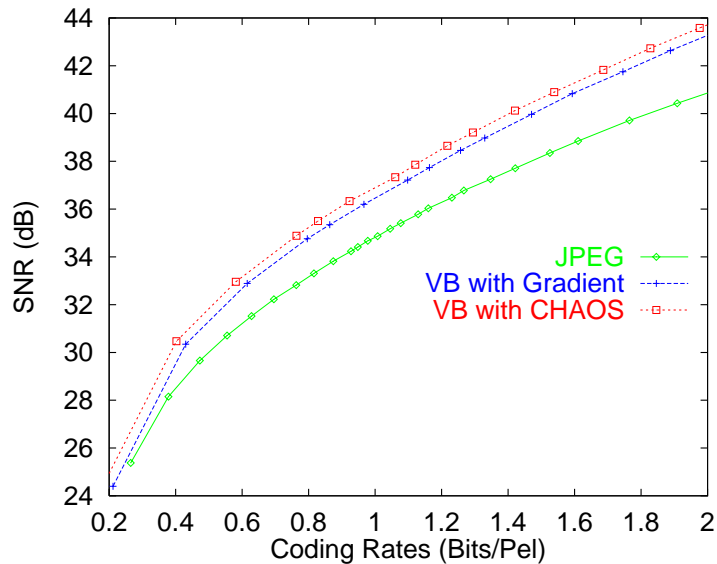


カオス (26.39dB)

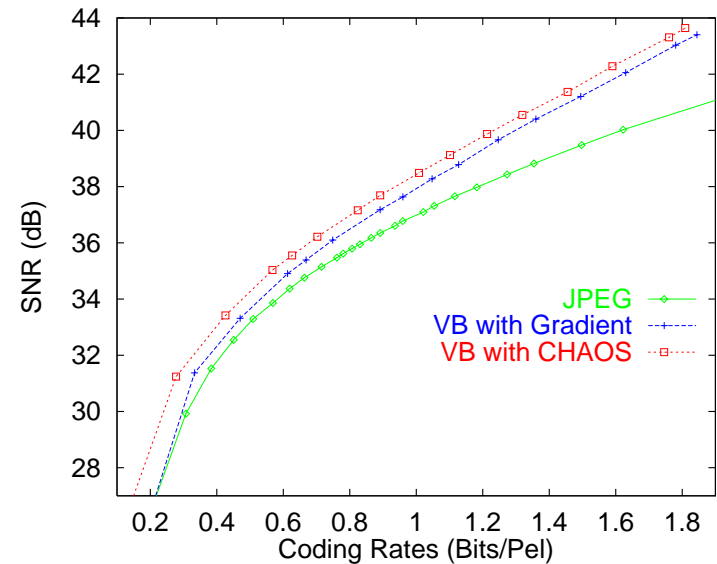
性能比較



レート歪特性



LENNA



COUPLE

カオスダイナミクスの導入による
性能改善が最も良好な圧縮率

カオス探索法とヒューリスティック解法


Hopfield ニューラルネット
+
カオスダイナミクス

- ◀ Good News
高い解法能力
実行可能解が出にくい
- ◀ Bad News
大規模問題適用困難

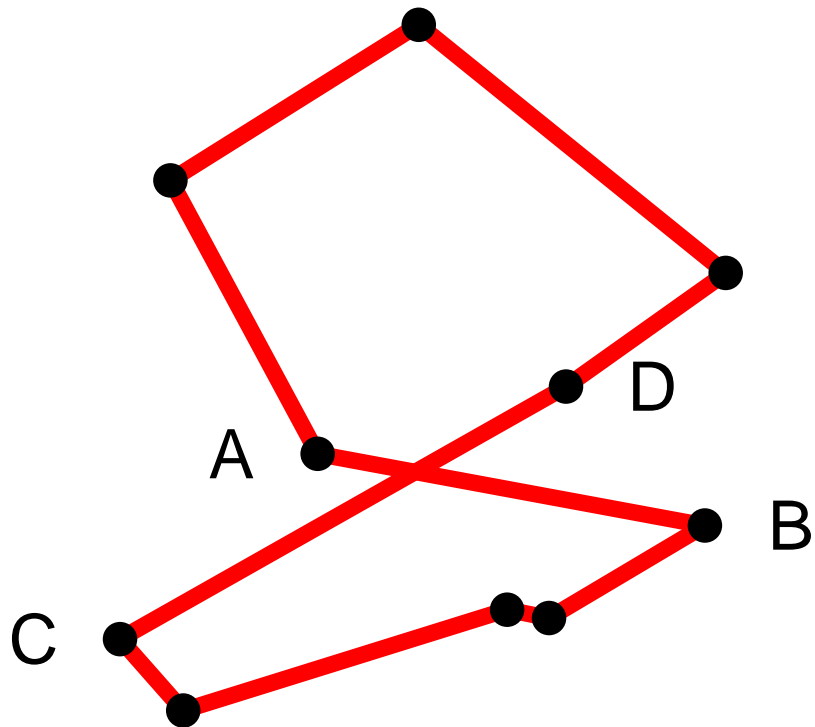
ヒューリスティック解法
例えば
2-opt, 3-opt, LK

- ◀ Good News
大規模問題適用可能
常に実行可能解
- ◀ Bad News
ローカルミニマム問題

カオスダイナミクス
+
ヒューリスティック解法 (2-opt)

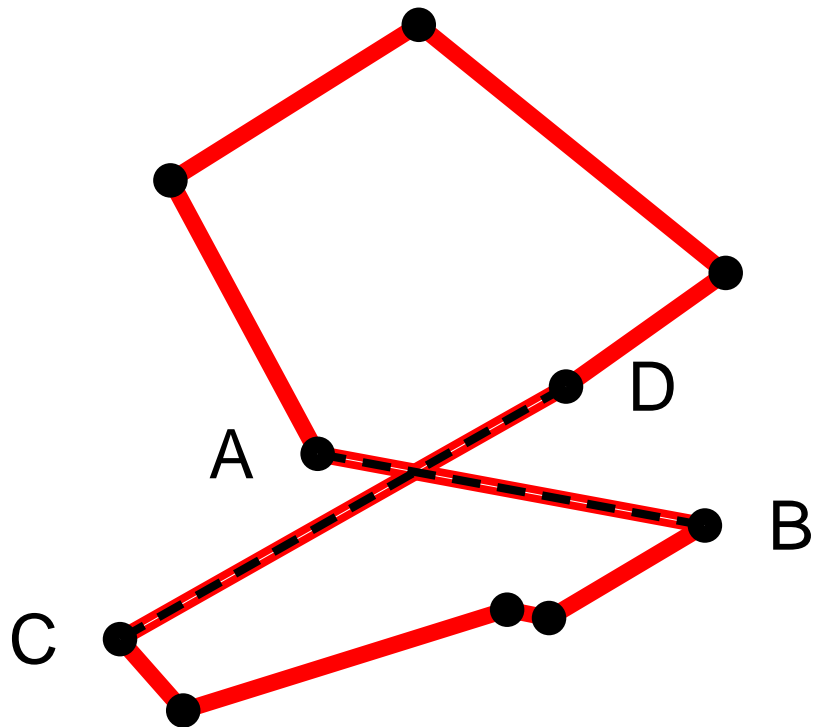


2-opt



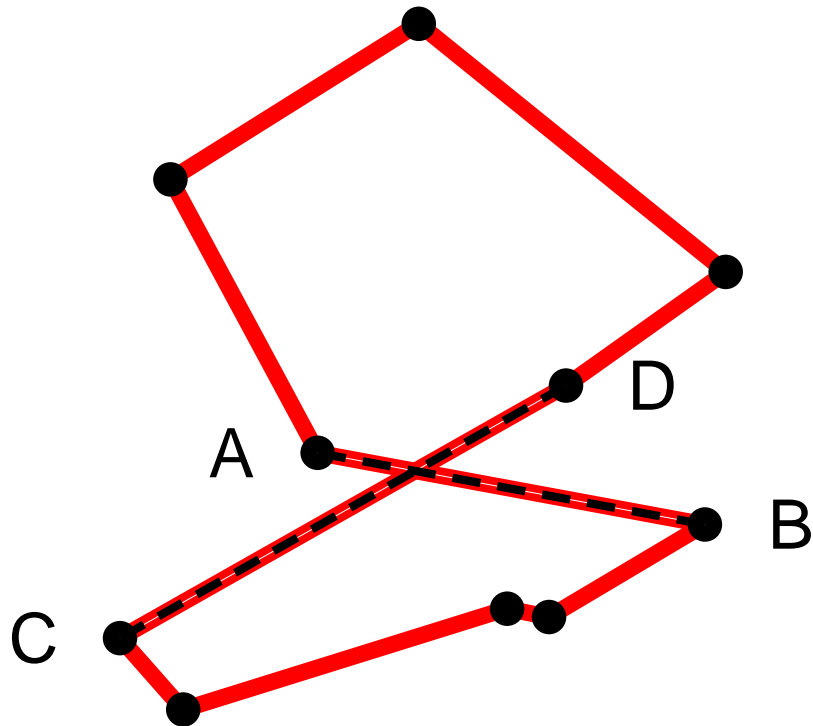
現在の巡回路長 D_0

2-opt

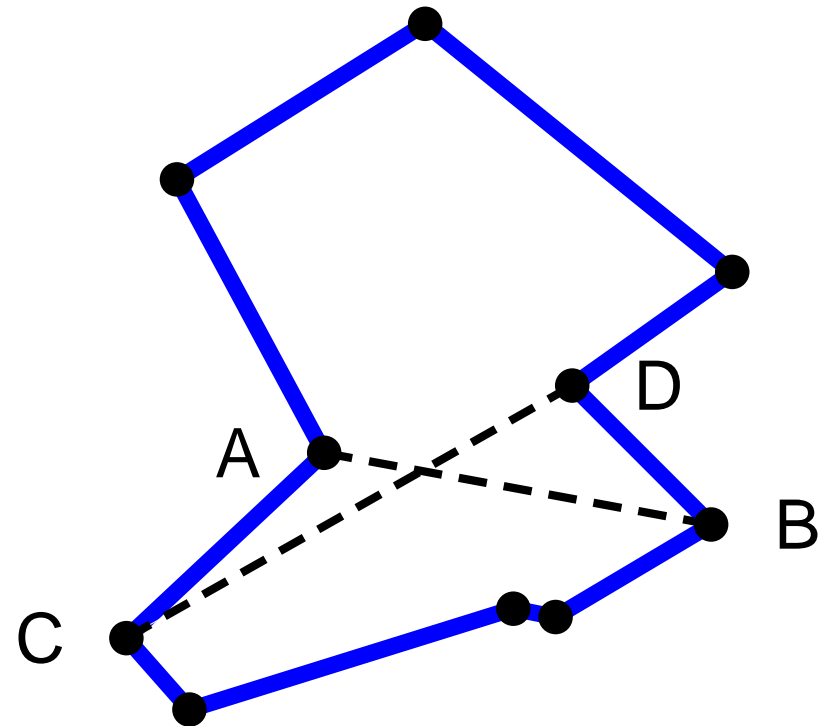


現在の巡回路長 D_0

2-opt

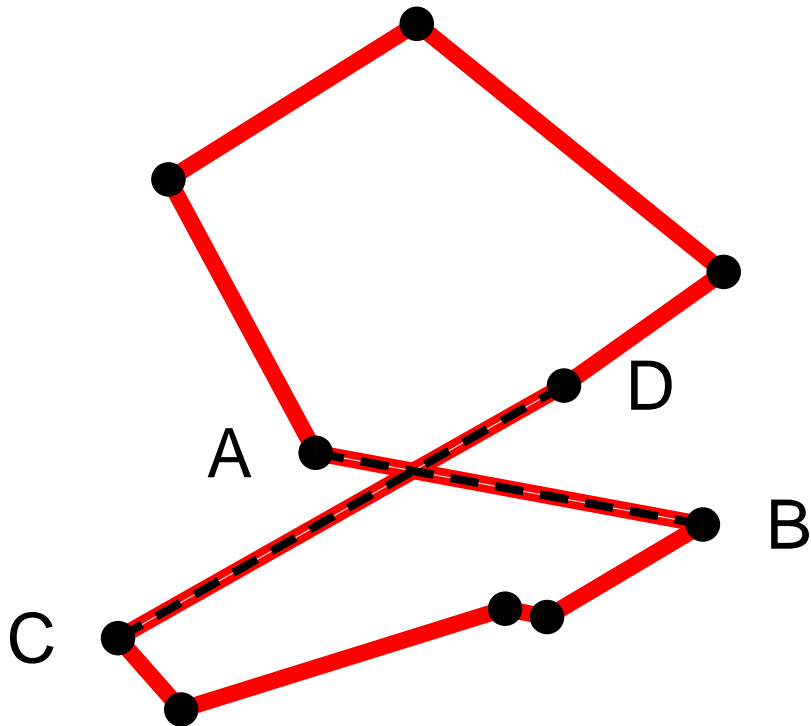


現在の巡回路長 D_0

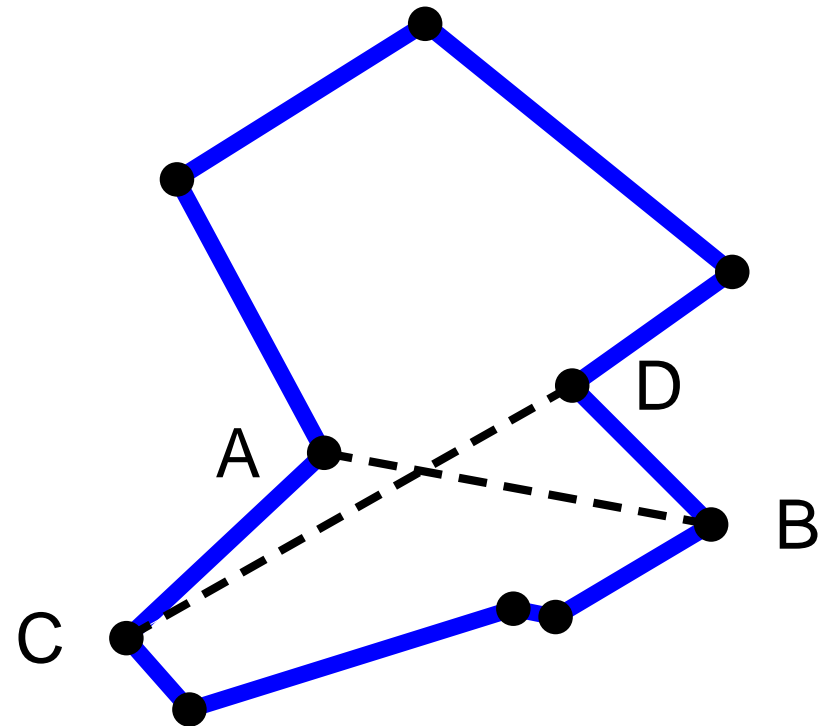


更新候補の巡回路長 D_1

2-opt



現在の巡回路長 D_0



更新候補の巡回路長 D_1

$D_0 > D_1$ の時 , 実際に更新する

2-opt を基盤とするカオス探索法

大規模問題へ適用可能なヒューリスティック解法
例えば, 2-opt 法

+

カオスニューラルネットワークのダイナミクス

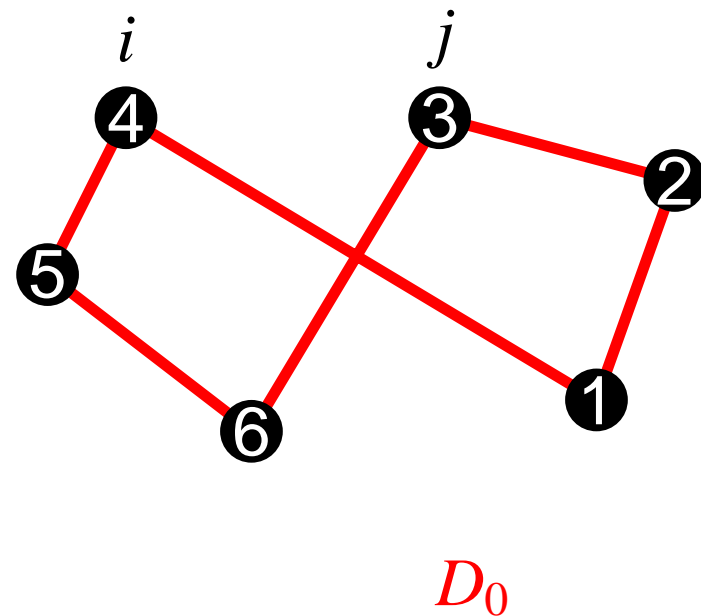
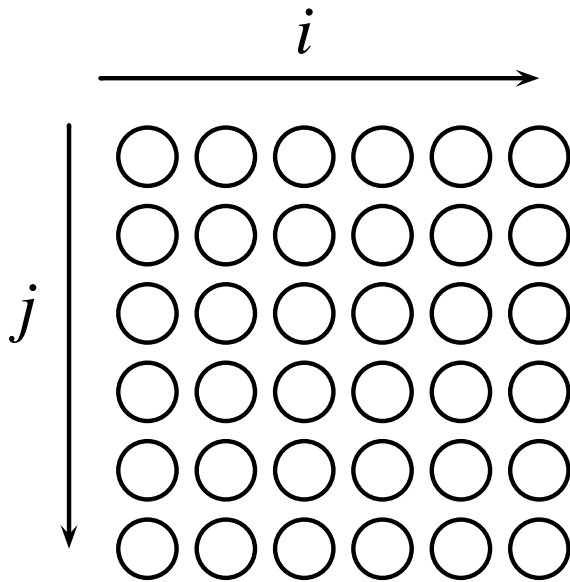
手順

1. 2-opt 法を実現するニューラルネットワークを構築
2. カオスニューラルネットワークへ変換

カオスニューラルネットワークが 2-opt 法を制御

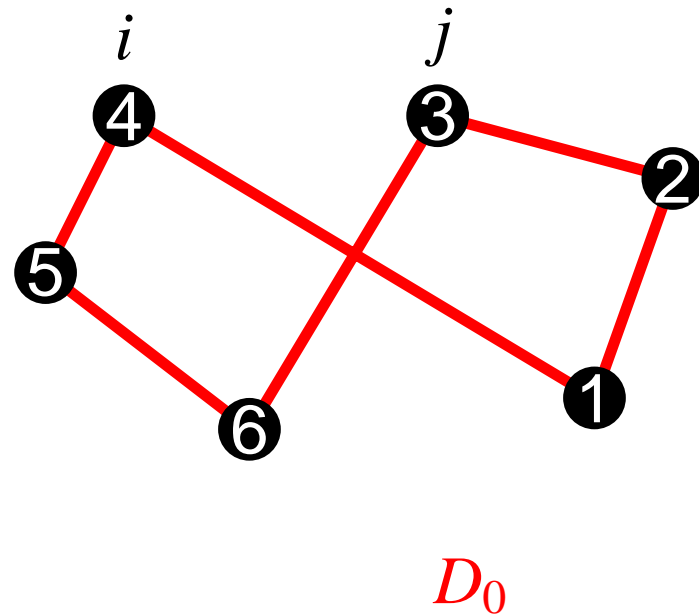
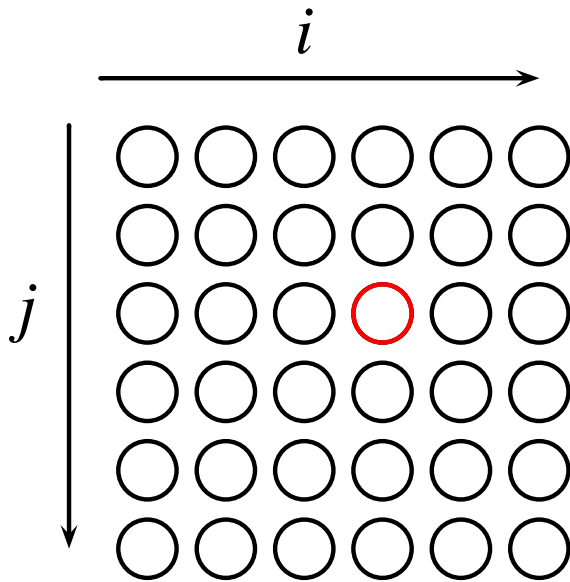
2-opt 法を実現するニューラルネット

- ◀ N -都市 TSP $\Rightarrow N \times N$ のニューロンを用意
- ◀ (i, j) のニューロンの発火
 \Leftrightarrow 都市 i の次に都市 j を訪問する経路を繋ぐ 2-opt



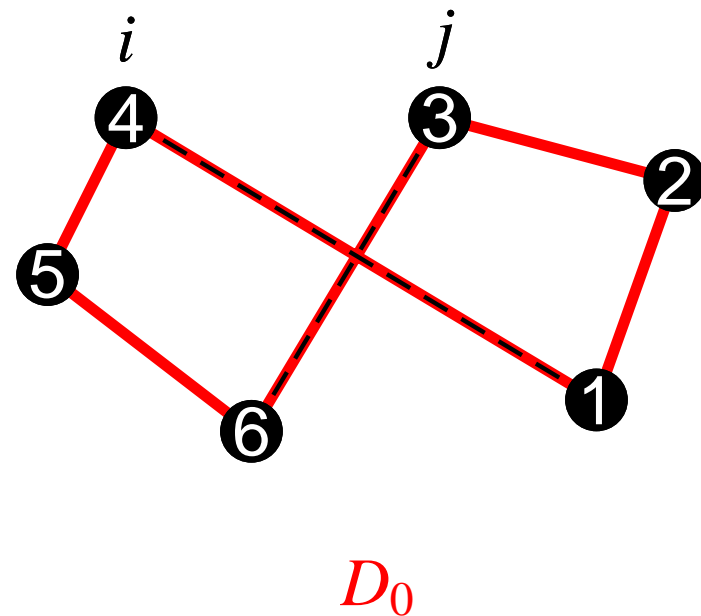
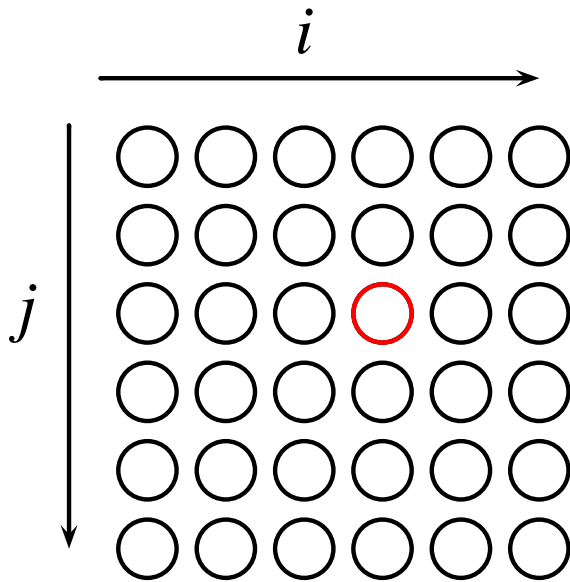
2-opt 法を実現するニューラルネット

- ◀ N -都市 TSP $\Rightarrow N \times N$ のニューロンを用意
- ◀ (i, j) のニューロンの発火
 \Leftrightarrow 都市 i の次に都市 j を訪問する経路を繋ぐ 2-opt



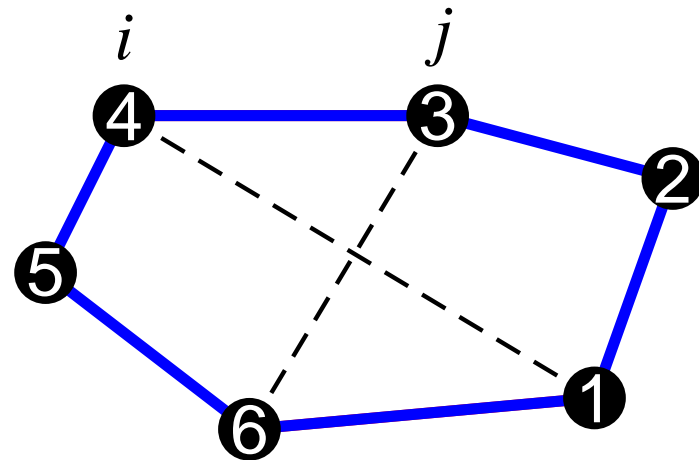
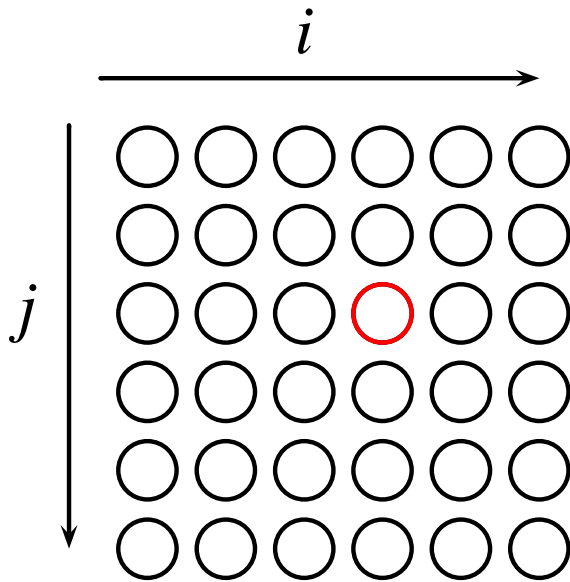
2-opt 法を実現するニューラルネット

- ◀ N -都市 TSP $\Rightarrow N \times N$ のニューロンを用意
- ◀ (i, j) のニューロンの発火
 \Leftrightarrow 都市 i の次に都市 j を訪問する経路を繋ぐ 2-opt



2-opt 法を実現するニューラルネット

- ◀ N -都市 TSP $\Rightarrow N \times N$ のニューロンを用意
- ◀ (i, j) のニューロンの発火
 \Leftrightarrow 都市 i の次に都市 j を訪問する経路を繋ぐ 2-opt



$$D_0 \Rightarrow D_{ij}$$

2-opt 法と CNN

1. 最急降下的に探索 \Rightarrow 2-opt

$$\xi_{ij}(t+1) = D_0(t) - D_{ij}(t)$$

$$\begin{cases} \xi_{ij}(t+1) \geq 0 & \text{発火：都市 } i \text{ と都市 } j \text{ を繋ぐ} \\ \xi_{ij}(t+1) < 0 & \text{静止：そのまま} \end{cases}$$

2. 2-opt 法を CNN が制御

$$\xi_{ij}(t+1) = D_0(t) - D_{ij}(t)$$

$$\eta_{ij}(t+1) = -C \sum_{l=1, l \neq i, j}^N x_{il}(t) - C \sum_{l=1, l \neq i, j}^N x_{lj}(t) - Bx_{ji}(t)$$

$$\zeta_{ij}(t+1) = -\alpha \sum_{d=0}^t k_r^d x_{ij}(t-d) - \theta$$

$$x_{ij}(t+1) = f[\xi_{ij}(t+1) + \eta_{ij}(t+1) + \zeta_{ij}(t+1)]$$

2-opt 法を制御する CNN

$$\xi_{ij}(t+1) = D_0(t) - D_{ij}(t)$$

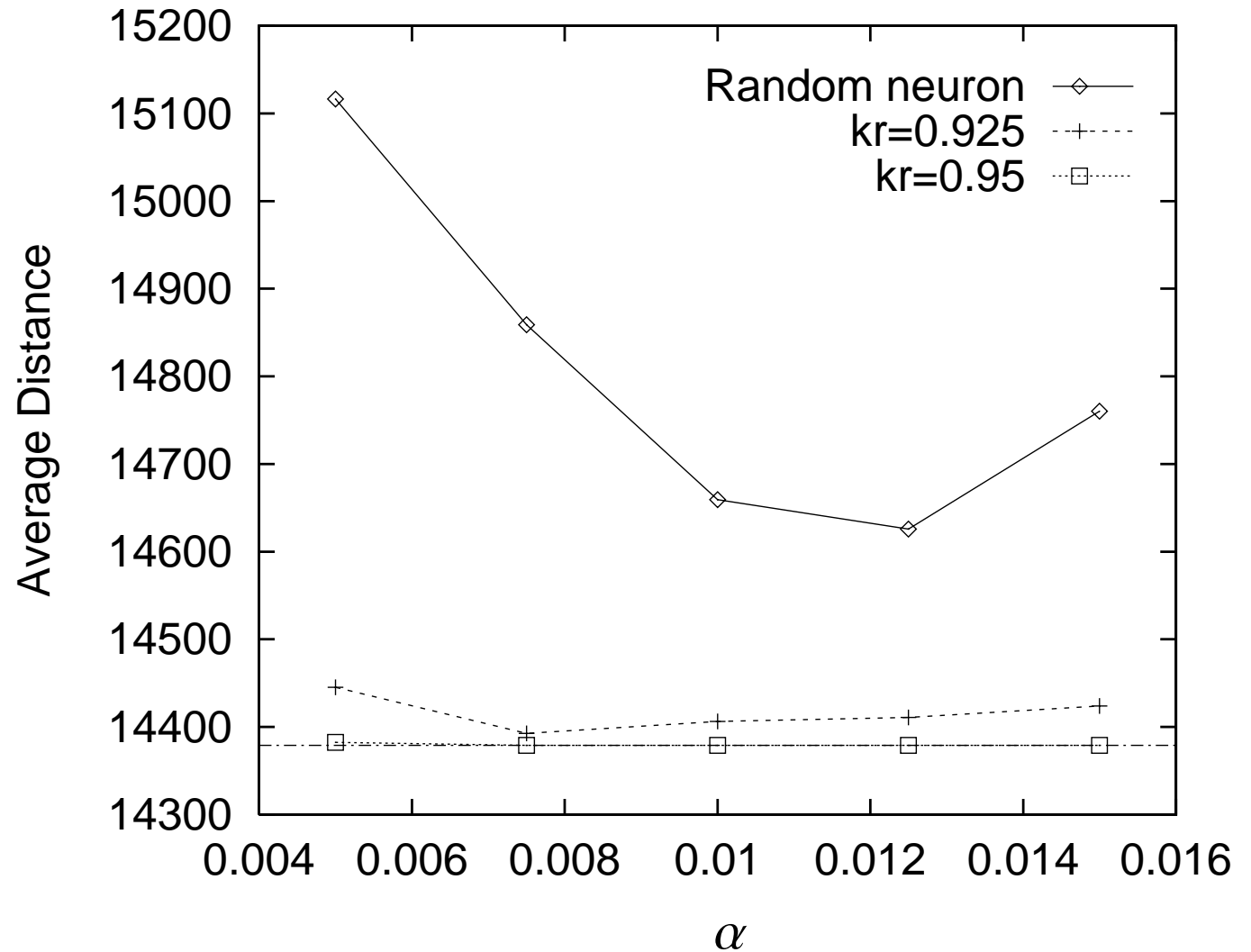
$$\eta_{ij}(t+1) = -C \sum_{l=1, l \neq i, j}^N x_{il}(t) - C \sum_{l=1, l \neq i, j}^N x_{lj}(t) - Bx_{ji}(t)$$

$$\zeta_{ij}(t+1) = -\alpha \sum_{d=0}^t k_r^d x_{ij}(t-d) - \theta$$

$$x_{ij}(t+1) = f[\xi_{ij}(t+1) + \eta_{ij}(t+1) + \zeta_{ij}(t+1)]$$

$$\begin{cases} x_{ij}(t) \geq 0.5 & \text{発火：都市 } i \text{ と都市 } j \text{ を繋ぐ} \\ x_{ij}(t) < 0.5 & \text{静止：そのまま} \end{cases}$$

Chaotic 2opt 法の性能



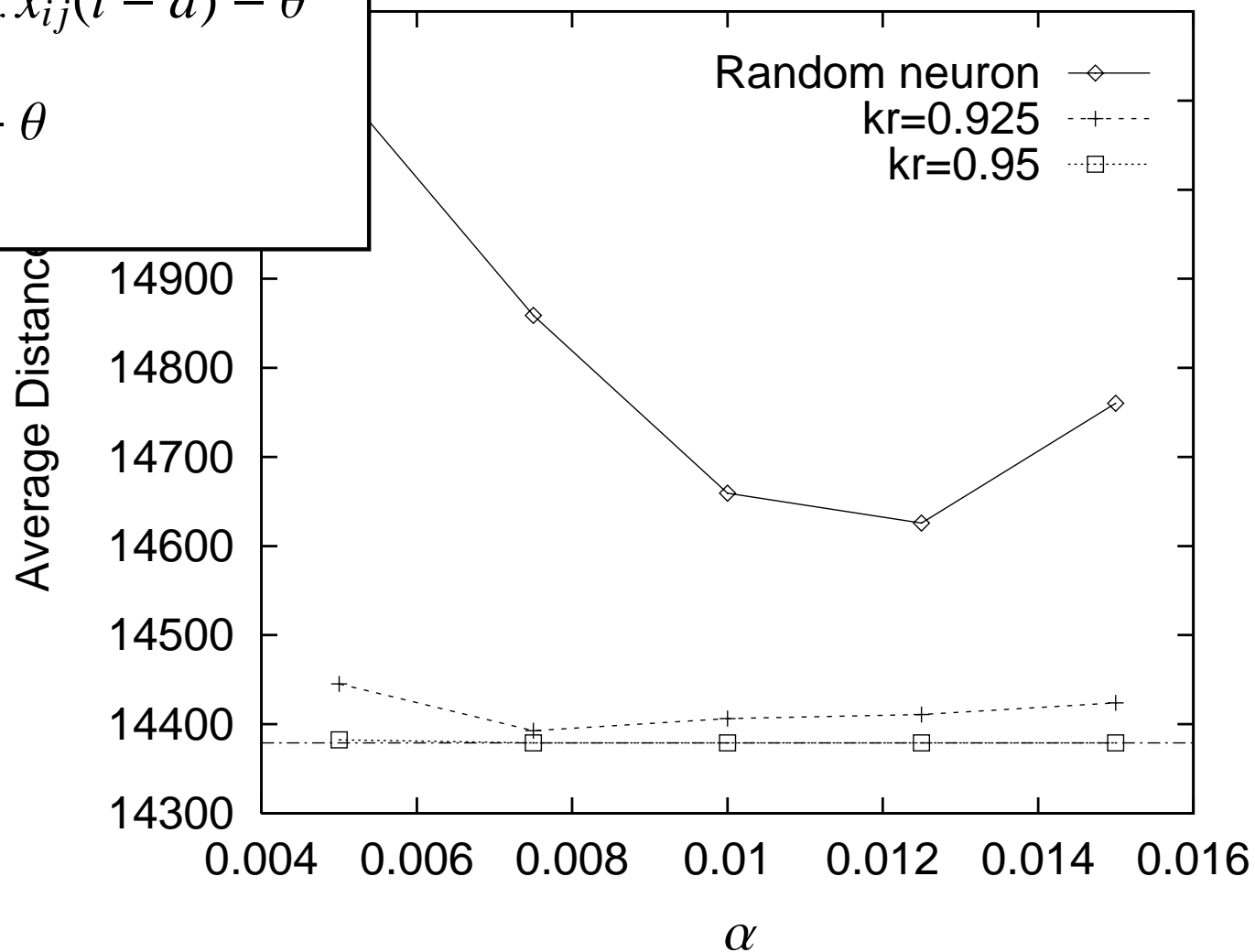
Lin105 ($k_r = 0.925, k_r = 0.95$, ランダム)

Chaotic 2opt 法の性能

$$\eta_{ij}(t+1) = -\alpha \sum_{d=0}^t k_r^d x_{ij}(t-d) - \theta$$

$$\eta_{ij}(t+1) = -\alpha Z(t) - \theta$$

$Z(t)$: 一様乱数



Lin105 ($k_r = 0.925, k_r = 0.95$, ランダム)

ロバスト性

	Optimum	CNN	Random Neuron
KroA100	21282	21285.8	21953.0
KroB100	22141	22150.7	22510.7
KroC100	20749	20749.7	21365.3
KroD100	21294	21294.0	21587.3
KroE100	22068	22078.7	22407.3

- ◀ 100 通りの初期値から得られた結果の平均値
- ◀ 10000 イタレーションでカット

パラメータ制御

問題	制御無し (%)	制御あり (%)
KroA100	0.3582	0.2048
KroB100	0.6240	0.4549
KroC100	0.2612	0.1221
KroD100	0.4702	0.3292
KroE100	0.5920	0.4622
Lin105	1.078	0.5282

- ◀ $\xi_{ij}(t + 1)$ (ゲイン入力) の平均と分散を制御 , 他は固定 .
- ◀ 100 通りの初期値から得られた結果の平均値と既知最良解とのギャップ
- ◀ 10000 イタレーションでカット

Chaotic 2-opt 法の性能

◀ Lin 318 (42,029)

□ 42,196 (0.397%)

$k_r = 0.875, k_m = 0.2, k_s = 0.0, R = 0.85, \alpha = 0.09, \epsilon = 0.003, C = 0.09, B = 0.045, h = 0.05$

□ 42,112 (0.197%)

$k_r = 0.875, k_m = 0.2, k_s = 0.0, R = 0.9, \alpha = 0.09, \epsilon = 0.003, C = 0.09, B = 0.045, h = 0.125$

ここまでのまとめ

- ◀ カオスを用いた組み合わせ最適化問題の従来解法
 - ホップフィールドニューラルネットワークを基盤
 - 10～20 都市程度が限界
 - 工学的な応用課題への適用は困難
 - ◀ ヒューリスティック解法とカオスダイナミクスの融合
 - 2-opt をニューラルネットで実現
 - 2-opt をカオスニューラルネットで実現
 - ⇒ 2-opt をカオスダイナミクスが制御
 - 100 都市以上の問題にも適用可能
1. Hasegawa, Ikeguchi, Matozaki, Aihara: IEICE Trans. Fund., Vol.E79–A, No.10, pp.1630–1637, 1996.
 2. Hasegawa, Ikeguchi, K. Aihara: Phys. Rev. Lett., Vol.79, No.10, pp.2344–2347, 1997.

タブーサーチとカオス

◀ タブーサーチ

- 近年，最も有効な解法の一つ
- 一度探索した状態を再度探索することを禁ずる
同じローカルミニマムを何度も探索しない

◀ カオスニューラルネットワーク

- カオスダイナミクスに基づく効率的な探索
- カオスニューロン (構成要素) は，
不応性 (refractoriness) を有する
⇔ 発火した直後は発火しづらい
⇔ 同じ状態は取りにくい

タブーサーチとカオスサーチの関連は？

2次割り当て問題 (QAP) への適用

$$F(p) = \sum_{i=1}^N \sum_{j=1}^N c_{ij} d_{p(i)p(j)}$$

- c_{ij} : フロー行列 C の第 (i, j) 要素
 d_{ij} : 距離行列 D の第 (i, j) 要素
 p_i : 順列 p の第 i 要素
 N : 問題サイズ

目的関数 F を最小化する順列 p を求めよ .

- ◀ タブーサーチ法 , シミュレーテッドアニーリング法
- ◀ 2つの要素を入れ替えて p を更新する

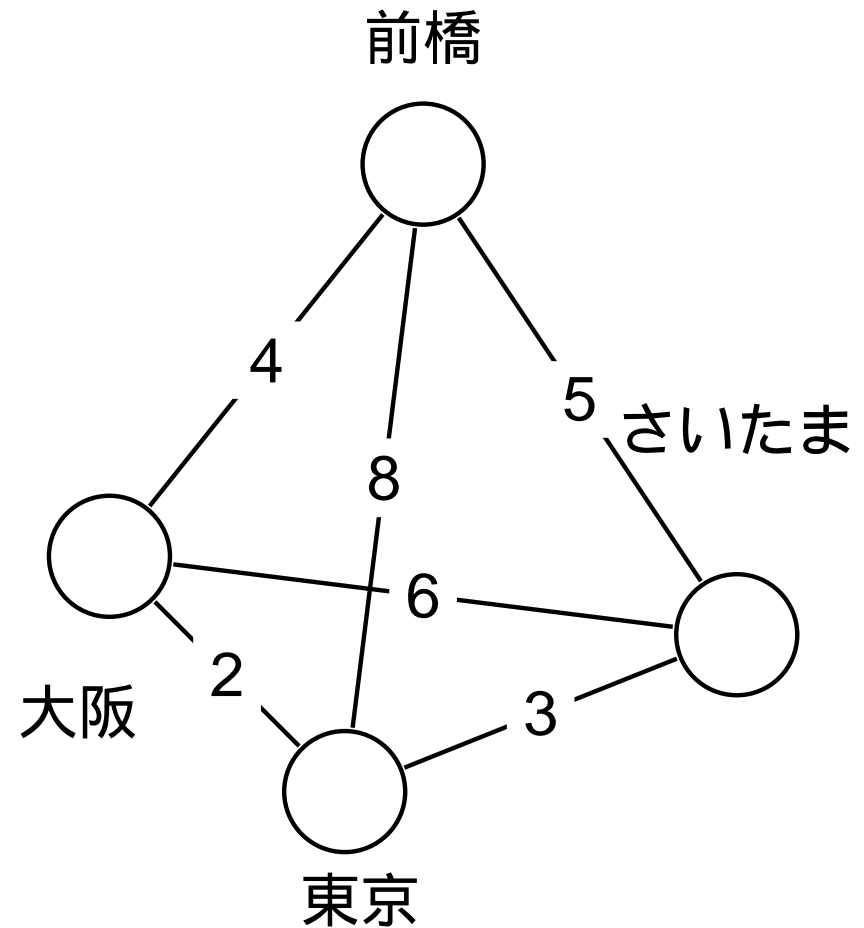
2次割り当て問題の例

1. 4つの工場 X, Y, Z, W を4都市に配置
2. 4工場間の商品流通量

$$C = \begin{array}{c} X \\ Y \\ Z \\ W \end{array} \begin{array}{cc} X & Y & Z & W \\ \begin{pmatrix} 0 & 4 & 3 & 5 \\ 4 & 0 & 6 & 3 \\ 3 & 6 & 0 & 4 \\ 5 & 3 & 4 & 0 \end{pmatrix} \end{array}$$

3. 4都市間の距離

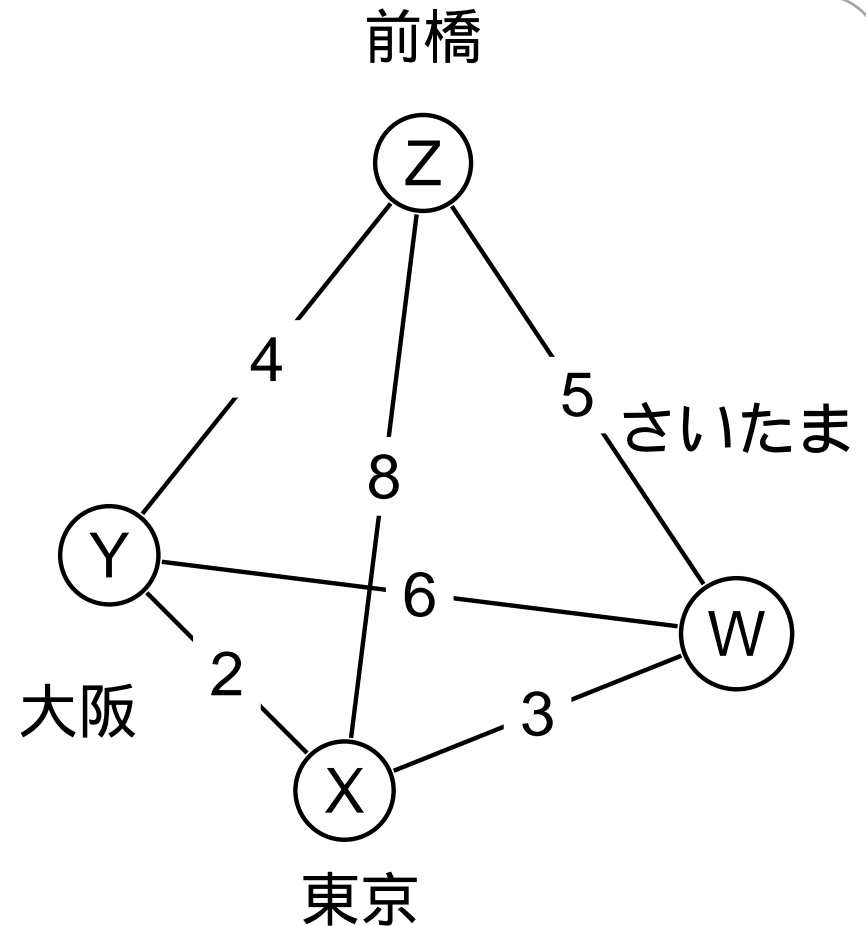
$$D = \begin{array}{c} T \\ O \\ M \\ S \end{array} \begin{array}{cc} T & O & M & S \\ \begin{pmatrix} 0 & 2 & 8 & 3 \\ 2 & 0 & 4 & 6 \\ 8 & 4 & 0 & 5 \\ 3 & 6 & 5 & 0 \end{pmatrix} \end{array}$$



2次割り当て問題の例

割り当て

X	東京
Y	大阪
Z	前橋
W	さいたま

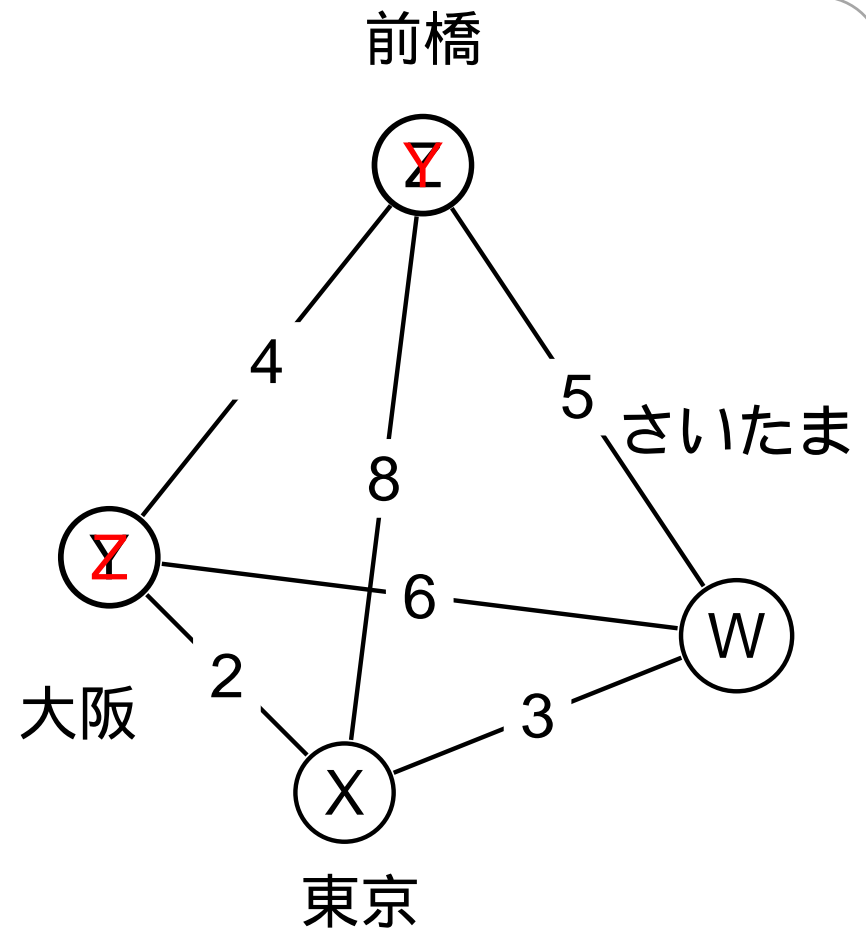


$$F(p_1) = 4 \cdot 2 + 3 \cdot 8 + 5 \cdot 3 + 6 \cdot 4 + 3 \cdot 6 + 4 \cdot 5 = 109$$

2次割り当て問題の例

割り当て

X	東京
Z	前橋
Y	大阪
W	さいたま



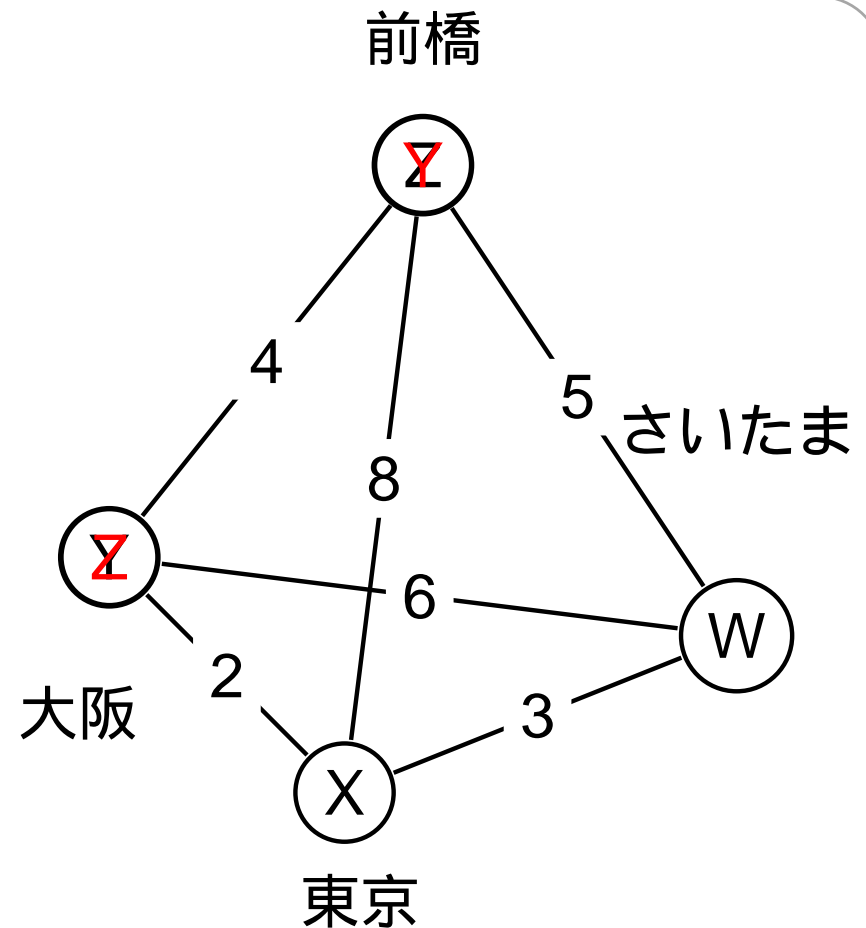
$$F(p_2) = 4 \cdot 8 + 3 \cdot 2 + 5 \cdot 3 + 6 \cdot 4 + 3 \cdot 5 + 4 \cdot 6 = 116$$

$$F(p_1) = 4 \cdot 2 + 3 \cdot 8 + 5 \cdot 3 + 6 \cdot 4 + 3 \cdot 6 + 4 \cdot 5 = 109$$

2次割り当て問題の例

割り当て

X	東京
Z	前橋
Y	大阪
W	さいたま




$$F(p_2) = 4 \cdot 8 + 3 \cdot 2 + 5 \cdot 3 + 6 \cdot 4 + 3 \cdot 5 + 4 \cdot 6 = 116$$

$$F(p_1) = 4 \cdot 2 + 3 \cdot 8 + 5 \cdot 3 + 6 \cdot 4 + 3 \cdot 6 + 4 \cdot 5 = 109$$

$$F(p_1) < F(p_2)$$

タブーサーチ法 (Taillard, 1991)

$$p : \begin{pmatrix} 1 & \dots & j & \dots & q(i) & \dots & n \\ 1 & , \dots , & p(j) & , \dots , & i & , \dots , & n \end{pmatrix}$$


A diagram showing a mapping from the element $p(j)$ in the second row to the element i in the second row, and from the element $q(i)$ in the first row to the element i in the second row. The arrows indicate that $p(j)$ and $q(i)$ are both mapped to i .

i を j に割り当てる
 $p(j)$ は $q(i)$ に割り当てられる

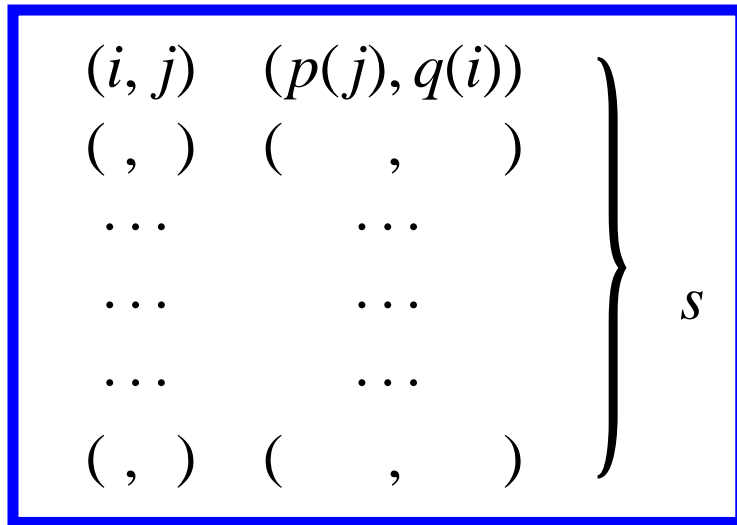
$$p : \begin{pmatrix} 1 & \dots & j & \dots & q(i) & \dots & n \\ 1 & , \dots , & i & , \dots , & p(j) & , \dots , & n \end{pmatrix}$$



$i \rightarrow j$ 及び $p(j) \rightarrow q(i)$ をタブーリストに

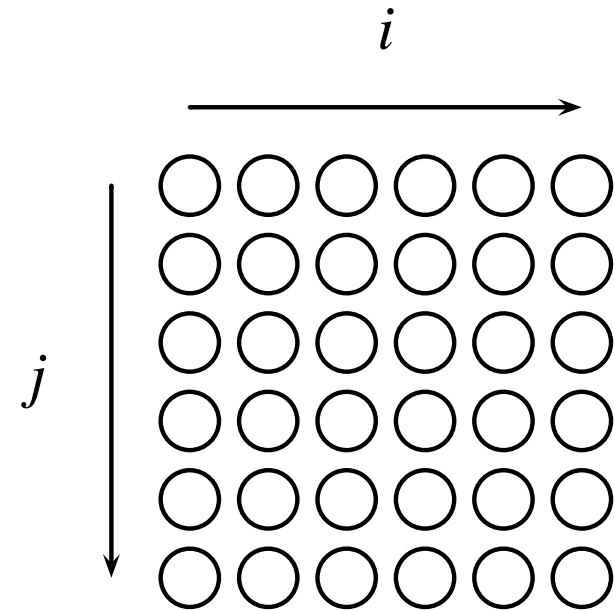
タブーサーチを実現するニューラルネット

タブーリスト



$n \times n$ 種類のペア

$$p : \begin{pmatrix} 1 & \dots & j & \dots & q(i) & \dots & n \\ 1 & , \dots , & p(j) & , \dots , & i & , \dots , & n \end{pmatrix}$$



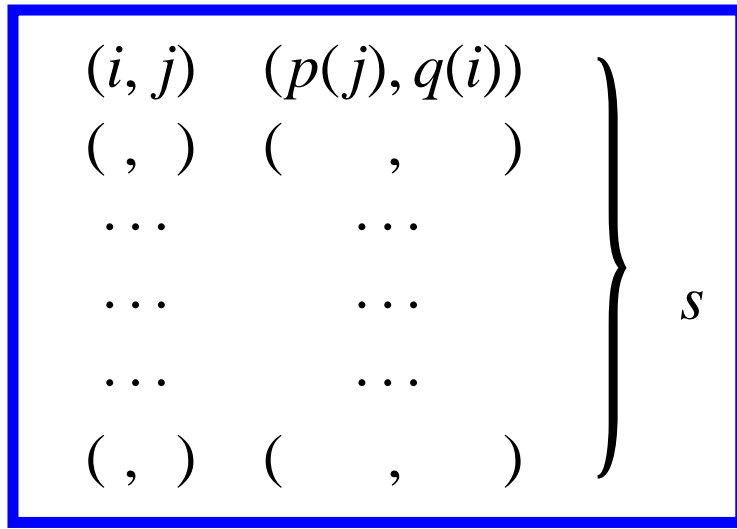
(i, j) ニューロンの発火



i と $p(j)$ を入れ替える .

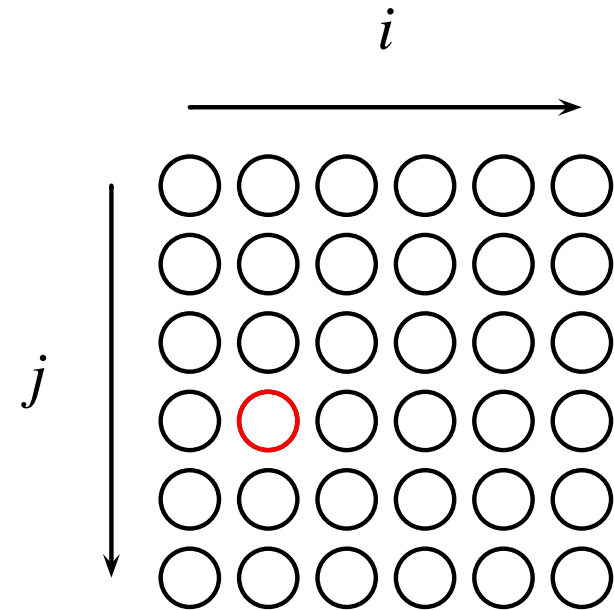
タブーサーチを実現するニューラルネット

タブーリスト



$n \times n$ 種類のペア

$$p : \begin{pmatrix} 1 & \dots & j & \dots & q(i) & \dots & n \\ 1 & , \dots , & p(j) & , \dots , & i & , \dots , & n \end{pmatrix}$$



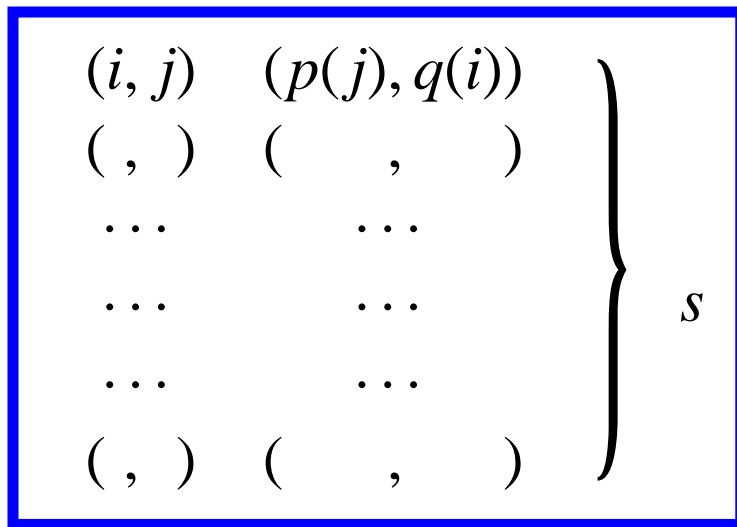
(i, j) ニューロンの発火



i と $p(j)$ を入れ替える .

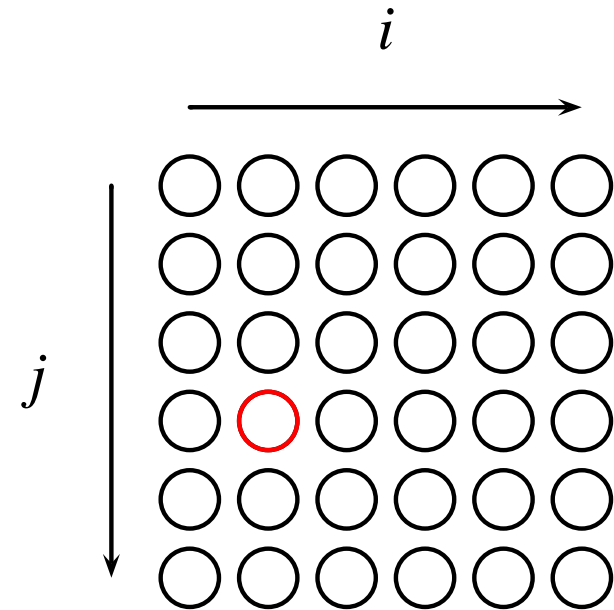
タブーサーチを実現するニューラルネット

タブーリスト



$n \times n$ 種類のペア

$$p : \begin{pmatrix} 1 & \dots & j & \dots & q(i) & \dots & n \\ 1 & , \dots , & p(j) & , \dots , & i & , \dots , & n \end{pmatrix}$$



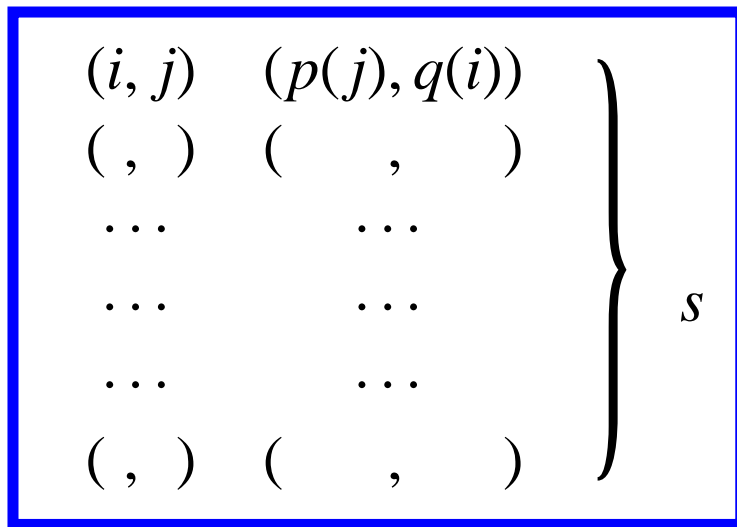
(i, j) ニューロンの発火



i と $p(j)$ を入れ替える .

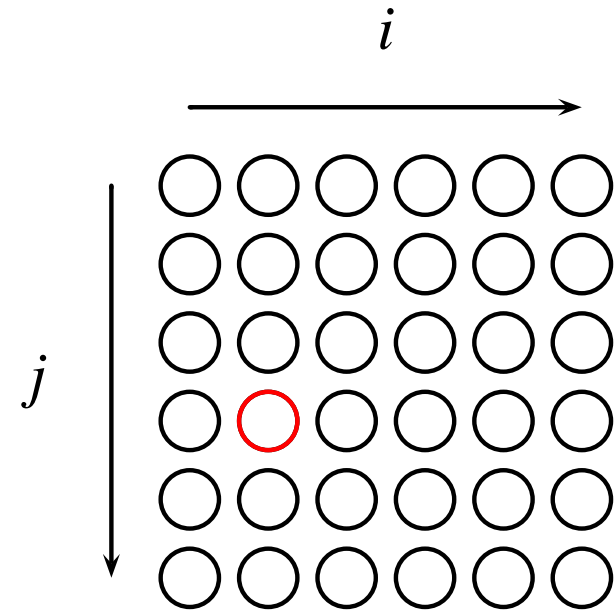
タブーサーチを実現するニューラルネット

タブーリスト



$n \times n$ 種類のペア

$$p : \begin{pmatrix} 1 & \dots & j & \dots & q(i) & \dots & n \\ 1 & , \dots , & i & , \dots , & p(j) & , \dots , & n \end{pmatrix}$$



(i, j) ニューロンの発火



i と $p(j)$ を入れ替える .

タブーサーチを実現するニューラルネット

$$\begin{aligned}\xi_{ij}(t+1) &= \beta \Delta_{ij}(t) \leftarrow \text{ゲイン入力} \\ \gamma_{ij}(t+1) &= -\alpha \sum_{d=0}^{s-1} k_r^d x_{p(j)q(i)}(t-d) \\ \eta_{ij}(t+1) &= -\alpha \sum_{d=0}^{s-1} k_r^d x_{ij}(t-d)\end{aligned}$$

- ◀ $\xi_{ij}(t+1) + \gamma_{ij}(t+1) + \eta_{ij}(t+1)$ が最大となる (i, j) ニューロンが発火したとする $\Rightarrow i \rightarrow j, p(j) \rightarrow q(i)$ の割当を実行
- ◀ 上記割当を記憶するために, $x_{ij}(t+1) = 1, x_{p(j)q(i)} = 1$.
- ◀ $\alpha \rightarrow \infty, k_r = 1$, s : タブーリストサイズ

$$\mathbf{p} : \left(\begin{array}{ccccccc} 1 & \cdots & j & \cdots & q(i) & \cdots & n \\ 1 & , \cdots , & p(j) & , \cdots , & i & , \cdots , & n \end{array} \right)$$

タブーサーチからカオスサーチへ

◀ タブーサーチニューラルネットワーク

- タブー効果を持つ
- 出力関数:最大発火 \Rightarrow 0 or 1

◀ カオスニューラルネットワーク

- 不応性
- 出力関数:シグモイド \Rightarrow アナログ値
ステップ関数 (0 or 1) ではカオスは生じない
cf. 南雲-佐藤ニューロンモデル



◀ タブーサーチを含むカオスサーチの実現

- (1) 不応性 (タブー) を有する .
- (2) 出力関数をシグモイド関数とする .

タブーサーチを含むカオスサーチ法

$$\xi_{ij}(t+1) = \beta \Delta_{ij}(t) \leftarrow \text{ゲイン入力}$$

$$\eta_{ij}(t+1) = -W \sum_{k=1}^n \sum_{l=1, (k \neq i \vee l \neq j)}^n x_{kl}(t) + W$$

$$\gamma_{ij}(t+1) = -\alpha \sum_{d=0}^t k_r^d \{x_{p(j)q(i)}(t-d) + z_{p(j)q(i)}(t-d)\} + \theta$$

$$\zeta_{ij}(t+1) = -\alpha \sum_{d=0}^t k_r^d \{x_{ij}(t-d) + z_{ij}(t-d)\} + \theta,$$

$$x_{ij}(t+1) = f\{\xi_{ij}(t+1) + \eta_{ij}(t+1) + \gamma_{ij}(t+1) + \zeta_{ij}(t+1)\}$$

- ◀ $x_{ij}(t+1) > 0.5$ の時 , (i, j) ニューロンが発火 .
- ◀ (i, j) ニューロンの更新時に $z_{ij}(t+1) = 0$ とし , $x_{ij}(t+1)$ を $z_{p(j)q(i)}(t+1)$ に加算

結果 1

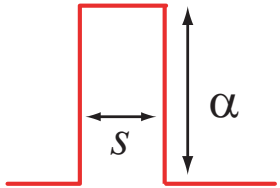
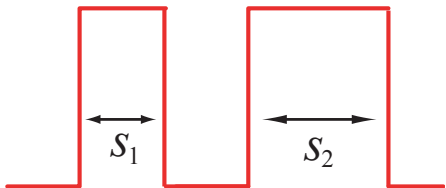
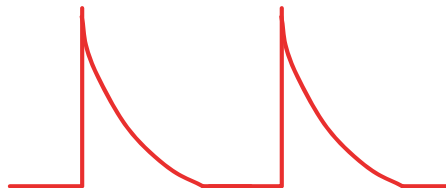
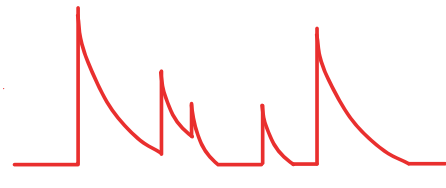
Problem	TS	Ro-TS		Ex-TS	CNN
		$(n \pm 10\%)$	$(5n \pm 10\%)$		
tai20b	13.3	15.9	9.38	4.94	3.00
tai30b	9.94	13.2	9.04	5.93	3.81
tai40b	7.19	8.31	6.25	5.27	4.05
tai50b	4.50	5.74	5.07	4.13	3.62

- ◀ CNN: カオス
- ◀ TS: 従来のタブーサーチ
- ◀ Ro-TS: ロバストタブーサーチ
- ◀ Ex-TS: タブーの影響が指数関数的に減衰
- ◀ 100 通りの初期値 . 既知最良解からのギャップ (%)

結果2

Problem	Random	SSA	Ro-TS1 ($n \pm 10\%$)	Ro-TS2 ($20n \pm 10\%$)	CNN	CSA
bur26a	1.04	0.162	0.133	0.147	0.159	0.0935
bur26b	0.819	0.246	0.246	0.102	0.0814	0.107
bur26c	0.610	0.0922	0.0855	0.0362	0.0496	0.0430
bur26d	0.425	0.106	0.108	0.0271	0.0234	0.0355
els19	29.4	16.7	10.3	8.13	8.90	5.27
ste36a	18.8	5.68	1.98	3.85	5.65	2.18
te36b	57.0	7.72	4.84	7.42	12.7	3.65
ste36c	15.3	3.99	1.67	3.09	4.40	1.57
tai20b	18.1	8.78	5.59	0.626	1.80	1.02
tai30b	9.35	8.48	10.3	2.28	2.33	1.67
tai40b	8.14	7.26	8.31	2.88	3.70	3.07
tai50b	5.21	4.16	6.48	1.854	2.21	1.851
tai60b	5.53	3.67	8.37	3.13	2.52	1.67
tai80b	4.28	3.81	5.46	2.64	2.88	2.23
tai150b	2.84	2.67	3.07	1.63	2.44	1.42

タブーの効果の比較

	パラメータ	出力関数	記憶
Tabu Search	$k_r = 1$ $\alpha \rightarrow \infty$ s : 固定	最大発火 0 or 1	
Robust TS	$k_r = 1$ $\alpha \rightarrow \infty$ s : ランダム		
Exponential TS	$0 < k_r < 1$ α : 有限 $s - 1 = t$		
CNN	$0 < k_r < 1$ α : 有限 $s - 1 = t$	シグモイド関数 [0, 1]	

ここまでのまとめ

- ◀ ロバストタブーサーチ法 (従来法の一つ)
 - タブーリストサイズをランダムに変化
 - 最も有効な解法の一つ
 - QAP@QAPLIB の既知最良解を解いている
 - ◀ タブーサーチを含むカオスサーチ法
 - タブーの効果とカオスニューロンの不応性
 - 出力のアナログ値化 ⇒ カオス応答
 - ロバストタブーサーチよりも良好な結果
1. Hasegawa, Ikeguchi, Aihara: Control and Cybernetics, Vol.29, No.3, pp.773–788, 2000.
 2. Hasegawa, Ikeguchi, Aihara, Itoh: European Journal of Operational Research, Vol.139, No.3, pp.543–556, 2002.

カオスサーチと大規模TSP

◀ タブーサーチ法を基盤とするカオスサーチ法

- タブーサーチの拡張がカオスサーチ
- 2次割当問題に対して適用



◀ 巡回セールスマン問題に応用

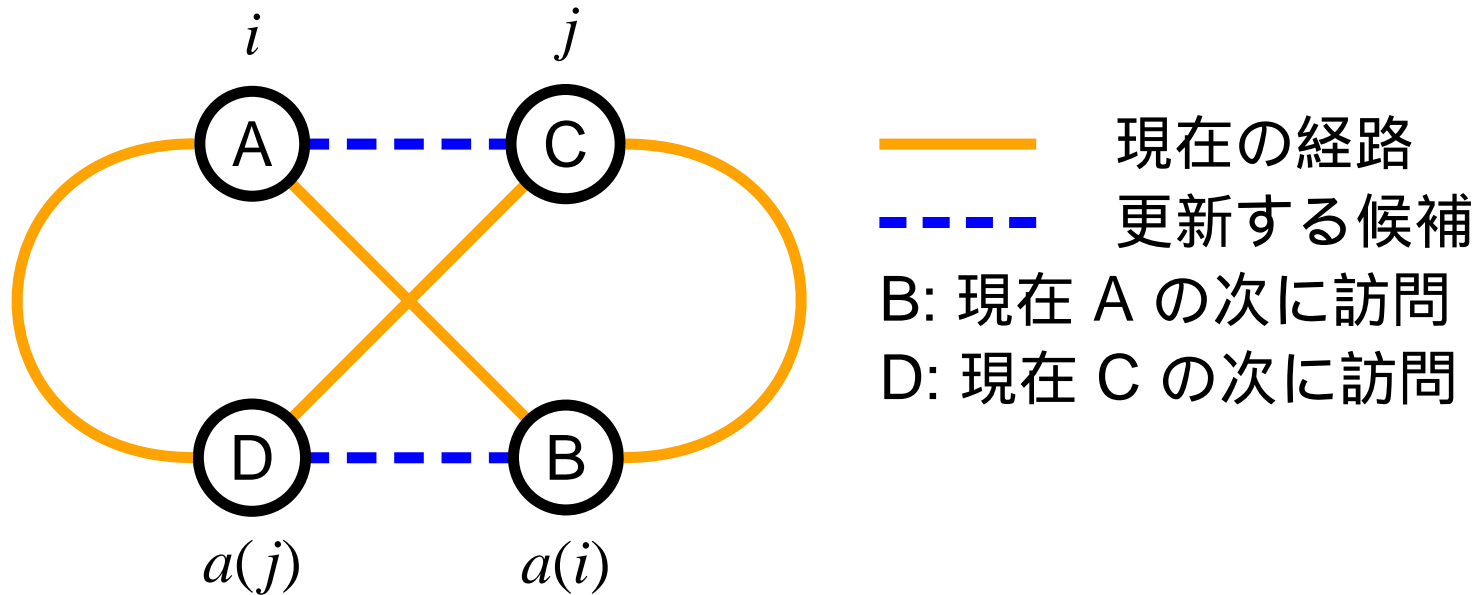
- 数千～数万都市
- 単純なタブーサーチ法を基盤



◀ 大きいサイズのTSPに適用可能なカオスサーチの構築

基盤とするタブーサーチ

都市のラベルを記憶するタブーサーチ



- ◀ タブーリストには, A に対応する都市 i を入れる
- ◀ 都市 A にとって,
最大ゲインとなる都市 C のラベルを j とする
- ◀ i, j 共にタブーでない時, 2-opt を実行

タブーサーチニューラルネット

$$\xi_{\textcolor{red}{i}}(t+1) = \max_j \{ \zeta_{\textcolor{blue}{j}}(t+1) + \beta \Delta_{\textcolor{red}{i}\textcolor{blue}{j}}(t) \}$$

$$\zeta_{\textcolor{red}{i}}(t+1) = -\alpha \sum_{d=0}^{s-1} k_r^d x_{\textcolor{red}{i}}(t-d)$$

- ◀ $\xi_i(t+1)$: ゲインの入力 (外部入力)
- ◀ $\zeta_i(t+1)$: i に関する更新のタブー
- ◀ $\xi_i(t+1) + \zeta_i(t+1)$ が最大となる i ニューロンを発火



- ◀ i と j を繋ぐように 2-opt で更新
- ◀ ニューロン i の発火を記憶するため $x_i(t+1) = 1$.
他ニューロン ($k \neq i$) の出力 $x_k(t+1) = 0$.

カオスニューラルネット

$$\xi_i(t+1) = \max_j \{ \zeta_j(t+1) + \beta \Delta_{ij}(t) \}$$

$$\eta_i(t+1) = -W \sum_{k=1}^N x_k(t) + W$$

$$\zeta_i(t+1) = -\alpha \sum_{d=0}^t k_r^d x_i(t-d) - \theta$$

$$x_i(t+1) = f[\xi_i(t+1) + \eta_i(t+1) + \zeta_i(t+1)]$$

◀ $x_i(t+1) > 0.5$ の時 , ニューロン i が発火



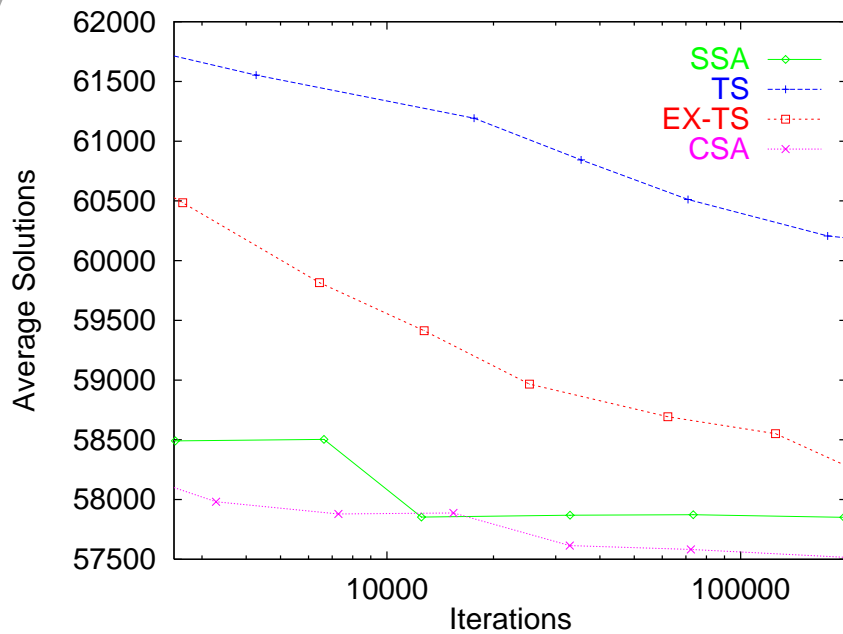
◀ i と j をつなぐように 2-opt で更新

結果

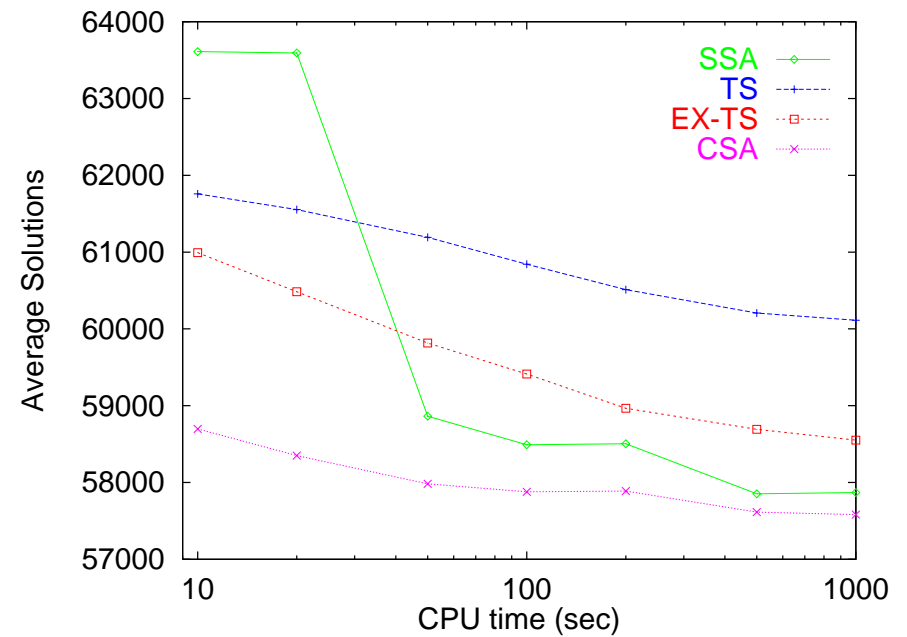
Problem	BM	TS	Ex-TS	CNN
KroA100	0.183	2.629	0.331	0.059
Lin105	0.562	2.199	0.200	0.000
KroA200	1.158	3.985	1.570	0.767
Lin318	2.211	5.120	2.705	1.599
Pcb1173	3.785	6.629	4.494	2.605

- ◀ BM : ボルツマンマシン (ランダム)
- ◀ TS : タブーサーチ
- ◀ Ex-TS : タブーの影響が指数関数的に減衰
- ◀ CNN : カオス

計算量比較



平均解とイタレーション数



平均解とCPU時間

CNNによる結果

problem	制御なし	制御あり	制御+アニーリング
Lin318	2.032	1.420	0.9946
Pcb1173	3.842	4.124	2.678
Pcb3038	4.490	4.483	2.967
Pla7397	5.561	5.580	2.192
RI5915	4.406	4.304	2.823
RI11849	3.892	3.919	2.486
Pla33810	9.708	7.850	3.372
Pla85900	10.995	6.697	3.046

- ◀ 解の平均値と既知最適解 (Pla33810, Pla85900 は下限) とのギャップ (%)
- ◀ 初期値 100 通り , 10,000 イタレーション

ここまでのまとめ

タブーサーチ



ニューラルネットワーク上で実現



カオスダイナミクスを有する形に拡張

カオスダイナミクスに基づく TSP の解法

- ◀ タブーサーチとカオスサーチを含む
- ◀ 高い解法能力
- ◀ 数万都市の TSP に適用

1. Hasegawa, Ikeguchi, Aihara: Neural Networks, Vol.15, No. 1, pp. 271–283, 2002.

おわりに

◀ カオスダイナミクスと組み合わせ最適化

相互結合型ニューラルネット \Rightarrow 小さなサイズのみ

1. 画像領域分割のローカルミニマム問題
 \Rightarrow カオスダイナミクス と 2-opt の融合 \Rightarrow TSP
2. Tabu Search から Chaos Search \Rightarrow QAP
3. Tabu Search から Chaos Search \Rightarrow TSP

カオスダイナミクスがヒューリスティックを制御する

◀ 今後の展望

- ダイナミクスの定量化
- 種々の問題への適用
グラフ分割問題, DNA からのモチーフ抽出
- 回路実装