最適化の数理──応用数理の視点

アルゴリズム

室田 一雄
計数工学科 (工学部)
数理情報学専攻 (情報理工学系)

http://www.misojiro.t.u-tokyo.ac.jp/∼murota

１．アルゴリズムとは…

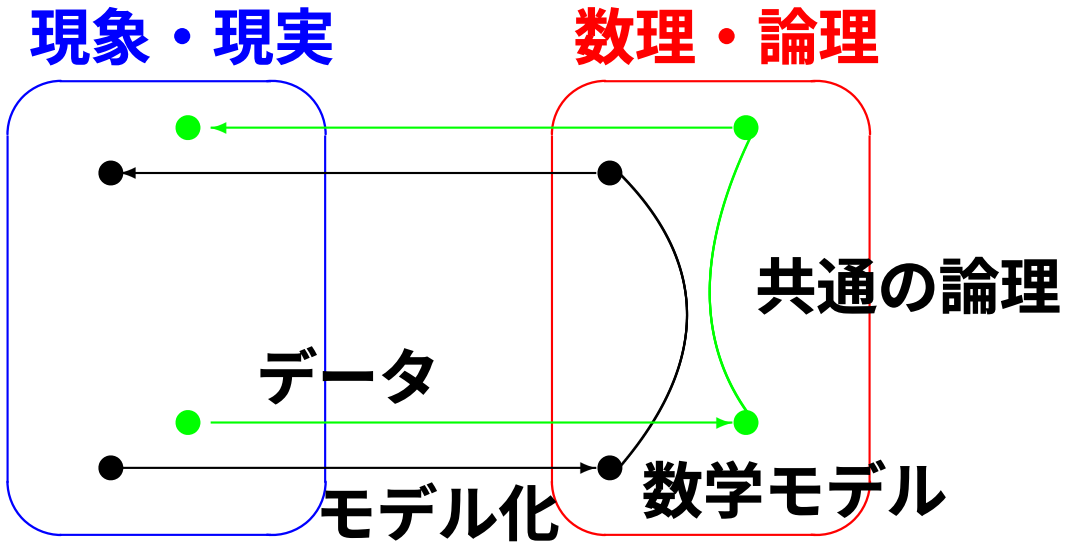
２．最適化の計算とは…

最適化の世界

（復習）

連続／離散

線形／凸／非線形



モデリング ＋ 理論 ＋ アルゴリズム

1．アルゴリズムとは…

Algorithm

アルゴリズムとは

有限的・機械的な計算手順

Algorithm ⟵ アル＝フワーリズミー

(算法) (780年頃–850年頃；アラビア)

cf. プログラム(算譜)

```
int v;
for(v = 1; v <= n; v++){ vfirst[v] = 0; }
for(int a = m; a > 0; a--){
  int v1 = head[2*a - 1];
  adjlist[a] = vfirst[v1];
  vfirst[v1] = a;
  }
```

cf. 存在定理

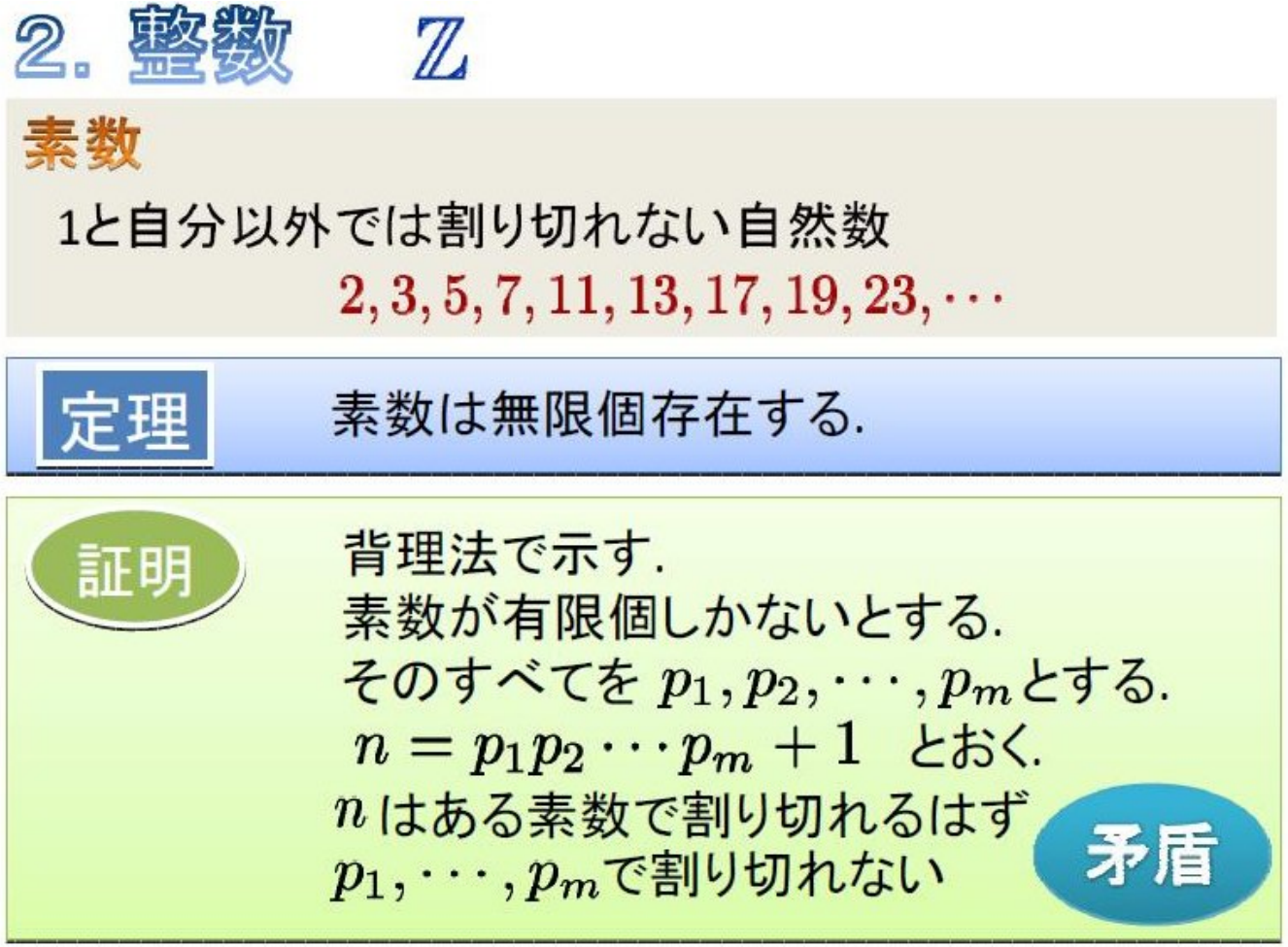
素数は無限に存在

(背理法の証明)

構成的でない存在証明（背理法）

素数は無限に存在

（桂先生のスライド）



この証明から素数の生成法は分からない

構成的な存在証明——とても簡単な例

定理：　偶数は無限に存在する

証明（帰納法）：

1）$n = 2$ は偶数である

2）ある自然数 n が偶数ならば，$n + 2$ も偶数である

この証明により無限個の偶数が生成できる

数学におけるアルゴリズムの例

最大公約数　　ユークリッドの互除法

(Euclidean algorithm)

3. ユークリッドの互除法

（桂先生のスライド）

補題

a, b を0でない2つの整数

$$a = qb + r \quad (q, r : \text{整数})$$

であるとする．このとき，$\gcd(a, b) = \gcd(b, r)$

←性質・事実
（静的）

例

$$\begin{aligned} 54 &= 2 \times 20 + 14 \\ 20 &= 1 \times 14 + 6 \\ 14 &= 2 \times 6 + 2 \\ 6 &= 3 \times 2 \end{aligned}$$

54と20の最大公約数は 2

←計算手続き
（動的）

数学におけるアルゴリズムの例

最大公約数　ユークリッドの互除法　**高速**　$\log n$

(Euclidean algorithm)

素数判定　エラトステネスの篩　**低速**　$\sqrt{n}$

2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, $\cdots$

作図問題　（定規とコンパスで）

- 正５角形は**可能**　　正７角形は**不可能**
- 角の２等分は**可能**　　３等分は**不可能**

有限回の基本操作　　**可能／不可能**　　**高速／低速**

アルゴリズムの理論（1）

アルゴリズム ＝ 有限的・機械的な計算手順

計算とは何か？　コンピュータに何ができるのか？

計算可能性　　(1930年代)

いろいろな計算モデルの同値性：

Turing機械計算可能 ≡ 帰納的関数 ≡ λ計算可能

⇒　Church–Turingの提唱

停止問題　（計算不可能の例）

プログラム e とその入力値 x が与えられて，
その実行が有限時間で終了するか判定する問題

$$\mathrm{Halt}(e,x) = \begin{cases} \mathrm{yes} & (\text{有限時間で停止する}) \\ \mathrm{no} & (\text{永久に停止しない}) \end{cases}$$

定理：

Halt を計算するアルゴリズムは存在しない

証明

Halt にアルゴリズムが存在しない

$$\mathrm{Halt}(e,x) = \begin{cases} \mathrm{yes} & (\text{有限時間で停止する}) \\ \mathrm{no} & (\text{永久に停止しない}) \end{cases}$$

$$f(e) = \begin{cases} 0 & (\mathrm{Halt}(e,e) = \mathrm{no}) \\ \text{未定義} & (\mathrm{Halt}(e,e) = \mathrm{yes}) \end{cases}$$

Halt にアルゴリズムが存在⇒fにアルゴリズムが存在

fを計算するプログラムにfを入力すると……

$\mathrm{Halt}(f,f) = \mathrm{no}$ ⇔ $f(f) = 0$

⇔ fはfに対して停止 ⇔ $\mathrm{Halt}(f,f) = \mathrm{yes}$ 矛盾

計算可能性（computability, solvability）

可能／不可能

⇓

高速／低速

計算複雑度（computational complexity）

アルゴリズムの計算量　大きな有限と戦う

並べ替え（ソーティング）問題

入力：$7,\ 15,\ 25,\ 27,\ 9,\ 10,\ 13,\ 19,\ 22,\ 2,\ 17,\ 3,\ 5,\ 14$

出力：$2,\ 3,\ 5,\ 7,\ 9,\ 10,\ 13,\ 14,\ 15,\ 17,\ 19,\ 22,\ 25,\ 27$

アルゴリズム１：　すべての並べ方を試す　$n!$

アルゴリズム２：　最小値を繰り返し探す　n^2

アルゴリズム３：　分割しソートして統合　$n \log n$

問題の複雑度 ——— アルゴリズムの複雑度

計算時間の増大度

入力サイズ	計算時間			
	n	n^2	2^n	$n!$
10	1×10^{-9} 秒	1×10^{-8} 秒	1×10^{-7} 秒	3.6×10^{-4} 秒
20	2×10^{-9} 秒	4×10^{-8} 秒	1×10^{-4} 秒	7.7 年
30	3×10^{-9} 秒	9×10^{-8} 秒	1.1×10^{-1} 秒	8.4×10^{14} 年
40	4×10^{-9} 秒	1.6×10^{-7} 秒	1.8 分	2.6×10^{30} 年
50	5×10^{-9} 秒	2.5×10^{-7} 秒	31 時間	9.6×10^{46} 年
100	1×10^{-8} 秒	1×10^{-6} 秒	4.0×10^{12} 年	3.0×10^{140} 年
1000	1×10^{-7} 秒	1×10^{-4} 秒	.....	.....

毎秒10^{10}回演算できるコンピュータを想定

多項式時間／指数時間

アルゴリズムの理論（2）

計算複雑度

高速／低速

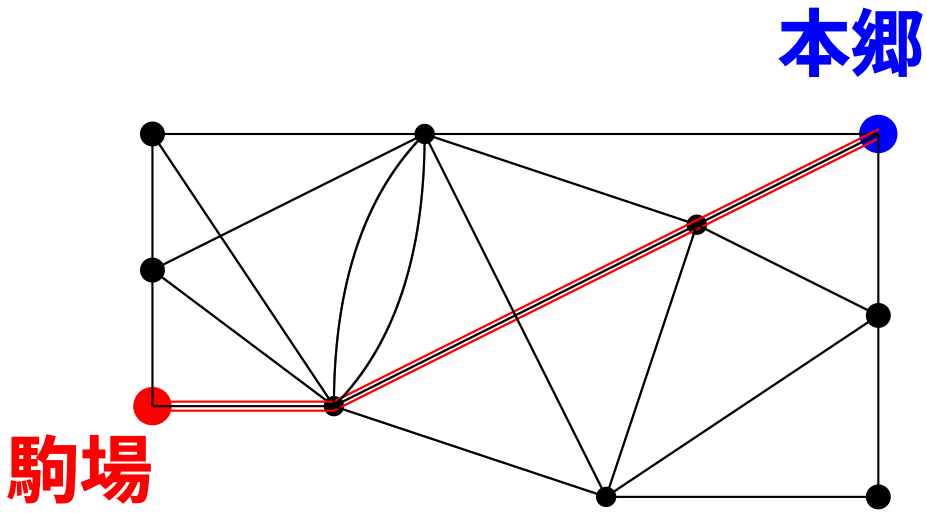
多項式時間／指数時間

1970年頃　**ＮＰ完全性**　(Cook, Levin)

⇒　アルゴリズム設計の枠組み（目標と限界）

クラスP vs クラスNP

問題：本郷から駒場まで 15km 以下の経路があるか？



P = Polynomial

NP = Nondeterministic Polynomial

発見（Find）
vs
確認（Check）

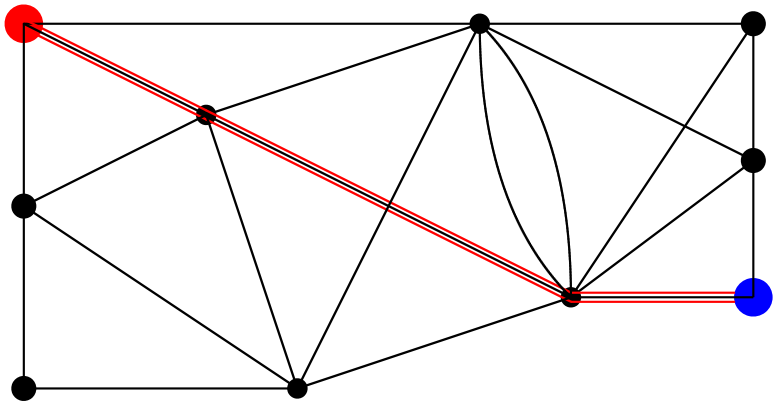
(13.7km)

goo地図： http://map.goo.ne.jp/

‡ 許諾番号： Z07LE第004号

問題：長さが α 以下の経路があるか？

２点間

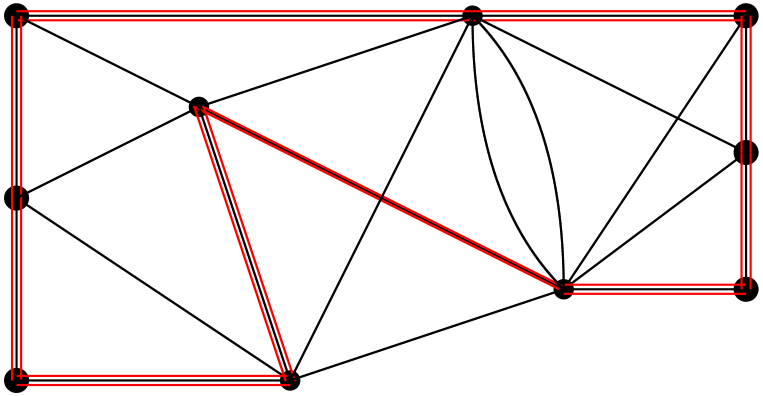


easy to check

easy to find

P

巡回セールスマン

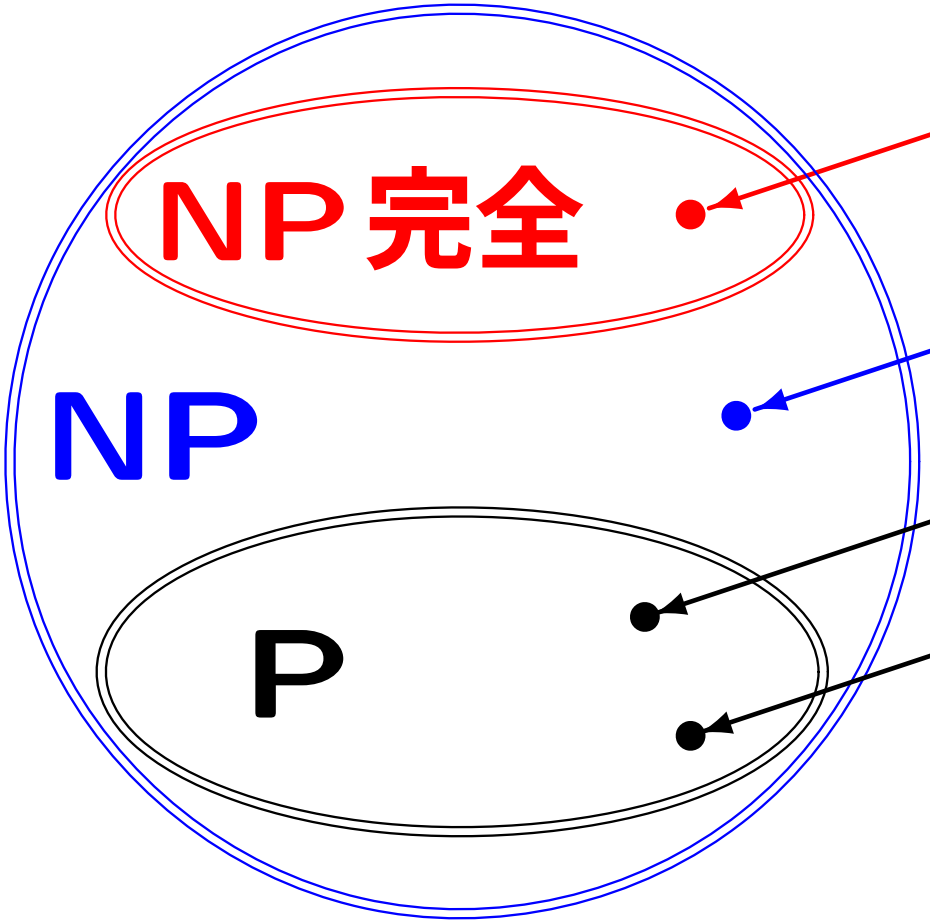


easy to check

difficult to find

NP

NP完全性



巡回セールスマン問題

グラフ同型？

最短路問題

線形計画問題

問題：P＝NP？

Clay 数学研究所のMillennium Problem (1/7)(100万ドル)

http://www.claymath.org/

少し話はかわって……

計算機の発達

そろばん→

←タイガー計算機

コンピュータの誕生

電子計算機：

1946年 ENIAC　(J. W. Mauchly, J.P.Eckert)

東大では：

1958年 PC-1（パラメトロン式）

（高橋秀俊，後藤英一，和田英一）

コンピュータ（ハードウェア）の進歩

ムーアの法則：　２倍／1.5年

２倍／1.5年 ＝ 100倍／10年 ＝ １億倍／40年

１秒で解ける問題のサイズ

演算回数/秒	計算時間 $T(n)$			
C	n	n^2	2^n	$n!$
10^{10}	10^{10}	10^5	33	13
40年 ↓	↓	↓	↓	↓
10^{18}	10^{18}	10^9	60	20

$T(n) = C$ となる n

計算パワーが c 倍になると…

- 多項式時間 n^2 なら　$n \longrightarrow n \cdot \sqrt{c}$
- 指数時間　2^n なら　$n \longrightarrow n + \log c$

教訓１：遅い**アルゴリズム**は
ハードウェアの進歩の恩恵を受けない

教訓２：難しい**問題**は
ハードウェアの進歩を待っても解けない

巡回セールスマン問題 アルゴリズムの進歩

1987年：532都市

（M.Padberg-G. Rinaldi）

n_0=532

1998年：13,509都市

（D.Applegate, et al.）

n_1=13509 (25倍)

Bill Cook
http://www.tsp.gatech.edu/history/pictorial/

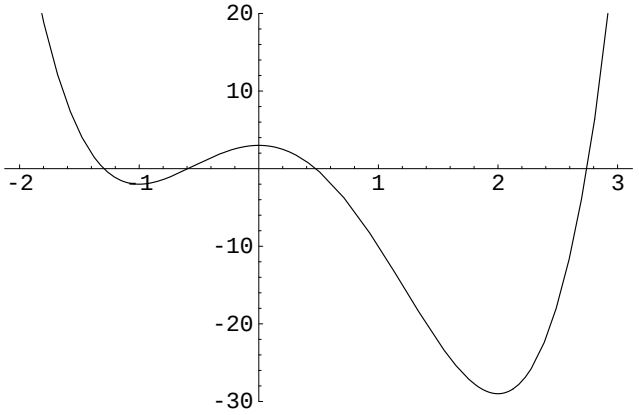
ここまでのまとめ

最適化計算の環境の進歩

- 計算の理論（計算可能性，計算量）
- ハードウェア
- アルゴリズム

2. 最適化の計算とは…

式による計算



$$f(x) = 3x^4 - 4x^3 - 12x^2 + 3$$

$$\Rightarrow \quad f'(x) = 12x^3 - 12x^2 - 24x$$

$$= 12x(x+1)(x-2)$$

$$\Rightarrow \quad f'(x) = 0 \Leftrightarrow x = 0,\ -1,\ 2$$

$$\Rightarrow \quad f(0) = 3,\ f(-1) = -2,\ \boxed{f(2) = -29}$$

$$f(x) = 3x^4 - 4x^3 - 12x^2 + 3 + 0.01x$$

$$\Rightarrow \quad f'(x) = 12x^3 - 12x^2 - 24x + 0.01$$

$$= 12x(x+1)(x-2) + 0.01$$

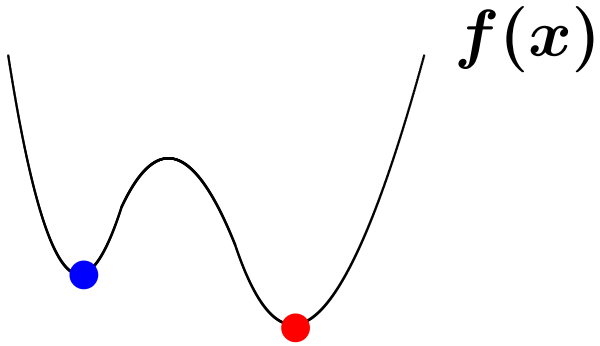
$$\Rightarrow \quad f'(x) = 0 \Leftrightarrow x = 0?,\ -1?,\ 2?$$

$$\Rightarrow \quad f(0?) = 3?,\ f(-1?) = -2?,\ f(2?) = -29?$$

どうする？

数値計算！

近 傍 探 索 法



S0: 初期近似解 x^*

S1: $f(x)$を x^*の「近傍」で最小化 $\Rightarrow$ $x^\bullet$

S2: $f(x^*) \leq f(x^\bullet)$ ならば終了　(x^* は局所最適解)

S3: $x^* = x^\bullet$ と更新してS1へ

最適化の発展　(連続変数)　(復習)

1947年　線形計画　Dantzig

1960年　非線形計画，ニュートン法　Powell, Fletcher

1970年　凸解析・双対理論　Rockafellar

1979年　楕円体法　Khachiyan

1984年　内点法　Karmarkar

1995年　半正定値計画　Alizadeh, Nesterov, Nemirovski

理論：線形／凸／非線形

環境：計算パワーの増大

ニュートン法（基本数値計算アルゴリズム）

テイラー展開（2次近似）：

$$f(x) = f(a) + f'(a)(x-a) + \frac{1}{2}f''(a)(x-a)^2 + \cdots$$
$$\approx \quad C \quad + \quad B\ (x-a) \quad + \quad A \quad (x-a)^2$$

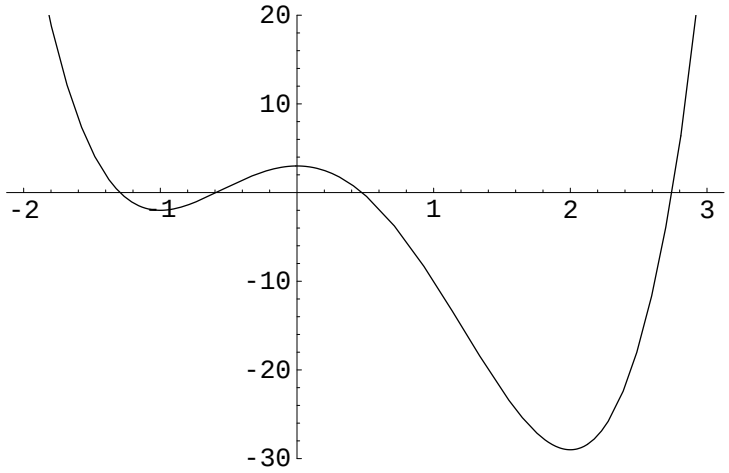
の最小化

$$\Rightarrow \quad x \;=\; a - \frac{B}{2A} \;=\; a - \frac{f'(a)}{f''(a)}$$

$$\Rightarrow \quad x_{k+1} \;=\; x_k - \frac{f'(x_k)}{f''(x_k)}$$

ニュートン法による数値計算

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$



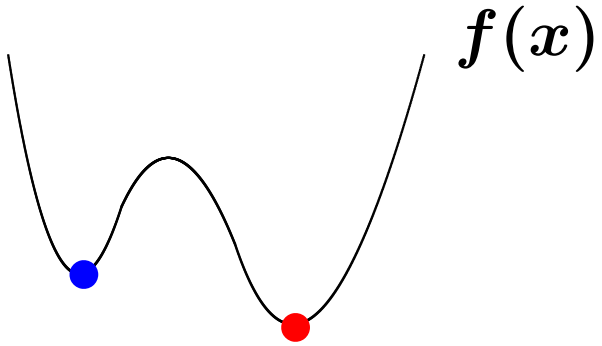
$$f(x) = 3x^4 - 4x^3 - 12x^2 + 3$$

$$\Rightarrow\ x_{k+1} = x_k - \frac{{x_k}^3 - {x_k}^2 - 2x_k}{3{x_k}^2 - 2x_k - 2}$$

k	x_k
0	3.00000
1	2.36842
2	2.07716
3	2.00452

⇒ テイラー展開により
多変数の関数へ拡張

近 傍 探 索 法 ——離散最適化でも——



S0: 初期近似解 x^*

S1: $f(x)$を x^*の「近傍」で最小化 $\Rightarrow$ $x^\bullet$

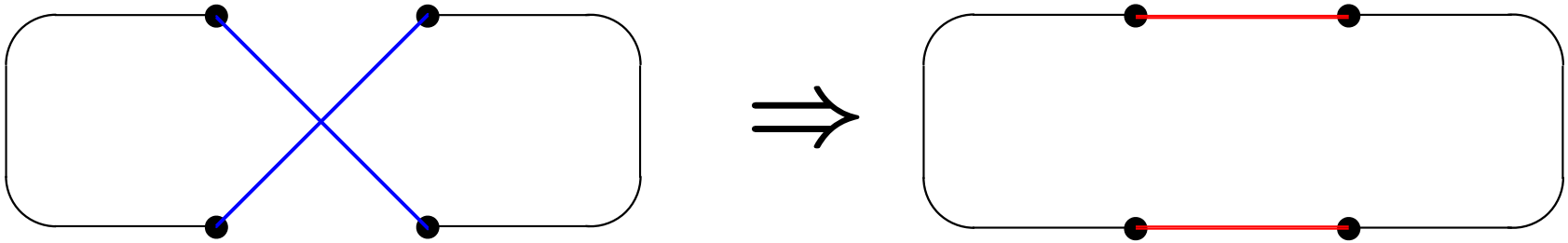
S2: $f(x^*) \leq f(x^\bullet)$ ならば終了　(x^* は局所最適解)

S3: $x^* = x^\bullet$ と更新してS1へ

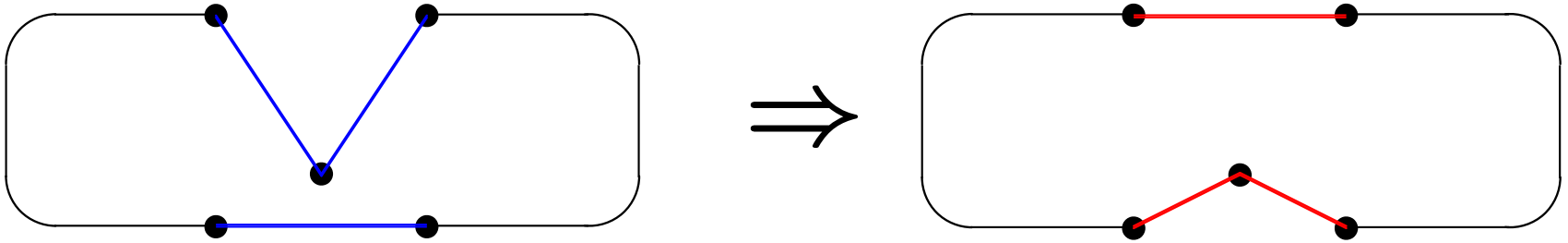
……「近傍」の定義 が問題

巡回セールスマン問題の「近傍」

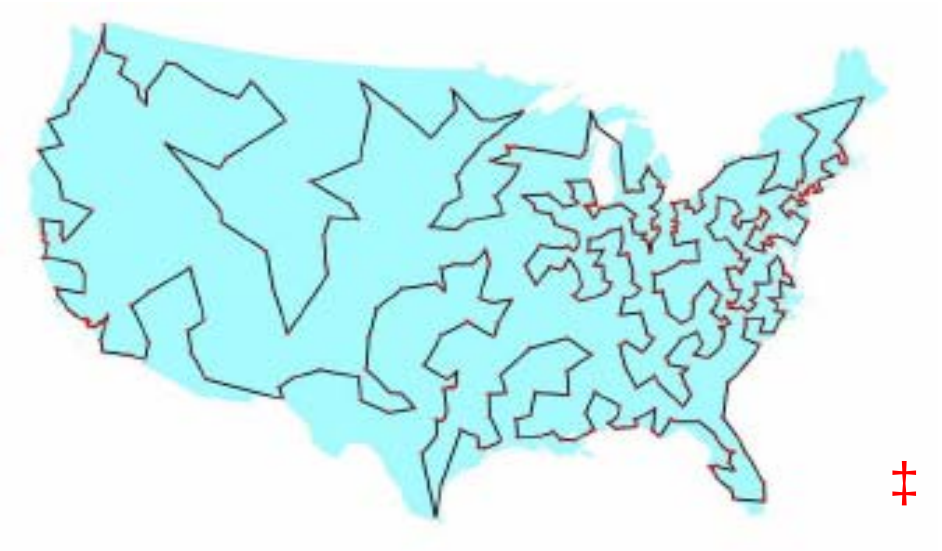
2-opt



Or-opt



巡回セールスマン問題



532 cities, M.Padberg-G.Rinaldi (1987)

http://www.tsp.gatech.edu/history/pictorial/

⇒デモ （計数工学科 土村展之 氏 作成）

最後に 計算量理論と最適化計算の関わりを……

【連続】　　【離散】　（復習）

1947年　――――――線形計画――――――

1970年　凸解析　　多項式性，ＮＰ完全性

双対理論●　　劣モジュラ関数●

1980年　――――――楕円体法――――――

1984年　内点法●●

1995年　半正定値計画●●　　近似アルゴリズム●

2000年　　離散凸解析●●

計算量理論（アルゴリズム）

「最適化の数理」　まとめ

双対定理

線形計画

ルジャンドル変換

凸解析

劣モジュラ

連続

局所最適

離散

近傍探索法

データ
モデル
（AIC）
最適設計

アルゴリズム

計算理論

コンピュータ