計算複雑性にまつわる 10 の誤解

岡本 吉央

電気通信大学 大学院情報理工学研究科 情報・通信工学専攻
okamotoy@uec.ac.jp

2013 年 8 月 8 日
日本オペレーションズ・リサーチ学会
北海道支部サマースクール 2013

効率よく解ける最適化問題とそうでない問題

効率よく解く方法が知られている

- 最小全域木問題
- 最短路問題
- 最大流問題
- 最小費用流問題
- 線形計画問題
- 凸二次計画問題
- ...

効率よく解く方法が知られていない

- 巡回セールスマン問題
- 集合被覆問題
- 集合分割問題
- ナップサック問題
- 整数計画問題
- 凹二次計画問題
- ...

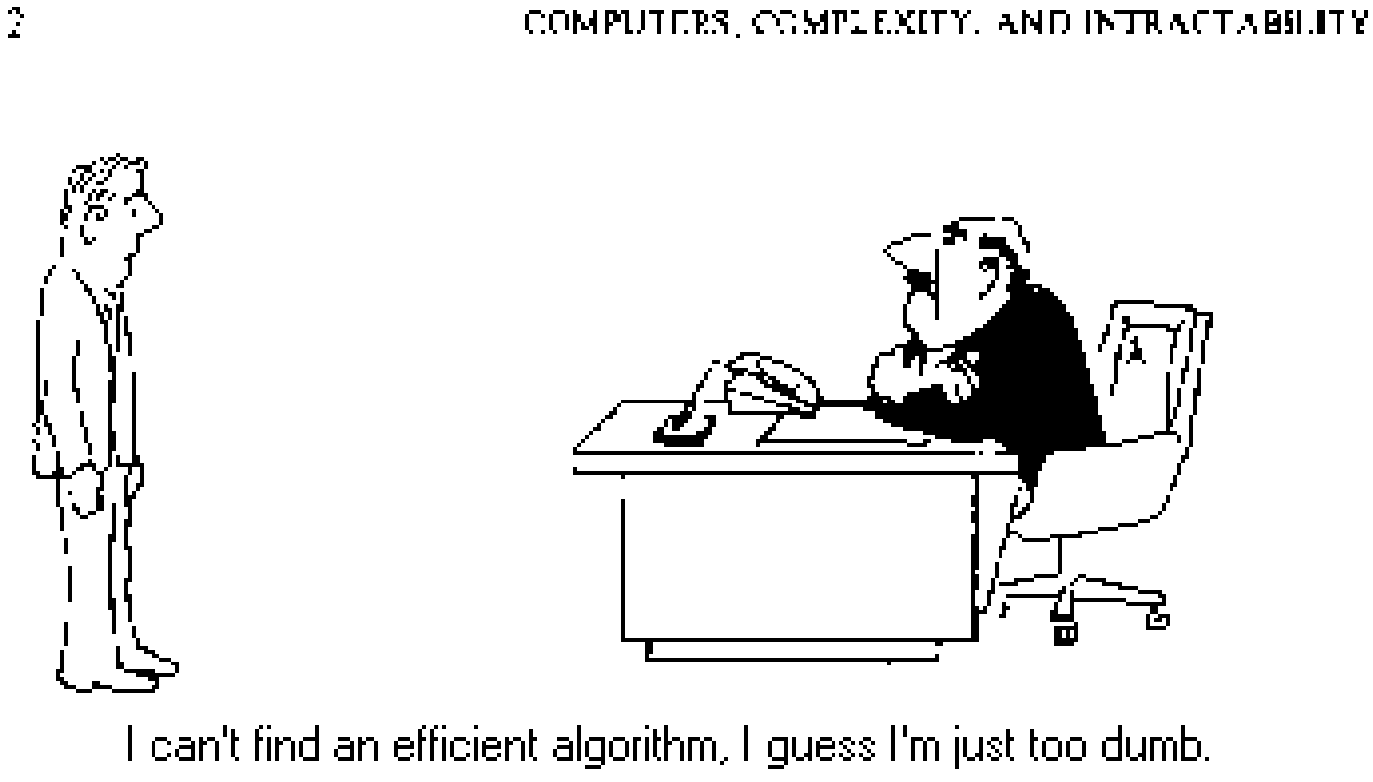
効率よく $=$ 多項式時間で

疑問

このように分かれるのはどうしてか？ $\rightsquigarrow$ 計算複雑性理論

Garey & Johnson '79 (1)

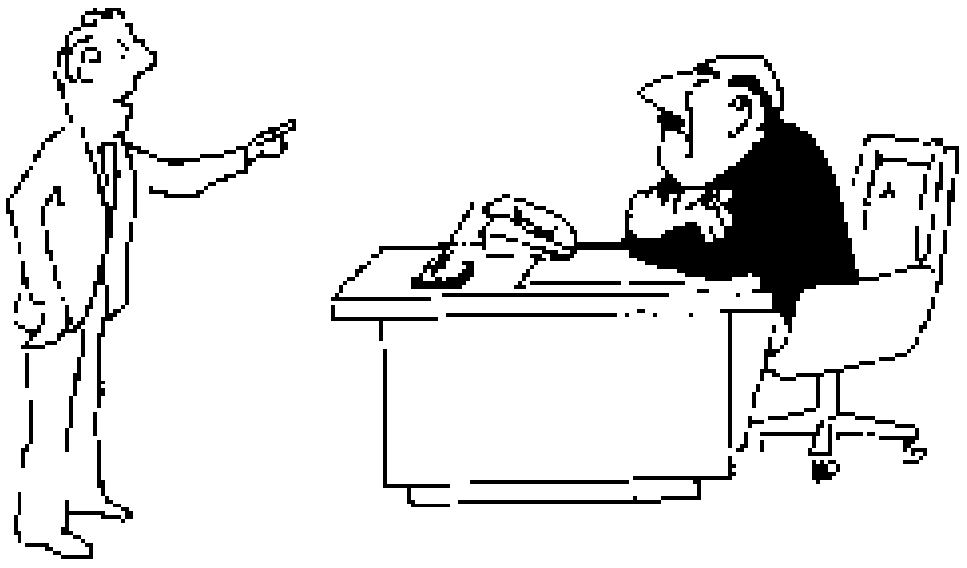
「効率よいアルゴリズムが見つからない」



「じゃあ，クビね」

Garey & Johnson '79 (2)

理想：「効率よいアルゴリズムは存在しないんです！」

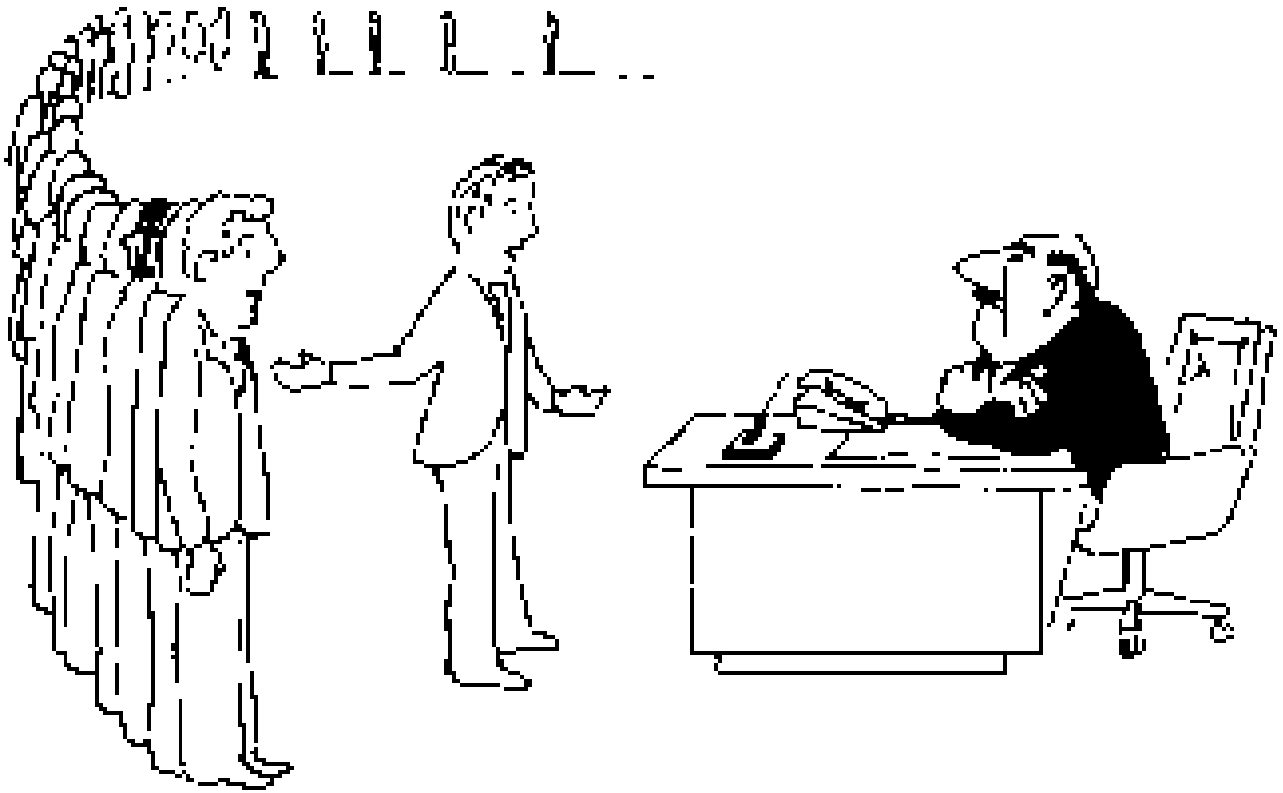


I can't find an efficient algorithm, because no such algorithm is possible

「……」

Garey & Johnson '79 (3)

現状：「ここにいる有名人でも見つけられないんです…」



I can't find an efficient algorithm, but neither can all these famous people.

「う〜〜ん」

クレイ数学研究所ミレニアム問題

重要な問題

P vs NP

誤解 1

「P vs NP 問題」は
多項式時間で解ける問題と指数関数時間で解ける問題のクラスが
同じであるかを問う

クレイ数学研究所ミレニアム問題

重要な問題

P vs NP

誤解 1

「P vs NP 問題」は
多項式時間で解ける問題と指数関数時間で解ける問題のクラスが同じであるかを問う

問題のクラス：クラス P，NP，EXP

クラス P

判定問題が P に属するとは，
それを多項式時間で解く決定性アルゴリズムが存在すること

クラス NP

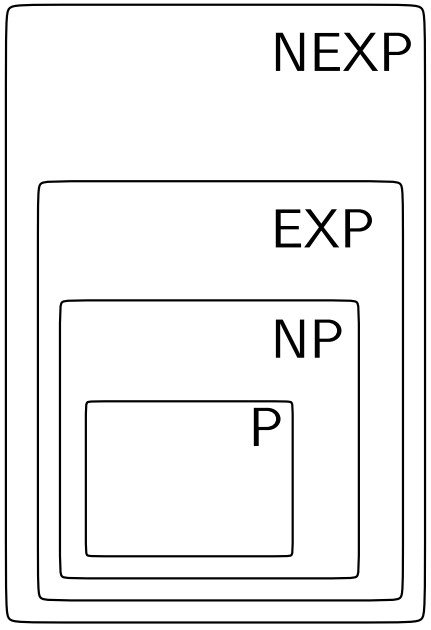
判定問題が NP に属するとは，
それを多項式時間で解く非決定性アルゴリズムが存在すること

クラス EXP

判定問題が EXP に属するとは，
それを指数関数時間で解く決定性アルゴリズムが存在すること

問題のクラス

	決定性	非決定性
多項式時間	P	NP
指数関数時間	EXP	NEXP



事実

- $\mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{EXP}$
- $\mathbf{P} \neq \mathbf{EXP}$ (**時間階層定理** (Stearn, Hartmanis '65) **の帰結**)

つまり,「多項式時間」と「指数関数時間」が異なることは既知

クラス NP：別の定義

クラス NP

判定問題が NP に属するとは，
それを多項式時間で解く非決定性アルゴリズムが存在すること

クラス NP：別の定義

判定問題が NP に属するとは，
その問題に対する答えが Yes であるとき，
Yes であるための証拠が存在して，
それを多項式時間で検証する決定性アルゴリズムが
存在すること

この 2 つの定義は同値

例：ナップサック問題

問題：収入を 5 万円以上にするような積み方はあるか？

商品	1	2	3	4
収入 [万円]	3	4	1	2
重さ [kg]	2	3	1	3

重量制限： 4 [kg]

答え：Yes

証拠：商品 2 と 3 を積む

(多項式時間決定性アルゴリズムで検証可能)

よって，ナップサック問題 $\in$ NP

クラス NP：別の定義 (再掲)

クラス NP

判定問題が NP に属するとは，
それを多項式時間で解く非決定性アルゴリズムが存在すること

クラス NP：別の定義

判定問題が NP に属するとは，
その問題に対する答えが Yes であるとき，
Yes であるための証拠が存在して，
それを多項式時間で検証する決定性アルゴリズムが
存在すること

この 2 つの定義は同値

真実

「P vs NP」の問題は
多項式時間で解けることと検証できることの差を問う

効率よく解ける最適化問題とそうでない問題

効率よく解く方法が知られている

- 最小全域木問題
- 最短路問題
- 最大流問題
- 最小費用流問題
- 線形計画問題
- 凸二次計画問題
- ...

クラス P に属する

効率よく解く方法が知られていない

- 巡回セールスマン問題
- 集合被覆問題
- 集合分割問題
- ナップサック問題
- 整数計画問題
- 凹二次計画問題
- ...

クラス NP に属する

(判定問題版を考えたとき)

P vs NP 問題

クラス P

判定問題が P に属するとは，
それを多項式時間で解く決定性アルゴリズムが存在すること

クラス NP

判定問題が NP に属するとは，
それを多項式時間で解く非決定性アルゴリズムが存在すること

誤解 2

専門家は皆

$$P \neq NP$$

であると信じている

P vs NP 問題

クラス P

判定問題が P に属するとは，
それを多項式時間で解く決定性アルゴリズムが存在すること

クラス NP

判定問題が NP に属するとは，
それを多項式時間で解く非決定性アルゴリズムが存在すること

誤解 2

専門家は皆

$$P \neq NP$$

であると信じている

P =? NP Open Poll (Gasarch, '02)

100 人の専門家へのアンケート結果

P vs NP 問題はどのように決着するでしょうか？

❶ P ≠ NP	61 人
❷ P = NP	9 人
❸ ZFC と独立	4 人
❹ 原始再帰算術と独立ではない	3 人
❺ モデルに依存	1 人
❻ 分からない	22 人

「P ≠ NP」だと答えた人も確証があるわけではない

Second **P** =? **NP** Open Poll (Gasarch, '12)

152 人の専門家へのアンケート結果

P vs NP 問題はどのように決着するでしょうか？

❶ **P** $\neq$ **NP**	126 人 (83%)
❷ **P** $=$ **NP**	12 人 (9%)
❸ ZFC と独立	5 人 (3%)
❹ 関心がない	5 人 (3%)
❺ 分からない	1 人 (0.6%)
❻ 分からないし，関心もない	1 人 (0.6%)

「**P** $\neq$ **NP**」だと答えた人も確証があるわけではない

効率よく解ける最適化問題とそうでない問題

効率よく解く方法が知られている

- 最小全域木問題
- 最短路問題
- 最大流問題
- 最小費用流問題
- 線形計画問題
- 凸二次計画問題
- ...

クラス P に属する

効率よく解く方法が知られていない

- 巡回セールスマン問題
- 集合被覆問題
- 集合分割問題
- ナップサック問題
- 整数計画問題
- 凹二次計画問題
- ...

クラス NP に属する
特に，NP 完全である

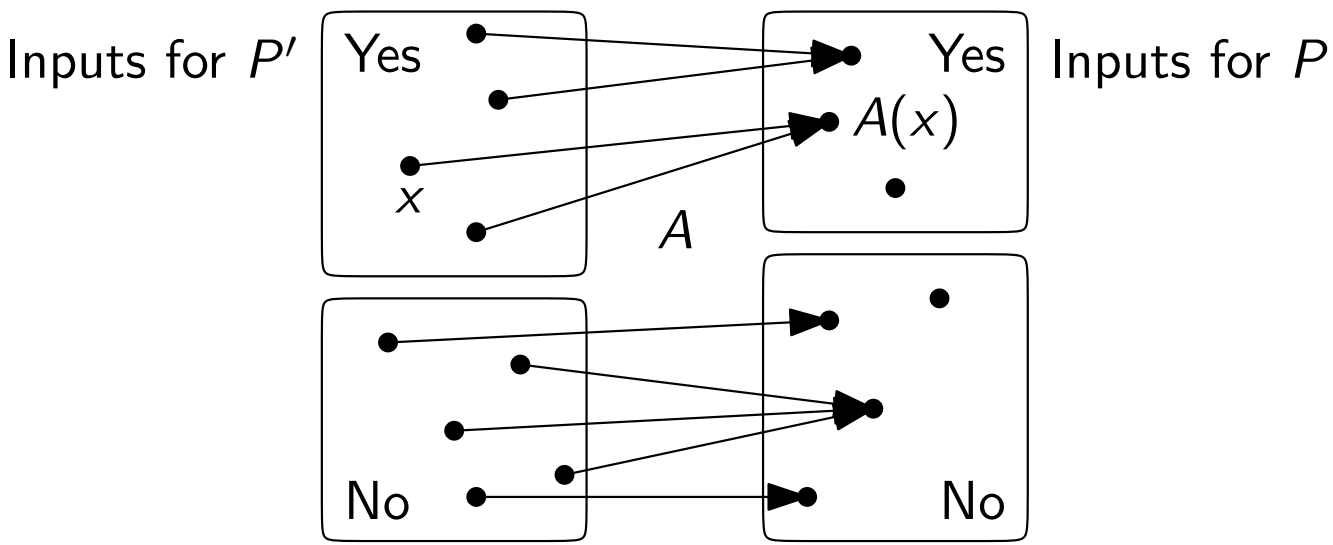
(判定問題版を考えたとき)

NP 完全問題

NP 完全問題とは？

NP に属する問題 P が NP 完全であるとは，
NP に属する任意の問題 P' に対して，
次のような アルゴリズム A が存在すること

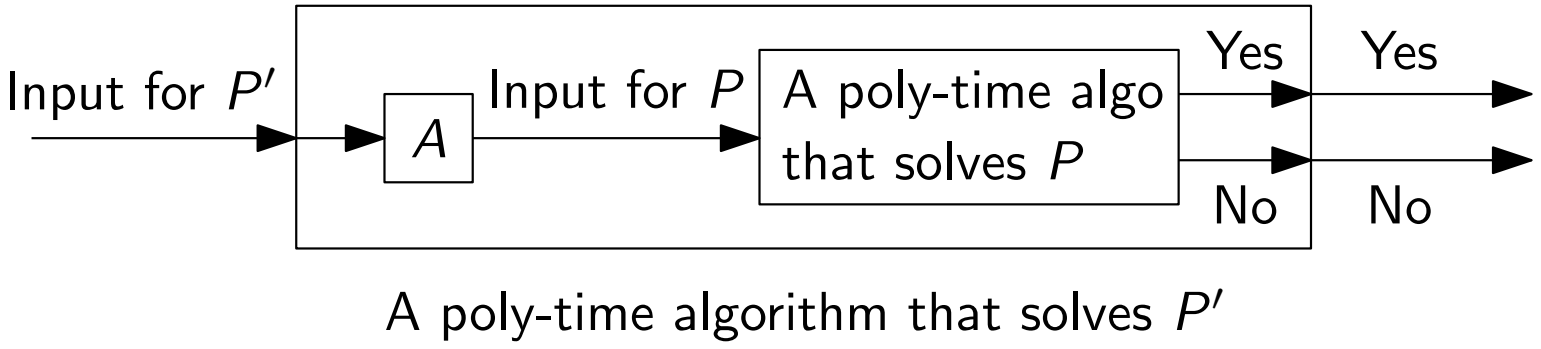
- $\forall$ P' の入力 x : $A(x)$ は P の入力を多項式時間で構成
- P' の入力 x に対する出力が Yes $\Leftrightarrow$ P の入力 $A(x)$ に対する出力が Yes



このような A のことを多対一多項式時間帰着と呼ぶ

重要な性質

P が NP 完全 かつ P が多項式時間で解ける $\Rightarrow$
NP に属するすべての問題が多項式時間で解ける



A poly-time algorithm that solves P'

直感

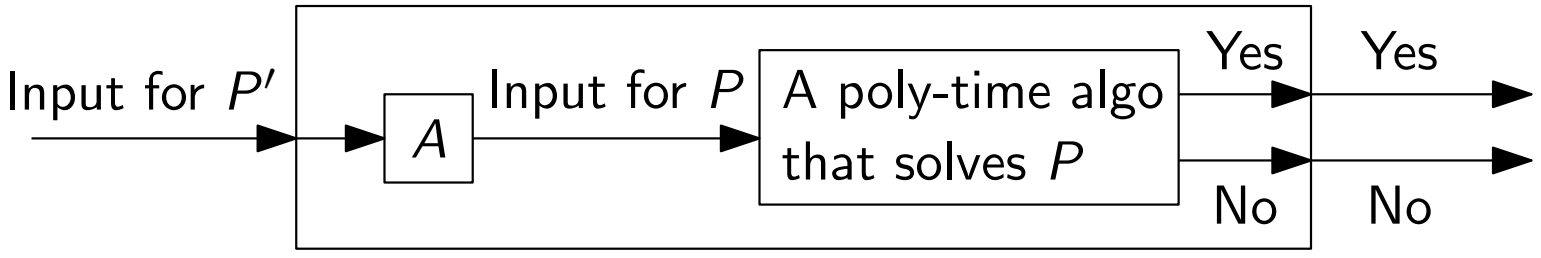
NP 完全問題は NP に属する問題の中で最も難しい問題

NP 完全問題：別の定義？

NP 完全問題を次のように定義したらどうなるか？

問題 P が NP 完全であるとは，次の性質を満たすこと

- **P が多項式時間で解ける $\Rightarrow$**
  NP に属するすべての問題が多項式時間で解ける



A poly-time algorithm that solves P'

誤解 3

この 2 つの NP 完全性の定義は同値である

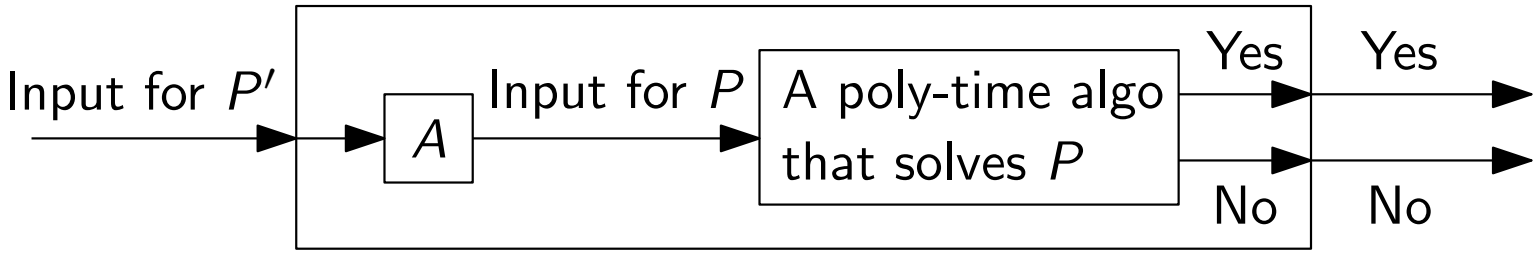
事実：この 2 つの定義が同値であるかどうかは未解決

NP 完全問題：別の定義？

NP 完全問題を次のように定義したらどうなるか？

問題 P が NP 完全であるとは，次の性質を満たすこと

- **P が多項式時間で解ける $\Rightarrow$**
  NP に属するすべての問題が多項式時間で解ける



A poly-time algorithm that solves P'

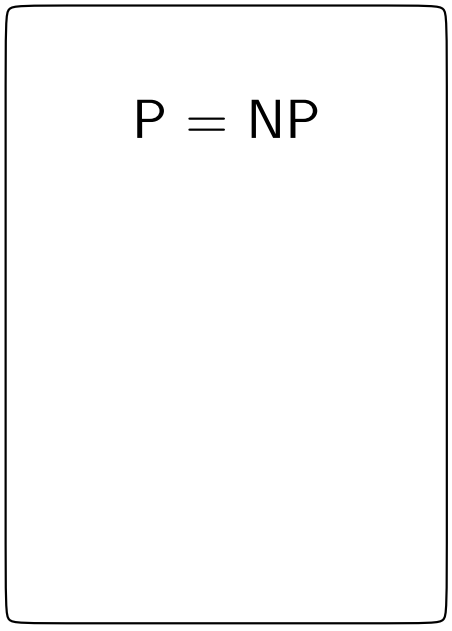
誤解 3

この 2 つの NP 完全性の定義は同値である

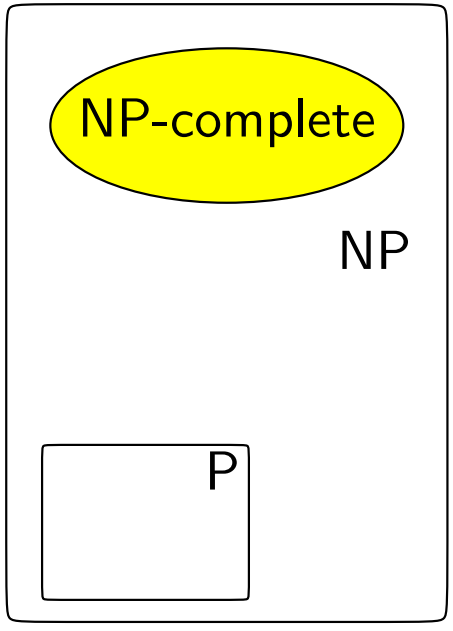
事実：この 2 つの定義が同値であるかどうかは未解決

P vs NP 問題と NP 完全性

P = NP の場合



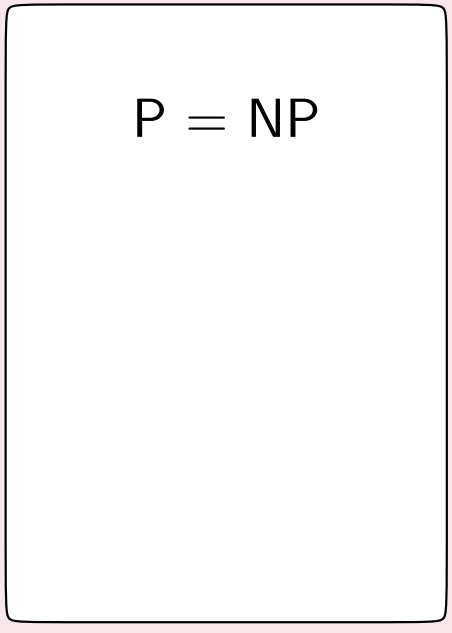
P ≠ NP の場合



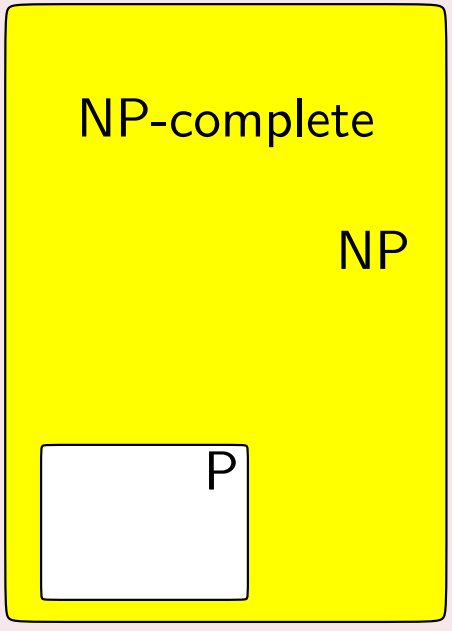
P vs NP 問題と NP 完全性：誤解

誤解 4

P = NP の場合



P ≠ NP の場合

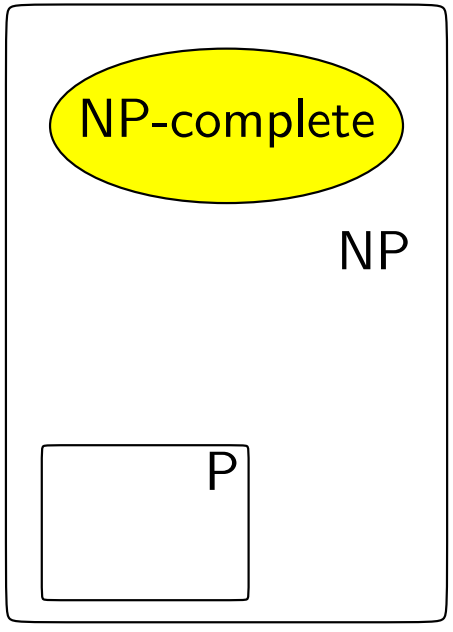


$P \in \mathbf{NP} - \mathbf{P}$
$\Rightarrow P$ **は** NP **完全**

Ladner の定理

事実 (Ladner '75)

P $\neq$ NP $\Rightarrow$
NP $-$ P に属するが NP 完全ではない問題が存在する



未解決問題：そのような問題で「自然」なものは存在するか？

P vs NP 問題の (非) 重要性 (1)

重要性に関する主張

現代暗号はある種の問題が計算複雑性の意味で難しいという仮定に依拠している

この主張は正しいが，次は正しくない

誤解 5

現代暗号は NP 完全問題をもとにして作られている

P vs NP 問題の (非) 重要性 (1)

重要性に関する主張

現代暗号はある種の問題が計算複雑性の意味で難しいという仮定に依拠している

この主張は正しいが，次は正しくない

誤解 5

現代暗号は NP 完全問題をもとにして作られている

NP 完全問題をもとにして作られた暗号 (例)

- Merkle–Hellman **のナップサック暗号系** '78
  - **破られる** (Shamir '84)
- **その後も**
  ナップサック問題 (とその変種) に基づく多くの暗号系
  - **ほとんど破られている**

なぜ破られるのか？

- NP 完全性は「最悪時」の困難性に関する概念
- 暗号は「平均時」や「ほとんどすべて」の困難性が必要

ナップサック問題は「平均時」に簡単な問題 (のようである)

NP 完全問題をもとにして作られた暗号 (例)

- Merkle–Hellman **のナップサック暗号系** '78
  - **破られる** (Shamir '84)
- **その後も**
  ナップサック問題 (とその変種) に基づく多くの暗号系
  - **ほとんど破られている**

なぜ破られるのか？

- **NP 完全性は「最悪時」の困難性に関する概念**
- **暗号は「平均時」や「ほとんどすべて」の困難性が必要**

ナップサック問題は「平均時」に簡単な問題 (のようである)

実際の暗号系は…

より強い (かもしれない) 困難性の仮定に依拠して作られている

- 一方向性関数の存在性
- (暗号理論的に安全な) 疑似乱数生成器の存在性

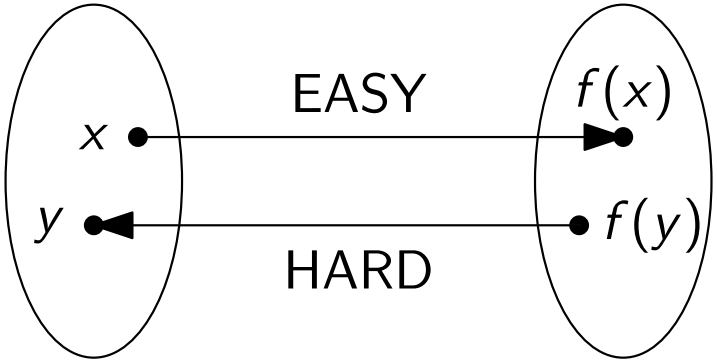
事実

- この 2 つは同値
- この 2 つが正しい $\Rightarrow$ P $\neq$ NP

補足：一方向性関数とは？

ビット列をビット列に写す関数 f が**一方向性関数**であるとは，

- 任意の列 x に対して，$f(x)$ が多項式時間で計算できる
- 任意の列 x と任意の多項式時間アルゴリズム A に対して，A が $f(x)$ から x を当てる確率が超多項式的に小さい



未解決問題

- 一方向性関数が存在するか？
- $\mathrm{P} \neq \mathrm{NP} \Rightarrow$ 一方向性関数が存在する

P vs NP 問題の (非) 重要性 (2)

非重要性に関する主張

並列計算によって難しい問題が解けるようになってきている

誤解 6

並列計算を用いると，NP 完全問題が多項式時間で解ける

「並列計算を用いると」ということばの厳密な意味に依存するが

- 多項式個のプロセッサを使い，同期するクロックを持ち，個別メモリも共有メモリも持つ並列コンピュータを考えても，NP 完全問題が多項式時間で解けるかどうか分からない

そして，解けるとすると，$\mathrm{P} = \mathrm{NP}$ となる

P vs NP 問題の (非) 重要性 (2)

非重要性に関する主張

並列計算によって難しい問題が解けるようになってきている

誤解 6

並列計算を用いると，NP 完全問題が多項式時間で解ける

「並列計算を用いると」ということばの厳密な意味に依存するが

- 多項式個のプロセッサを使い，同期するクロックを持ち，個別メモリも共有メモリも持つ並列コンピュータを考えても，NP 完全問題が多項式時間で解けるかどうか分からない

そして，解けるとすると，P = NP となる

P vs NP 問題の (非) 重要性 (2)

非重要性に関する主張

並列計算によって難しい問題が解けるようになってきている

誤解 6

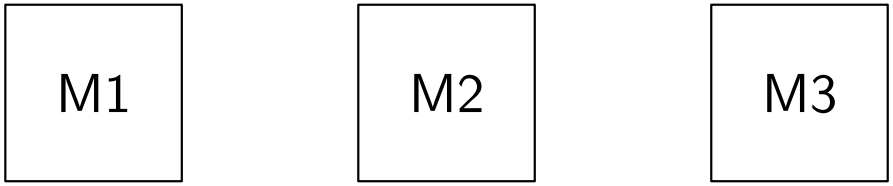
並列計算を用いると，NP 完全問題が多項式時間で解ける

「並列計算を用いると」ということばの厳密な意味に依存するが

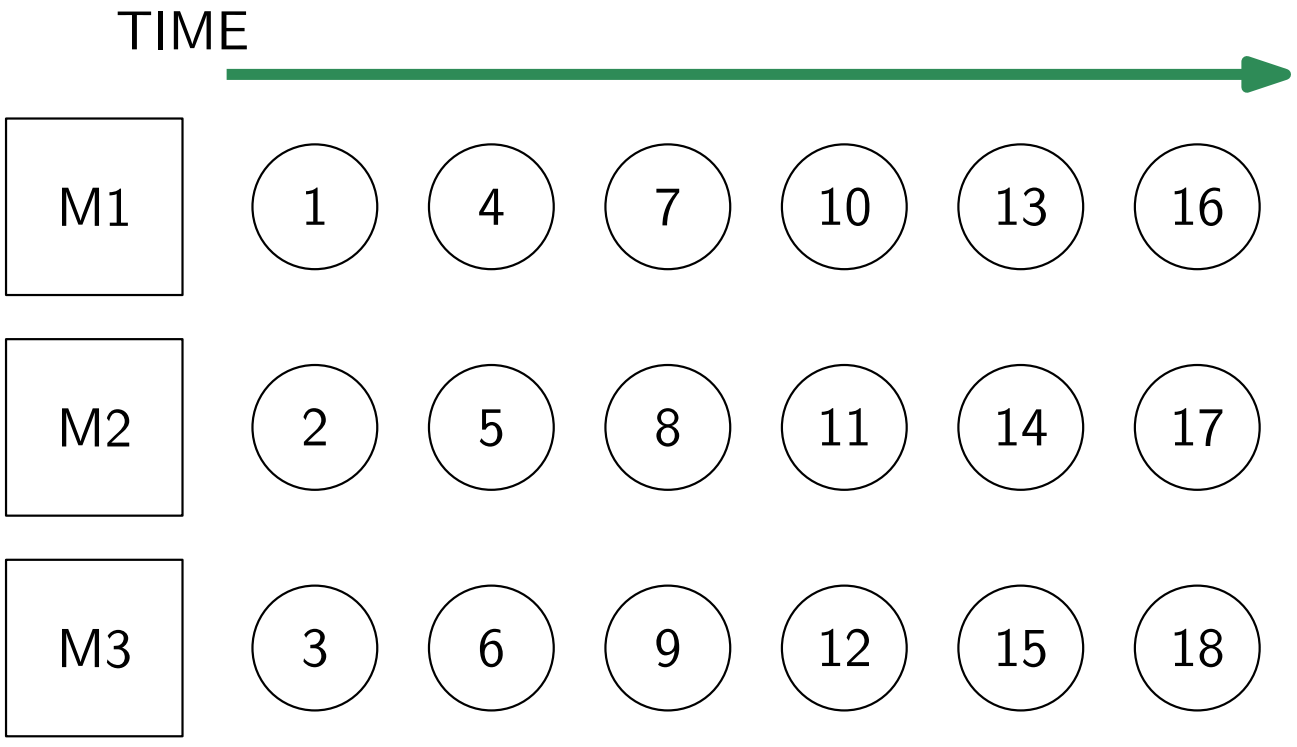
- 多項式個のプロセッサを使い，同期するクロックを持ち，個別メモリも共有メモリも持つ並列コンピュータを考えても，NP 完全問題が多項式時間で解けるかどうか分からない

そして，解けるとすると，$\mathrm{P} = \mathrm{NP}$ となる

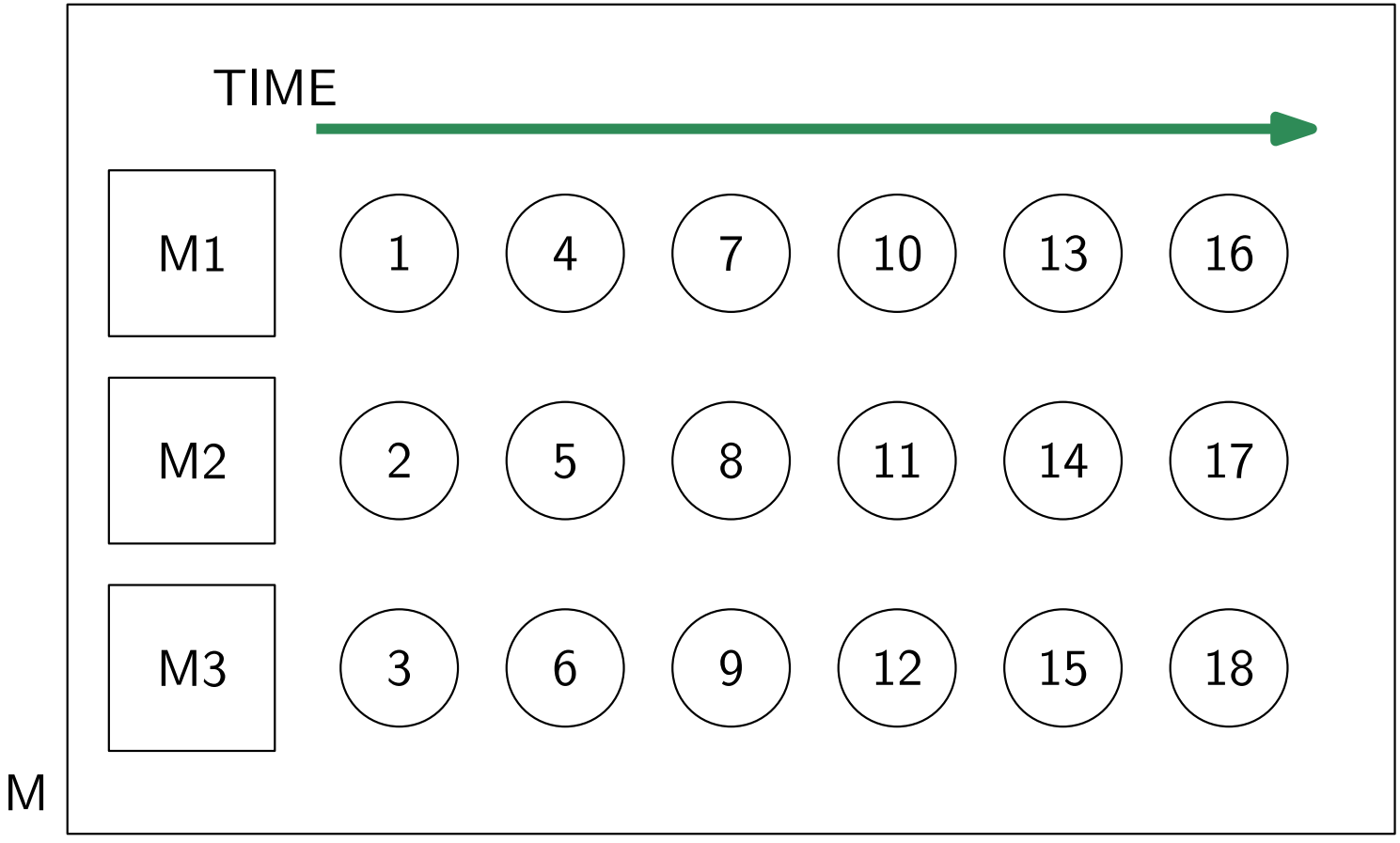
並列計算の模倣



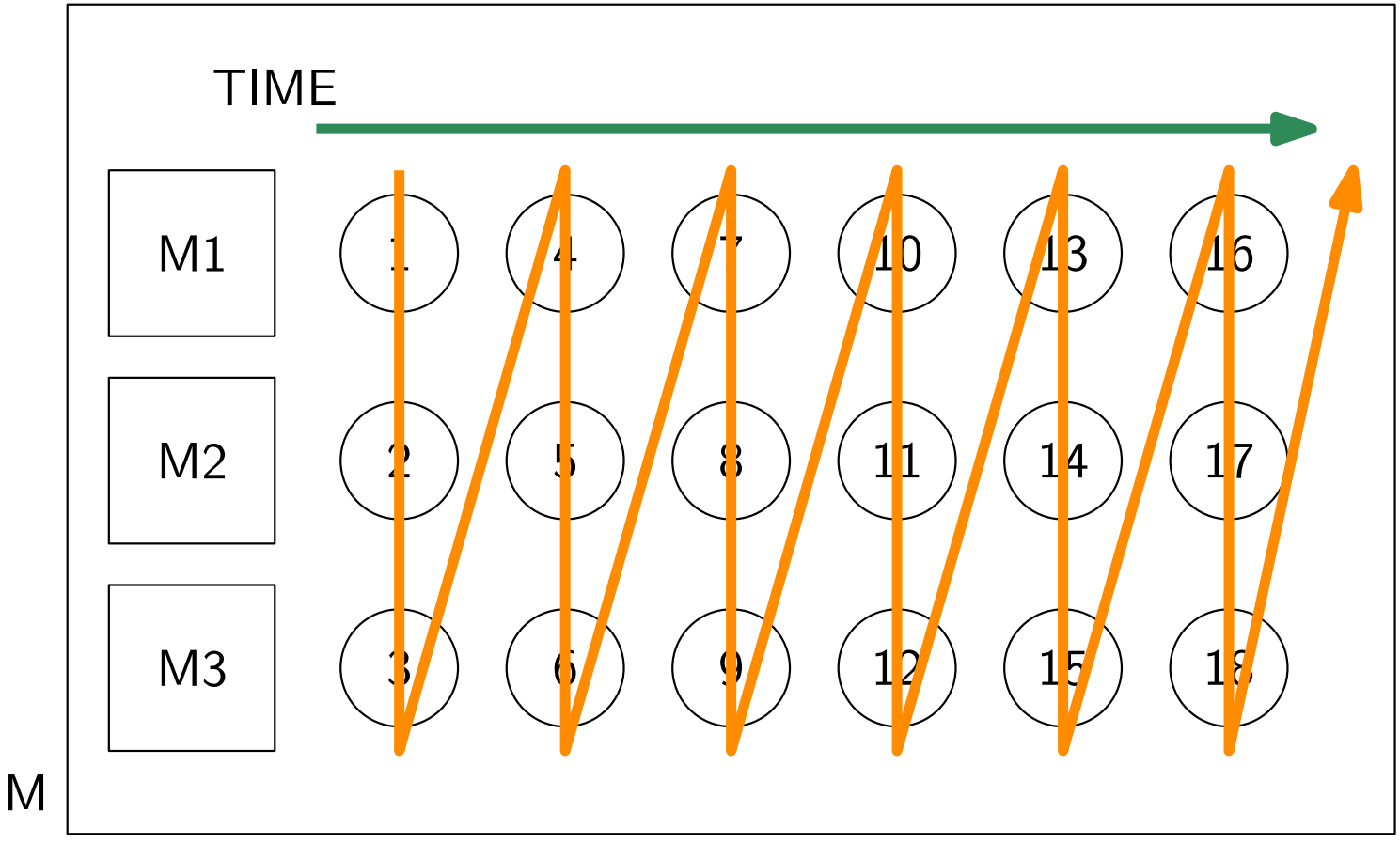
並列計算の模倣



並列計算の模倣



並列計算の模倣



並列計算とクラス P

疑問

並列計算によって，クラス P に属する問題を
多項式時間よりも高速に解くことはできるのだろうか？

クラス NC

判定問題がクラス NC に属するとは，
多項式個のプロセッサを使い，それを対数多項式時間で解く
決定性アルゴリズムが存在すること

対数多項式：$\log n, \log^2 n, \log^3 n, \ldots$

- 事実：NC $\subseteq$ P
- 未解決：NC $\neq$ P

NP 完全問題のように P 完全問題が多く知られている

P vs NP 問題の (非) 重要性 (3)

非重要性に関する主張

量子コンピュータが実現されれば，難しい問題も簡単に解けるようになる

誤解 7

量子計算によって NP 完全問題が多項式時間で解ける

事実

- 量子計算で NP 完全問題が多項式時間で解けるかは未解決

ただし，量子計算によって計算量が削減される例は多く存在する

P vs NP 問題の (非) 重要性 (3)

非重要性に関する主張

量子コンピュータが実現されれば，難しい問題も簡単に解けるようになる

誤解 7

量子計算によって NP 完全問題が多項式時間で解ける

事実

- 量子計算で NP 完全問題が多項式時間で解けるかは未解決

ただし，量子計算によって計算量が削減される例は多く存在する

量子計算によって計算量が削減される例

素因数分解

- **古典：多項式時間アルゴリズムが知られていない**
  (NP 困難であるかどうかも知られていない)
- **量子：** $O(\log^3 N)$ (Shor '97)

探索問題

- **古典：** n
- **量子：** $O(\sqrt{n})$ (Grover '96)

これらは NP 完全問題を多項式時間で解いているわけではない

P vs NP 問題の (非) 重要性 (4)

非重要性に関する主張

NP 完全問題も現実的には解くことができる

「現実的」とは？

- **平均的，あるいは，ほとんどすべての入力に対して**
  (cf. **暗号系**)
- **近似的**

近似アルゴリズムと数理計画法

現代的な多くの近似アルゴリズムは線形計画法や半正定値計画法を用いている

線形計画問題

$$\begin{aligned} \text{minimize} \quad & c \cdot x \\ \text{subject to} \quad & a_i \cdot x = b_i, i = 1, \ldots, m, \\ & x \geq 0 \end{aligned}$$

線形計画問題の利用法 (例)

1. 与えられた問題を整数計画問題として記述する
2. それの線形計画緩和を解く
3. 得られた最適解を丸めて，元の問題の許容解を得る

事実

線形計画問題は多項式時間で解ける

近似アルゴリズムと半正定値計画法

現代的な多くの近似アルゴリズムは線形計画法や半正定値計画法を用いている

半正定値計画問題

$$\begin{aligned} \text{minimize} \quad & C \bullet X \\ \text{subject to} \quad & A_i \bullet X = b_i, i = 1, \ldots, m, \\ & X \text{ は対称半正定値} \end{aligned}$$

半正定値問題の利用法 (例)

1. 与えられた問題を整数計画問題として記述する
2. それの半正定値計画緩和を解く
3. 得られた最適解を丸めて，元の問題の許容解を得る

誤解 8

半正定値計画問題は多項式時間で解ける

近似アルゴリズムと半正定値計画法

現代的な多くの近似アルゴリズムは線形計画法や半正定値計画法を用いている

半正定値計画問題

$$\begin{aligned} \text{minimize} \quad & C \bullet X \\ \text{subject to} \quad & A_i \bullet X = b_i, i = 1, \ldots, m, \\ & X \text{ は対称半正定値} \end{aligned}$$

半正定値問題の利用法 (例)

1. 与えられた問題を整数計画問題として記述する
2. それの半正定値計画緩和を解く
3. 得られた最適解を丸めて，元の問題の許容解を得る

誤解 8

半正定値計画問題は多項式時間で解ける

半正定値計画法に対する多項式時間アルゴリズム？

現代的な多くの近似アルゴリズムは線形計画法や半正定値計画法を用いている

半正定値計画問題

$$\begin{aligned} \text{minimize} \quad & C \bullet X \\ \text{subject to} \quad & A_i \bullet X = b_i, i = 1, \ldots, m, \\ & X \text{ は対称半正定値} \end{aligned}$$

誤解 8

半正定値計画問題は多項式時間で解ける

事実

- 半正定値計画問題が多項式時間で解けると分かっているのは，入力がある種の仮定を満たすときのみ
  - 許容領域の有界性，内点許容解の存在性 (内点法)，…
- 「解ける」といっても，ϵ 近似が多項式時間で得られるのみ

近似アルゴリズムと局所探索

現代的な多くの近似アルゴリズムは局所探索法を用いている

誤解 9

局所探索は多項式ステップで終了する

事実

- 多くの問題に対して，最悪の場合，
  局所探索が多項式ステップで終了しない
- 局所最適解を見つけることが難しい，
  という計算複雑性による裏付けを持つ問題も多い

近似アルゴリズムと局所探索

現代的な多くの近似アルゴリズムは局所探索法を用いている

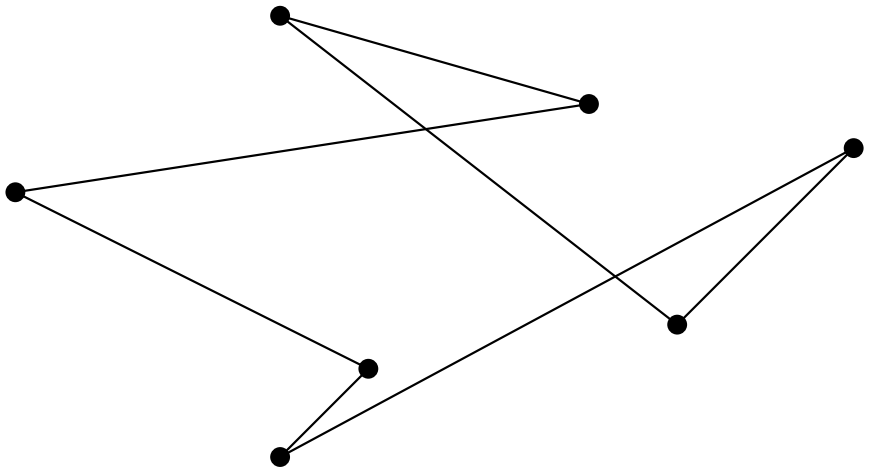
誤解 9

局所探索は多項式ステップで終了する

事実

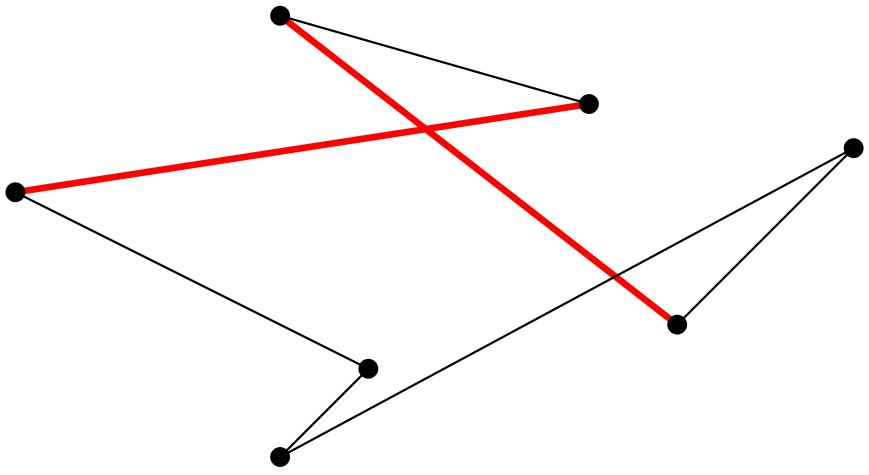
- 多くの問題に対して，最悪の場合，
  局所探索が多項式ステップで終了しない
- 局所最適解を見つけることが難しい，
  という計算複雑性による裏付けを持つ問題も多い

巡回セールスマン問題と 2opt 近傍



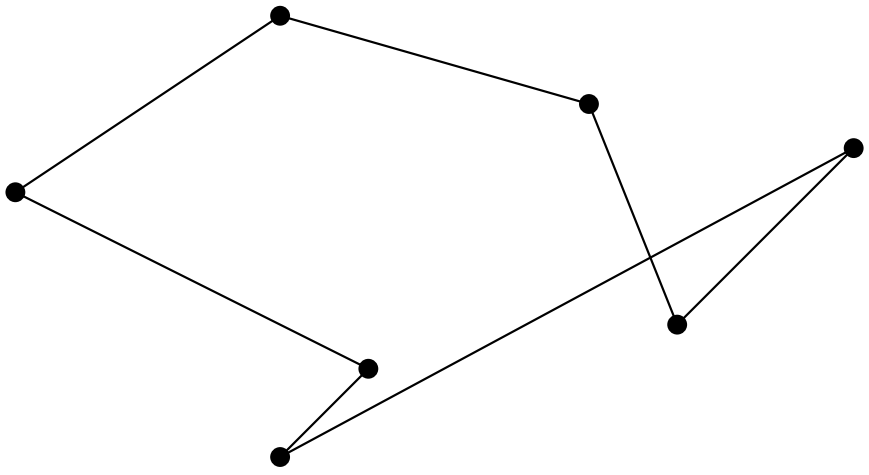
局所最適解

巡回セールスマン問題と 2opt 近傍



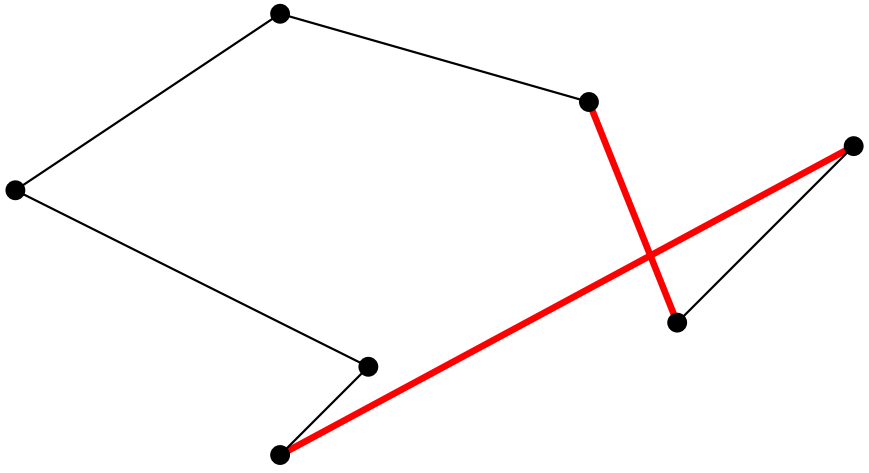
局所最適解

巡回セールスマン問題と 2opt 近傍



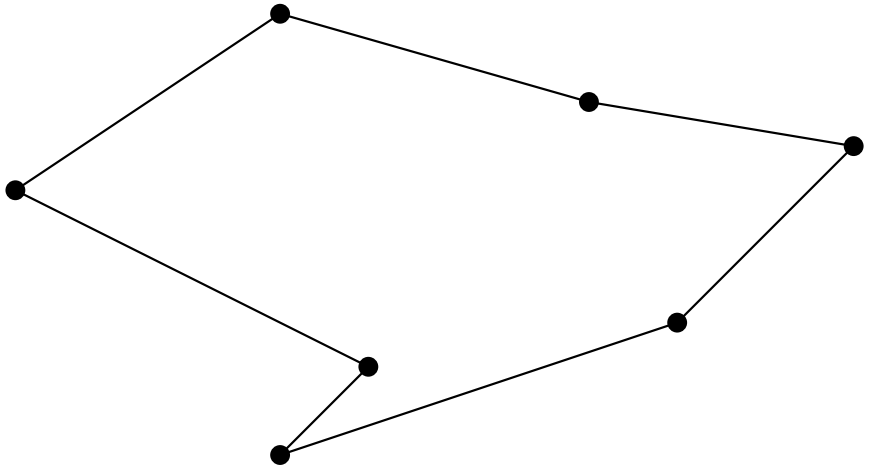
局所最適解

巡回セールスマン問題と 2opt 近傍



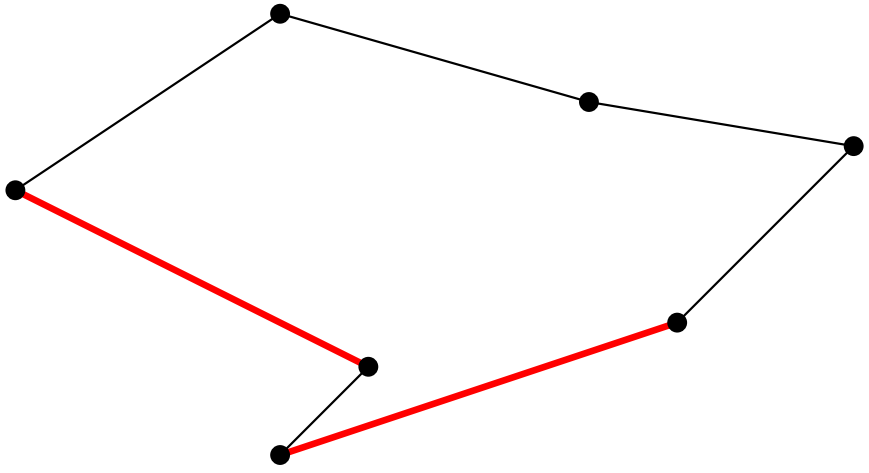
局所最適解

巡回セールスマン問題と 2opt 近傍



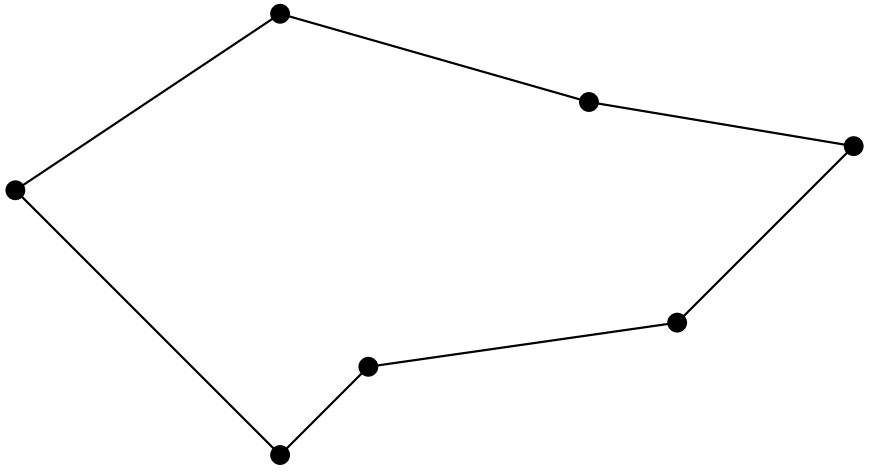
局所最適解

巡回セールスマン問題と 2opt 近傍



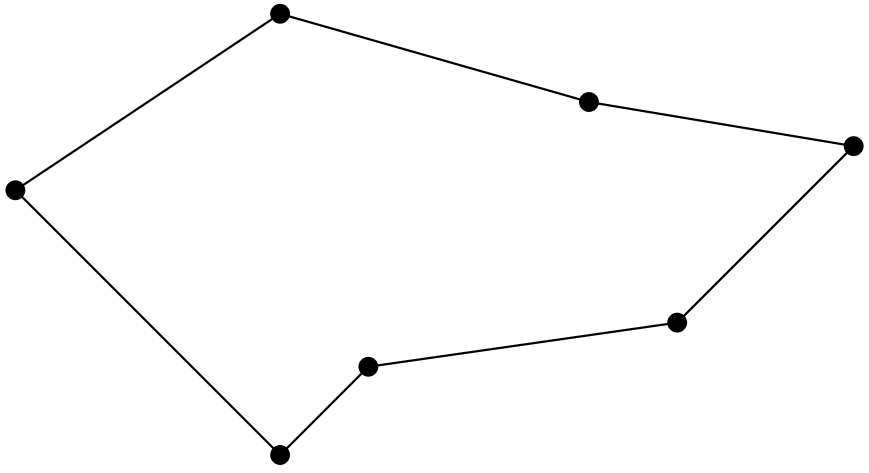
局所最適解

巡回セールスマン問題と 2opt 近傍



局所最適解

巡回セールスマン問題と 2opt 近傍



局所最適解

2opt 近傍における局所最適化

事実

巡回セールスマン問題に対して，
2opt 近傍における局所最適化のために，最悪の場合，
2opt 操作を指数関数回繰り返さなくてはならない

- **グラフの場合** (Lueker '75)
- **2 次元ユークリッド平面の場合でも** (Englert, Röglin, Vöcking '07)

つまり，2opt 操作の反復で局所最適化を行うアルゴリズムは
多項式時間アルゴリズムではない

P vs NP 問題の (非) 重要性 (5)

非重要性に関する主張

メニコア，分散コンピューティングの時代になり，ボトルネックとなるのは計算ではなく通信である

誤解 10

「計算量」ということばは演算回数を指している

誤解の原因

「計算量」とは「computational complexity」の訳語

- 「complexity」という語に「演算回数」という意味はない

P vs NP 問題の (非) 重要性 (5)

非重要性に関する主張

メニコア，分散コンピューティングの時代になり，ボトルネックとなるのは計算ではなく通信である

誤解 10

「計算量」ということばは演算回数を指している

誤解の原因

「計算量」とは「computational complexity」の訳語

- **「complexity」という語に「演算回数」という意味はない**

いろいろな「計算量」

- **時間計算量** (time complexity)
- **領域計算量** (space complexity)
- **通信計算量** (communication complexity)
- **回路計算量** (circuit complexity)
- **反転計算量** (reversal complexity)
- ……

様々な側面から計算の本質に迫るのが計算複雑性理論

計算複雑性にまつわる 10 の誤解 — 最後に

「10 の誤解」シリーズ

- アルゴリズム (ソーティング)　(2009 年，KSMAP 合宿)
- 計算複雑性　(2013 年，北海道支部)
- ……

みなさんの分野における「10 の誤解」を教えて下さい

計算複雑性に関する研究プロジェクト

文部科学省 科学研究費補助金 新学術領域研究　('12 年度–'17 年度)

多面的アプローチの統合による計算限界の解明

英文略称：ELC (Exploring the Limits of Computation)

http://www.al.ics.saitama-u.ac.jp/elc/

計算複雑性にまつわる 10 の誤解 — 最後に

「10 の誤解」シリーズ

- アルゴリズム (ソーティング)　(2009 年，KSMAP 合宿)
- 計算複雑性　(2013 年，北海道支部)
- ……

みなさんの分野における「10 の誤解」を教えて下さい

計算複雑性に関する研究プロジェクト

文部科学省 科学研究費補助金 新学術領域研究　('12 年度–'17 年度)

多面的アプローチの統合による計算限界の解明

英文略称：ELC (Exploring the Limits of Computation)

http://www.al.ics.saitama-u.ac.jp/elc/

計算複雑性にまつわる 10 の誤解

岡本 吉央

電気通信大学 大学院情報理工学研究科 情報・通信工学専攻
okamotoy@uec.ac.jp

2013 年 8 月 8 日
日本オペレーションズ・リサーチ学会
北海道支部サマースクール 2013