

情報工学の最先端

—組み合わせ最適化問題への招待—

池口 徹

埼玉大学 工学部 情報システム工学科

(大学院 理工学研究科 数理電子情報部門 情報領域)

〒 338-8570 さいたま市桜区下大久保 255

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

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

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

本日の内容

1. 埼玉大学 工学部 情報システム工学科
1 年前期 必修「情報システム工学入門」
 - 学科全教員によるオムニバス形式
 - 最新の情報システム工学について

本日の内容

1. 埼玉大学 工学部 情報システム工学科
1 年前期 必修「情報システム工学入門」
 - 学科全教員によるオムニバス形式
 - 最新の情報システム工学について



2. 今日の題材は
 - (a) H16 年度 池口 担当分
「組み合わせ最適化問題への招待」
 - (b) 50 分版として短縮
 - 大学の講義 = 90 分一コマ .
 - 少し簡単にしてあります .

最適化とは？

1. 最適化

高校 関数 $y = f(x)$ の最大値 or 最小値を求めよ .

大学 関数 の最大値 or 最小値を求めよ .

- 情報数学入門
- 応用線形代数
- 応用解析学

2. 最適化 = 今日の主題

- 離散数学
- 計算論
- アルゴリズムとデータ構造

最適化とは？

1. 連続 最適化

高校 関数 $y = f(x)$ の最大値 or 最小値を求めよ .

大学 関数 の最大値 or 最小値を求めよ .

- ・ 情報数学入門
- ・ 応用線形代数
- ・ 応用解析学

2. 離散 最適化

= 今日の主題

- ・ 離散数学
- ・ 計算論
- ・ アルゴリズムとデータ構造

最適化とは？

1. 連続 最適化

高校 関数 $y = f(x)$ の最大値 or 最小値を求めよ。

大学 関数 $y = f(x_1, x_2, \dots, x_n)$ の最大値 or 最小値を求めよ。

- ・ 情報数学入門
- ・ 応用線形代数
- ・ 応用解析学

2. 離散 最適化

= 今日の主題

- ・ 離散数学
- ・ 計算論
- ・ アルゴリズムとデータ構造

最適化とは？

1. 連続 最適化

高校 関数 $y = f(x)$ の最大値 or 最小値を求めよ。

大学 関数 $y = f(x_1, x_2, \dots, x_n)$ の最大値 or 最小値を求めよ。

- ・ 情報数学入門
- ・ 応用線形代数
- ・ 応用解析学

2. 離散 最適化 = 組み合わせ最適化 = 今日の主題

- ・ 離散数学
- ・ 計算論
- ・ アルゴリズムとデータ構造

ある高校生3年生 小石川 太郎 君の場合

- 受験勉強に励むべき夏休みに， から に連れて行ってほしいと頼まれた
- 憧れの からのお願いに，太郎君は大喜び．
- しかし太郎君は 未経験のため，状況が全く分からない．
- ホームページを調べたところ，
のアトラクションの数は ！
のアトラクションの数は ！
- 人気女優 の休日は，たったの一日．
と のアトラクション全てを一日で回り切り，憧れの の に喜んでもらいたい．
- 全部で ものアトラクション全てを回り，エントランスに戻る最短巡回路は？

ある高校生3年生 小石川 太郎 君の場合

- 受験勉強に励むべき夏休みに、長澤まさみ ちゃん から連れて行ってほしいと頼まれた
- 憧れの まさみちゃん からのお願いに、太郎君は大喜び。
- しかし太郎君は 未経験のため、状況が全く分からない。
- ホームページを調べたところ、
のアトラクションの数は ！
のアトラクションの数は ！
- 人気女優 まさみちゃん の休日は、たったの一日。
と のアトラクション全てを一日で回り切り、憧れの まさみちゃん に喜んでもらいたい。
- 全部で ものアトラクション全てを回り、エントランスに戻る最短巡回路は？

ある高校生3年生 小石川 太郎 君の場合

- 受験勉強に励むべき夏休みに、長澤まさみ ちゃん から
に連れて行ってほしいと頼まれた



-
-
-
-

ない。
、憧れ

のよごみちやろ に言われてもらいたい。

- 全部で 物のアトラクション全てを回り、エントランスに戻る最短巡回路は?

ある高校生3年生 小石川 太郎 君の場合

- ❑ 受験勉強に励むべき夏休みに、長澤まさみ ちゃん から連れて行ってほしいと頼まれた
- ❑ 憧れの まさみちゃん からのお願いに、太郎君は大喜び。
- ❑ しかし太郎君は 未経験のため、状況が全く分からない。
- ❑ ホームページを調べたところ、
のアトラクションの数は ！
のアトラクションの数は ！
- ❑ 人気女優 まさみちゃん の休日は、たったの一日。
と のアトラクション全てを一日で回り切り、憧れの まさみちゃん に喜んでもらいたい。
- ❑ 全部で ものアトラクション全てを回り、エントランスに戻る最短巡回路は？

ある高校生3年生 小石川 太郎 君の場合

- 受験勉強に励むべき夏休みに、長澤まさみ ちゃん から TDR に連れて行ってほしいと頼まれた
- 憧れの まさみちゃん からのお願いに、太郎君は大喜び。
- しかし太郎君は TDR 未経験のため、状況が全く分からない。
- ホームページを調べたところ、
のアトラクションの数は ！
のアトラクションの数は ！
- 人気女優 まさみちゃん の休日は、たったの一日。
と のアトラクション全てを一日で回り切り、憧れの まさみちゃん に喜んでもらいたい。
- 全部で ものアトラクション全てを回り、エントランスに戻る最短巡回路は？

ある高校生3年生 小石川 太郎 君の場合

- 受験勉強に励むべき夏休みに、長澤まさみ ちゃん から TDR に連れて行ってほしいと頼まれた
- 憧れの まさみちゃん からのお願いに、太郎君は大喜び。
- しかし太郎君は TDR 未経験のため、状況が全く分からない。
- ホームページを調べたところ、TDL のアトラクションの数は 52 ！
のアトラクションの数は ！
- 人気女優 まさみちゃん の休日は、たったの一日。
- TDL と のアトラクション全てを一日で回り切り、憧れの まさみちゃん に喜んでもらいたい。
- 全部で ものアトラクション全てを回り、エントランスに戻る最短巡回路は？

ある高校生3年生 小石川 太郎 君の場合

- ❑ 受験勉強に励むべき夏休みに、長澤まさみ ちゃん から TDR に連れて行ってほしいと頼まれた
- ❑ 憧れの まさみちゃん からのお願いに、太郎君は大喜び。
- ❑ しかし太郎君は TDR 未経験のため、状況が全く分からない。
- ❑ ホームページを調べたところ、
TDL のアトラクションの数は 52 ！
TDS のアトラクションの数は 35 ！
- ❑ 人気女優 まさみちゃん の休日は、たったの一日。
- ❑ TDL と TDS のアトラクション全てを一日で回り切り、憧れの まさみちゃん に喜んでもらいたい。
- ❑ 全部で どのアトラクション全てを回り、エントランスに戻る最短巡回路は？

ある高校生3年生 小石川 太郎 君の場合

- ❑ 受験勉強に励むべき夏休みに、長澤まさみ ちゃん から TDR に連れて行ってほしいと頼まれた
- ❑ 憧れの まさみちゃん からのお願いに、太郎君は大喜び。
- ❑ しかし太郎君は TDR 未経験のため、状況が全く分からない。
- ❑ ホームページを調べたところ、
TDL のアトラクションの数は 52 ！
TDS のアトラクションの数は 35 ！
- ❑ 人気女優 まさみちゃん の休日は、たったの一日。
- ❑ TDL と TDS のアトラクション全てを一日で回り切り、憧れの まさみちゃん に喜んでもらいたい。
- ❑ 全部で 87 ものアトラクション全てを回り、エントランスに戻る最短巡回路は？

ある高校生3年生 小石川 太郎 君の場合

- ❑ 受験勉強に励むべき夏休みに、長澤まさみ ちゃん から TDR に連れて行ってほしいと頼まれた
- ❑ 憧れの まさみちゃん からのお願いに、太郎君は大喜び。
- ❑ しかし太郎君は TDR 未経験のため、状況が全く分からない。
- ❑ ホームページを調べたところ、
TDL のアトラクションの数は 52 ！
TDS のアトラクションの数は 35 ！
- ❑ 人気女優 まさみちゃん の休日は、たったの一日。
- ❑ TDL と TDS のアトラクション全てを一日で回り切り、憧れの まさみちゃん に喜んでもらいたい。
- ❑ 全部で 87 ものアトラクション全てを回り、エントランスに戻る最短巡回路は？

S大学1年生 小石川 太郎 君の場合

- ❑ 大学に見事入学し，楽しい大学生生活が始まった
- ❑ 選んだサークルは，S大学でも有名なテニス系サークル!!
- ❑ 新歓コンパでの先輩達のいじめを何とかか乗り切り，
- ❑ 女の子たちとのサークル生活を楽しもうとしていた矢先
- ❑ ある先輩から以下のような命令が来てしまった!
「Sたま市にあるコンビニを全部回って，
サークルのポスターを張ってこい!」
- ❑ しかし，Sたま市内のコンビニ総数は，なんと672(推定)!!!
- ❑ 貴方に残された時間はごく僅かしかない．
- ❑ 672店ものコンビニを全て回りキャンパスに戻ってくる最短巡回回路は?

ある会社員 小石川 太郎 君の場合

- ❑ S 大学工学部情報システム工学科を見事優秀な成績で卒業し、日本でも有数の電気メーカーに就職できた
- ❑ 4 月からの新人研修も無難にこなし、いよいよ最後の営業研修が済めば、憧れの三田本社だ！
- ❑ しかし、営業研修で担当となった上司は、社内でも新人いじめで評判のベテラン課長．
- ❑ 研修最終日の上司からの命令が、「今から取引先の全営業所を回って、各所長に挨拶をして来い。」というトンでもないもの．
- ❑ 上司に行けと命令された営業所数は、全部で 318 ．
- ❑ 318 もの営業所全てを回り切り、各営業所で所長に挨拶をして、担当課長に終了の報告をするための最短巡回路は？

ある新郎 小石川 太郎 君の場合

- ❑ 就職先の電気メーカーで、ほぼ毎日の残業をこなしつつも、高校時代から付き合っている彼女(長澤まさみ?) といよいよ結婚
- ❑ こんどこそ、誰の虐めにもあわないぞ! と強く決心をする太郎君であったが …
- ❑ 新婚旅行のために数ヵ月も前から決めてあった7日間の休暇が、直属上司の命令で3日に短縮されてしまった。
- ❑ 困ったことに、彼女も、新婚旅行で回ろうと思っていた街は全て訪ねたいと絶対に譲らない。
- ❑ こんなことで新婚早々喧嘩をされていては、成田離婚(かなり古い!)にもなりかねない。
- ❑ 彼女の要望を聞き入れ、かつ、上司の言うとおりに3日で新婚旅行を切り上げるためには、当初計画にあった全ての街を効率よく訪問する必要がある。
- ❑ 全ての街を一度だけ訪れ、成田に戻ってくるための最短巡回路は?

渡る世間は鬼ばかり …

- ❑ 長澤まさみとのTDRデート問題
- ❑ サークルポスター配布問題
- ❑ 新入社員営業所挨拶回り問題
- ❑ 新婚旅行問題

人生の諸困難をうまく切り抜けるには?

⇒

の解を発見する効果的なアルゴリズムを考えれば良い

↓

まず、

について説明しよう!

渡る世間は鬼ばかり …

- ❑ 長澤まさみとのTDRデート問題
- ❑ サークルポスター配布問題
- ❑ 新入社員営業所挨拶回り問題
- ❑ 新婚旅行問題

人生の諸困難をうまく切り抜けるには?

⇒

巡回セールスマン問題 (Traveling Salesman Problem, TSP)
の解を発見する効果的なアルゴリズムを考えれば良い



まず、

について説明しよう!

渡る世間は鬼ばかり …

- ❑ 長澤まさみとのTDRデート問題
- ❑ サークルポスター配布問題
- ❑ 新入社員営業所挨拶回り問題
- ❑ 新婚旅行問題

人生の諸困難をうまく切り抜けるには?

⇒

巡回セールスマン問題 (Traveling Salesman Problem, TSP)
の解を発見する効果的なアルゴリズムを考えれば良い



まず、巡回セールスマン問題 について説明しよう!

巡回セールスマン問題

あるセールスマンが、 N 都市を巡回して営業をかけることになった。各都市間の距離が与えられたとして、これらの N 都市を一度だけ訪れ、最後に出発点となる都市に戻るような経路 (巡回路) の中で最も短いものを求めよ。



大学では、なぜか格好つけた方法で表現する。



N 個の節点 (都市) から構成されるグラフ $G = (V, E)$ 、枝上の距離 (重み, 費用) 関数 $c : E \rightarrow R$ が与えられたとき、全ての節点をちょうど 1 回ずつ経由する巡回路で、枝上の距離の合計 (巡回路長) を最小にするものを求めよ。

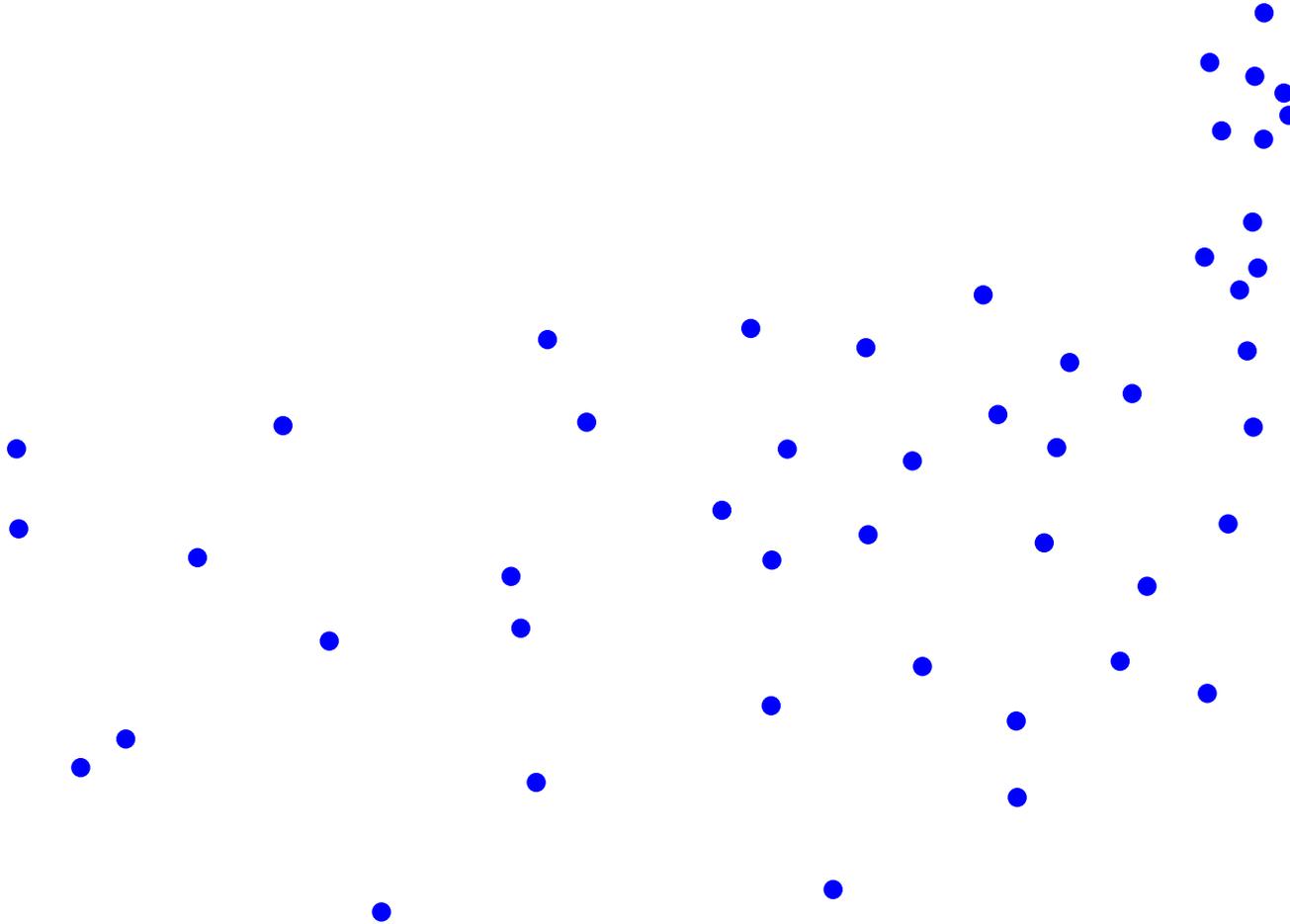
巡回セールスマン問題

あるセールスマンが、 N 都市を巡回して営業をかけることになった。各都市間の距離が与えられたとして、これらの N 都市を一度だけ訪れ、最後に出発点となる都市に戻るような経路 (巡回路) の中で最も短いものを求めよ。



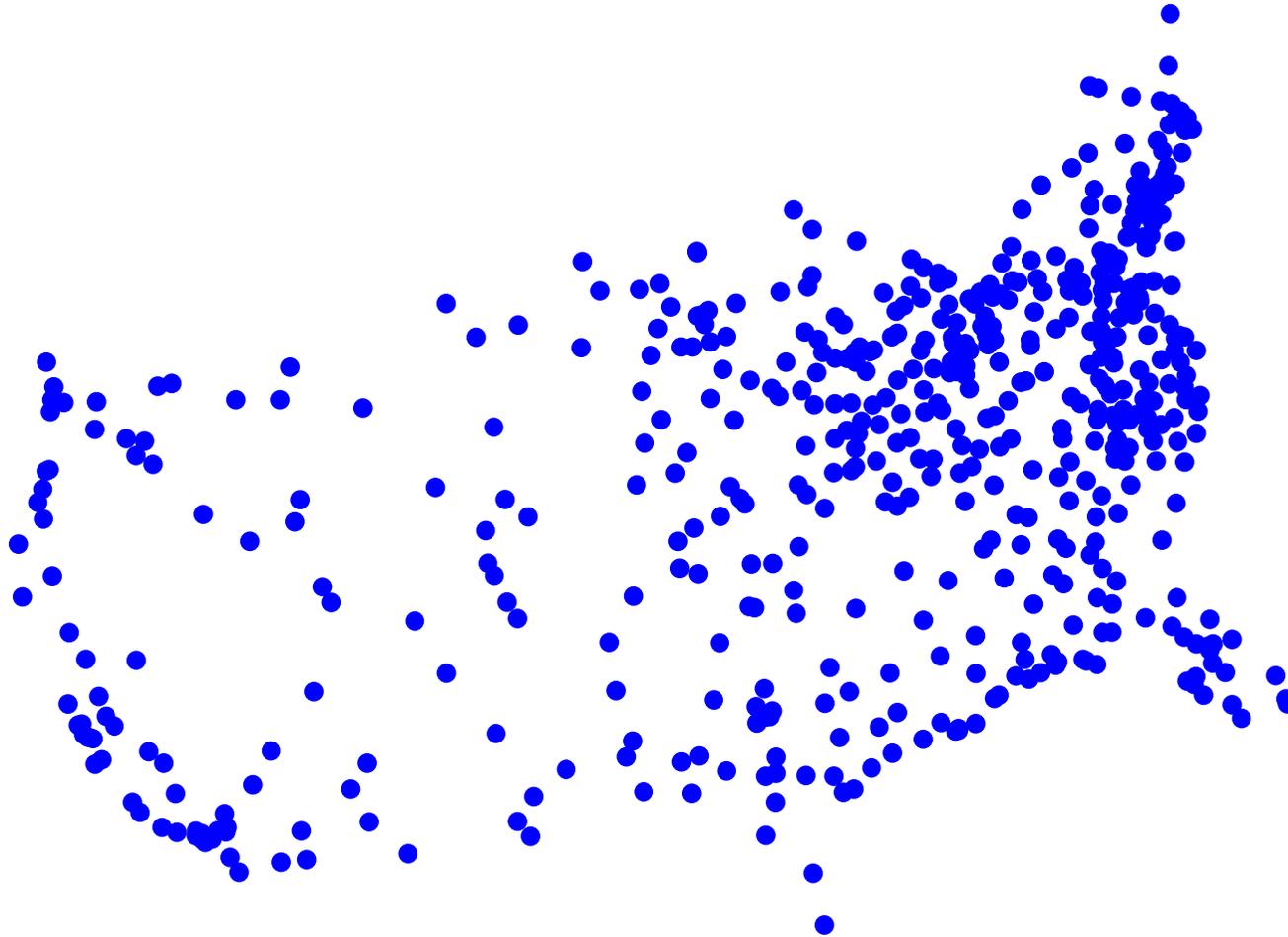
実際の研究現場では、どのような具体例が解かれているのかを確認してみよう。

TSP の例 (att48)



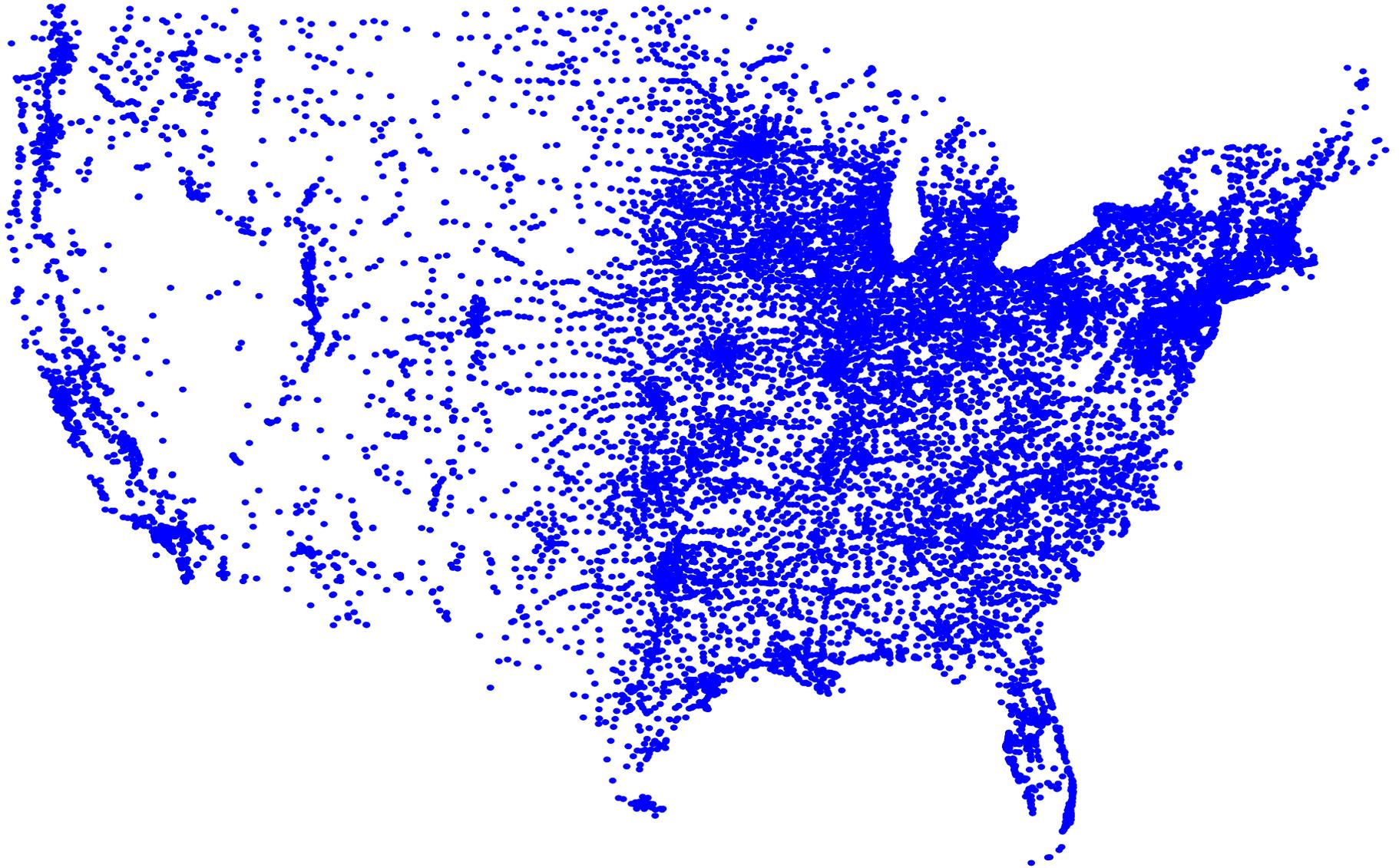
<http://www.iwr.euni-heidelberg.de/groups/comopt/software/TSPLIB95/>

TSP の例 (att532)

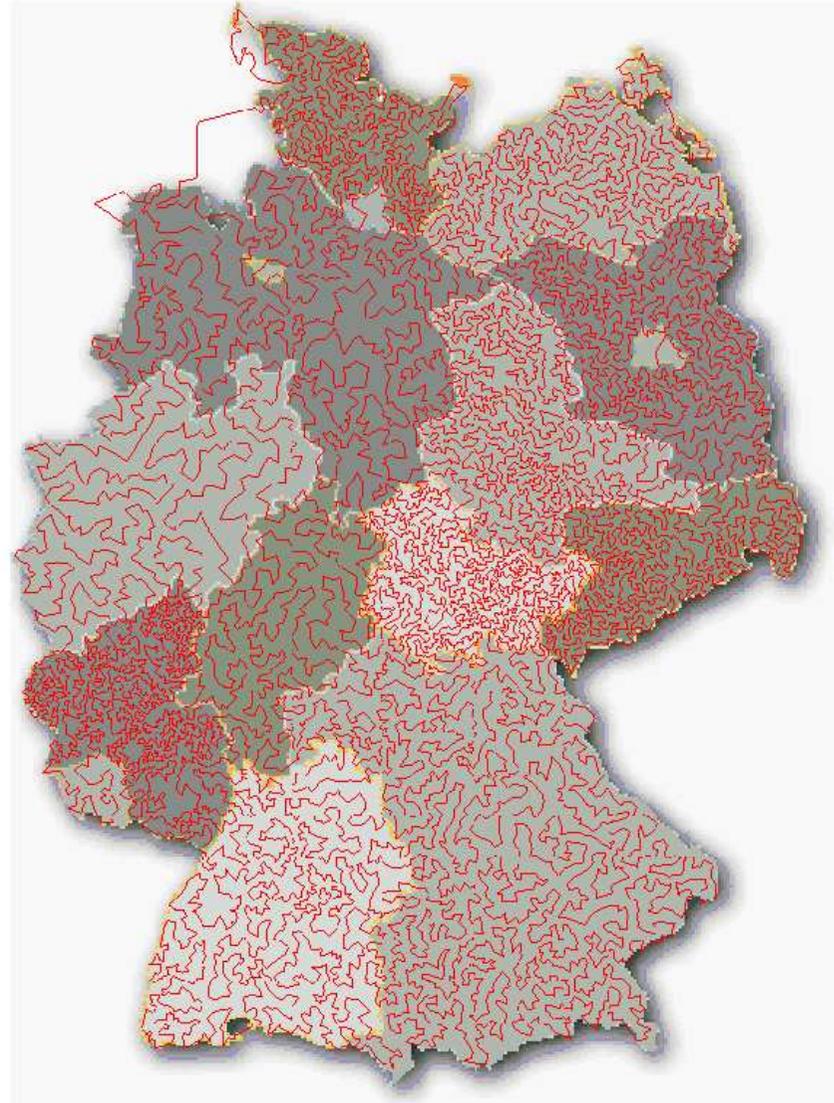
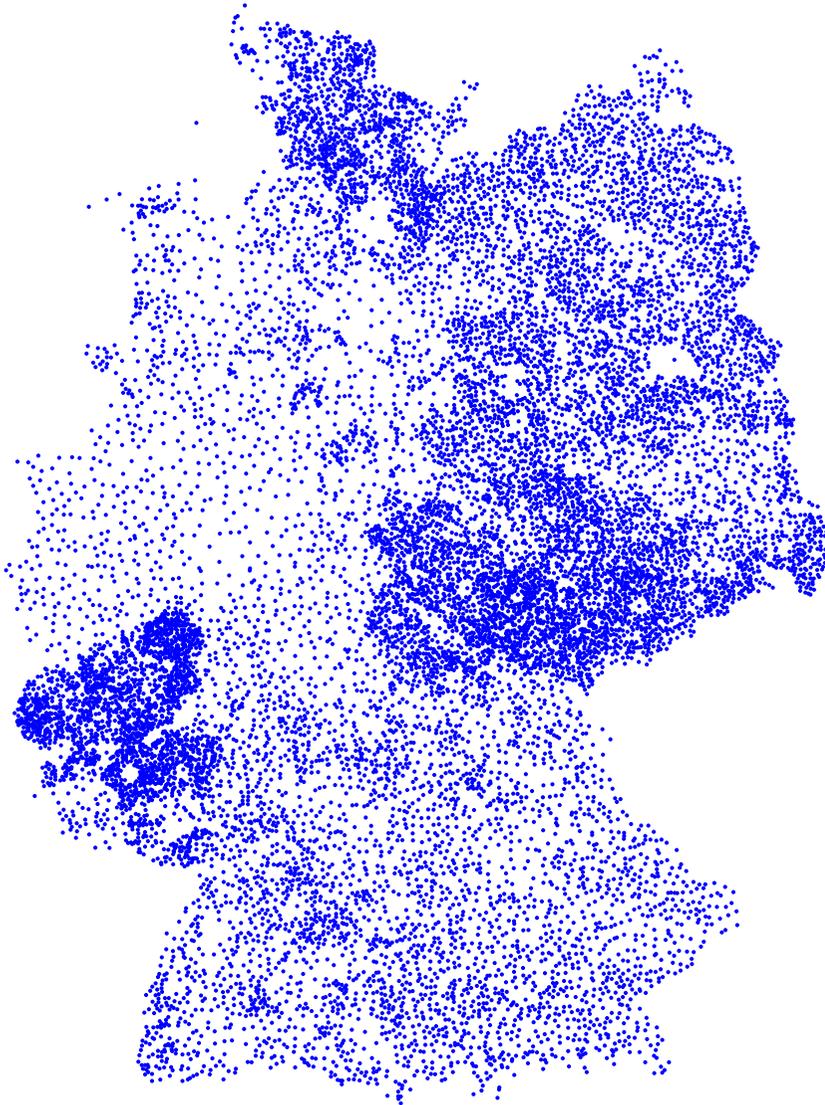


<http://www.iwr.euni-heidelberg.de/groups/comopt/software/TSPLIB95/>

TSP の例 (usa13509)

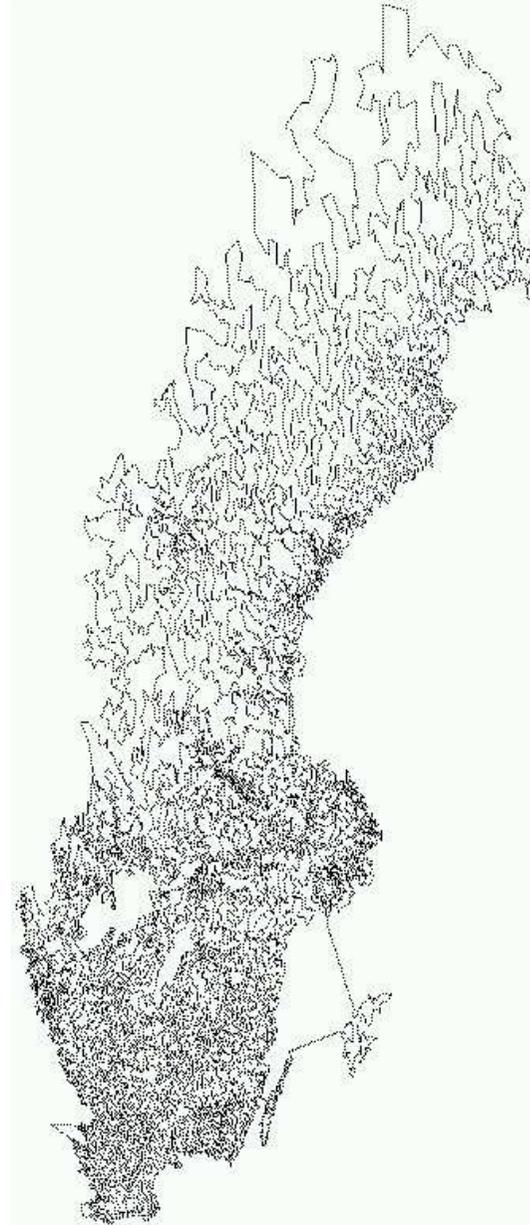
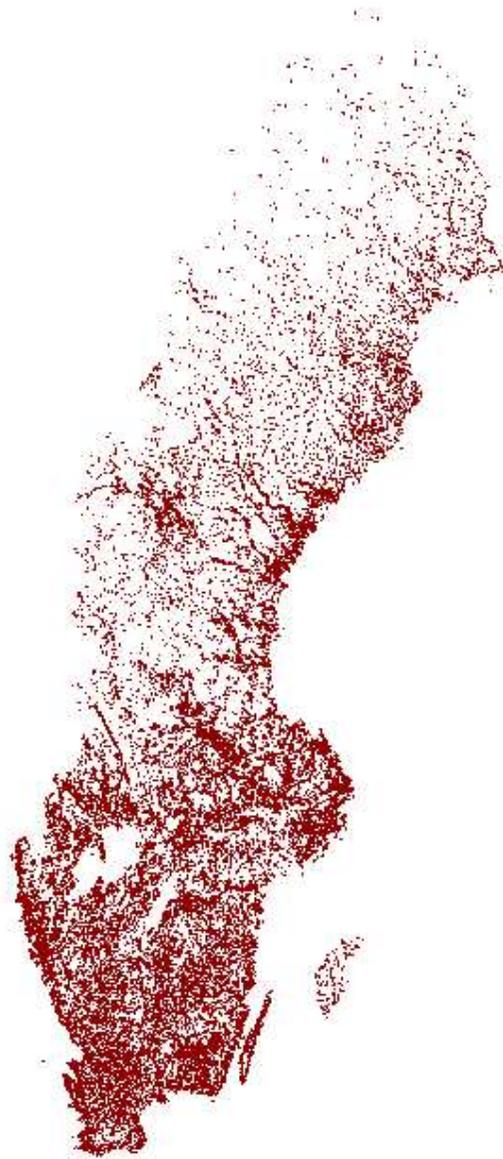


TSP の例 (d15112)

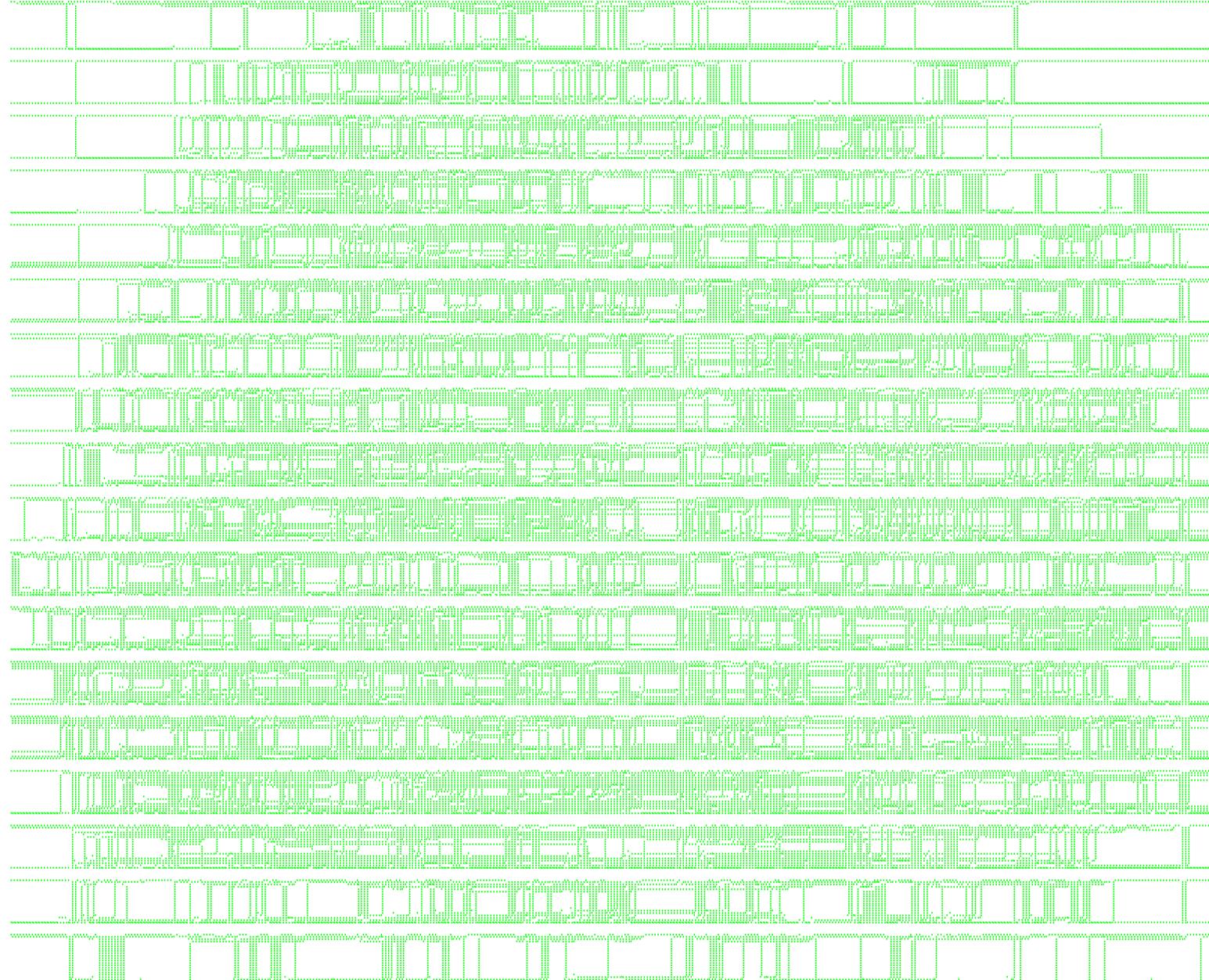


<http://www.iwr.euni-heidelberg.de/groups/comopt/software/TSPLIB95/>

TSP の例 (sw24978)



TSP の例 (pla85900)



TSP 関連の情報

1. Solving Traveling Salesman Problem
<http://www.math.princeton.edu/tsp/>
2. TSPLIB (16 都市 ~ 85900 都市のベンチマーク問題)
<http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>
3. TSPBIB
http://www.densis.fee.unicamp.br/moscato/TSPBIB_home.html

なぜTSPの解を求めようとするのか？

□ TSPを説明するのは **簡単** だが、最適解を求めるのは **難しい** で
はない。

□ **応用** も多い

⇒ TSPがうまく解けると小石川 太郎君の人生困難問題以外の
例題も解ける。

1. 基盤配線 (数百点)
2. 運搬経路計画
3. スケジューリング
4. 基盤穿孔 (数万点)
5. タンパク質構造解析
6. VLSI 設計 (数百万点 ~ 数千万点)

 **組み合わせ最適化問題の典型例**

⇒ TSPが解けると

なぜTSPの解を求めようとするのか？

□ TSPを説明するのは簡単だが、最適解を求めるのは簡単ではない。

□ も多い

⇒ TSPがうまく解けると小石川太郎君の人生困難問題以外の例題も解ける。

1. 基盤配線 (数百点)
2. 運搬経路計画
3. スケジューリング
4. 基盤穿孔 (数万点)
5. タンパク質構造解析
6. VLSI設計 (数百万点～数千万点)

 組み合わせ最適化問題の典型例

⇒ TSPが解けると

なぜTSPの解を求めようとするのか？

□ TSPを説明するのは簡単だが、最適解を求めるのは簡単ではない。

□ 応用 も多い

⇒ TSPがうまく解けると小石川 太郎君の人生困難問題以外の例題も解ける。

1. 基盤配線 (数百点)
2. 運搬経路計画
3. スケジューリング
4. 基盤穿孔 (数万点)
5. タンパク質構造解析
6. VLSI 設計 (数百万点 ~ 数千万点)

 組み合わせ最適化問題の典型例

⇒ TSPが解けると

なぜTSPの解を求めようとするのか？

□ TSPを説明するのは簡単だが、最適解を求めるのは簡単ではない。

□ 応用も多い

⇒ TSPがうまく解けると小石川太郎君の人生困難問題以外の例題も解ける。

1. 基盤配線 (数百点)
2. 運搬経路計画
3. スケジューリング
4. 基盤穿孔 (数万点)
5. タンパク質構造解析
6. VLSI設計 (数百万点～数千万点)

☞ **組み合わせ最適化問題の典型例**

⇒ TSPが解けると 他の問題もうまく解ける (可能性が高い)

それでは TSP を如何にして解くか?

□ 最初に思いつきそうな解法は? \implies

を描き, の巡回路の長さを出す.

– 最も短い巡回路長を与える訪問順を解とする.

それでは TSP を如何にして解くか?

- 最初に思いつきそうな解法は? \implies 列挙法
を描き, の巡回路の長さを出す.
 - 最も短い巡回路長を与える訪問順を解とする.

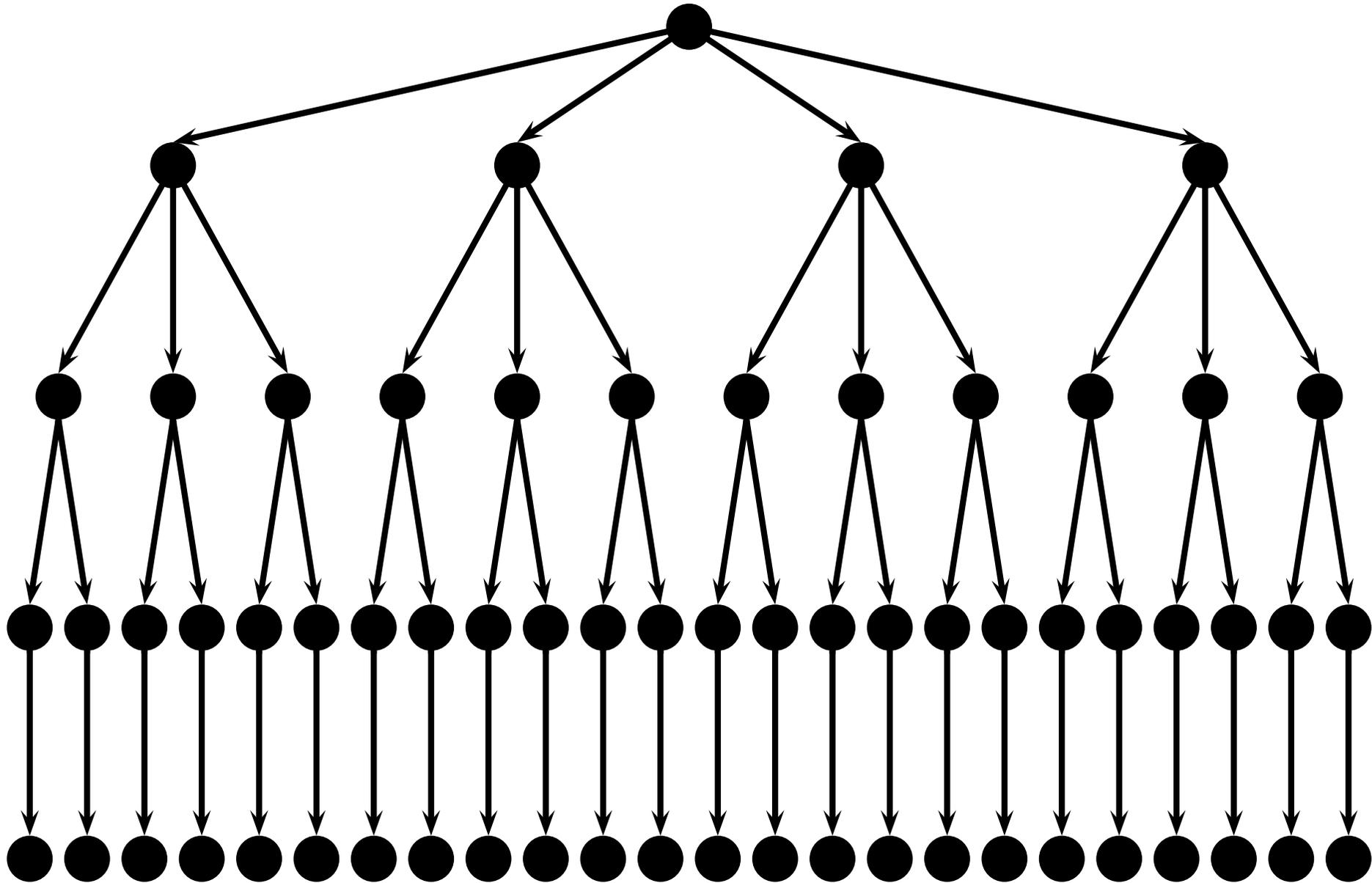
それでは TSP を如何にして解くか?

- 最初に思いつきそうな解法は? \implies 列挙法
 - 列挙木 を描き , の巡回路の長さを出す .
 - 最も短い巡回路長を与える訪問順を解とする .

それでは TSP を如何にして解くか?

- 最初に思いつきそうな解法は? \implies 列挙法
 - 列挙木 を描き, 全て の巡回路の長さを出す.
 - 最も短い巡回路長を与える訪問順を解とする.

列挙木



それでは TSP を如何にして解くか?

- 最初に思いつきそうな解法は? \implies 列挙法
 - 列挙木を描き全ての巡回路の長さを出す .
 - 最も短い巡回路長を与える訪問順を解とする .
- ところが, この方法は .
 - 適用できるのは, せいぜい ぐらいまで
 - \therefore 巡回路の総数は
 - $\implies N$ の増加とともに巡回路総数が

それでは TSP を如何にして解くか?

- 最初に思いつきそうな解法は? \implies 列挙法
 - 列挙木を描き全ての巡回路の長さを出す .
 - 最も短い巡回路長を与える訪問順を解とする .
- ところが, この方法は 全く実用的でない .
 - 適用できるのは, せいぜい ぐらいまで
 - ∴ 巡回路の総数は
 - $\implies N$ の増加とともに巡回路総数が

それでは TSP を如何にして解くか?

- 最初に思いつきそうな解法は? \implies 列挙法
 - 列挙木を描き全ての巡回路の長さを出す .
 - 最も短い巡回路長を与える訪問順を解とする .
- ところが, この方法は 全く実用的でない .
 - 適用できるのは, せいぜい $N = 13$ ぐらいまで
 - ∴ 巡回路の総数は
 - $\implies N$ の増加とともに巡回路総数が

それでは TSP を如何にして解くか?

- 最初に思いつきそうな解法は? \implies 列挙法
 - 列挙木を描き全ての巡回路の長さを出す .
 - 最も短い巡回路長を与える訪問順を解とする .
- ところが, この方法は 全く実用的でない .
 - 適用できるのは, せいぜい $N = 13$ ぐらいまで
 - \therefore 巡回路の総数は $(N - 1)!/2$
 $\implies N$ の増加とともに巡回路総数が

それでは TSP を如何にして解くか?

- 最初に思いつきそうな解法は? \implies 列挙法
 - 列挙木を描き全ての巡回路の長さを出す .
 - 最も短い巡回路長を与える訪問順を解とする .
- ところが, この方法は 全く実用的でない .
 - 適用できるのは, せいぜい $N = 13$ ぐらいまで
 - \therefore 巡回路の総数は $(N - 1)!/2$
 - $\implies N$ の増加とともに巡回路総数が 指数関数的に増大

指数関数は曲者

指数関数 \iff あっという間に増える！

- ❑ 曾呂利新左衛門 (vs 豊臣秀吉)
将棋盤の升目に米粒を，一日目は一粒，二日目は二粒，三日目は四粒，四日目は八粒 … 置いていく
- ❑ 新聞紙を 回折れば月に届く； 万 km
(中原欧介の言葉，CX 月 9「やまとなでしこ」2000.10.9 ~ 12.18 より)

❑



指数関数は曲者

指数関数 \iff あっという間に増える！

- 曾呂利新左衛門 (vs 豊臣秀吉)

将棋盤の升目に米粒を，一日目は一粒，二日目は二粒，三日目は四粒，四日目は八粒 … 置いていく

$$\sum_{k=0}^{80} 2^k = 2^{81} - 2 \approx 2.4 \times 10^{24} = 8.9 \times 10^{17} \text{俵} = 3.5 \times 10^{11} \text{百万石}$$

- 新聞紙を 回折れば月に届く； 万 km
(中原欧介の言葉，CX 月 9「やまとなでしこ」2000.10.9 ~ 12.18 より)

□



指数関数は曲者

指数関数 \iff あっという間に増える！

- 曾呂利新左衛門 (vs 豊臣秀吉)

将棋盤の升目に米粒を，一日目は一粒，二日目は二粒，三日目は四粒，四日目は八粒 … 置いていく

$$\sum_{k=0}^{80} 2^k = 2^{81} - 2 \approx 2.4 \times 10^{24} = 8.9 \times 10^{17} \text{俵} = 3.5 \times 10^{11} \text{百万石}$$

- 新聞紙を 42 回折れば月に届く； 万 km
(中原欧介の言葉，CX 月 9「やまとなでしこ」2000.10.9 ~ 12.18 より)

□



指数関数は曲者

指数関数 \iff あっという間に増える！

- 曾呂利新左衛門 (vs 豊臣秀吉)

将棋盤の升目に米粒を，一日目は一粒，二日目は二粒，三日目は四粒，四日目は八粒 … 置いていく

$$\sum_{k=0}^{80} 2^k = 2^{81} - 2 \approx 2.4 \times 10^{24} = 8.9 \times 10^{17} \text{ 俵} = 3.5 \times 10^{11} \text{ 百万石}$$

- 新聞紙を 42 回折れば月に届く； $0.1 \times 2^{42} = 40$ 万 km
(中原欧介の言葉，CX 月 9「やまとなでしこ」2000.10.9 ~ 12.18 より)

□



指数関数は曲者

指数関数 \iff あっという間に増える！

- 曾呂利新左衛門 (vs 豊臣秀吉)

将棋盤の升目に米粒を，一日目は一粒，二日目は二粒，三日目は四粒，四日目は八粒 … 置いていく

$$\sum_{k=0}^{80} 2^k = 2^{81} - 2 \approx 2.4 \times 10^{24} = 8.9 \times 10^{17} \text{ 俵} = 3.5 \times 10^{11} \text{ 百万石}$$

- 新聞紙を 42 回折れば月に届く； $0.1 \times 2^{42} = 40$ 万 km
(中原欧介の言葉，CX 月 9「やまとなでしこ」2000.10.9 ~ 12.18 より)

- ハノイの塔



スターリングの公式

- TSP の場合の列挙法は $N!$ のオーダー

$$N! = \sqrt{2\pi N} \left(\frac{N}{e}\right)^N \left(1 + \frac{1}{12N} + \frac{1}{288N^2} - \frac{139}{51840N^3} + O\left(\frac{1}{N^4}\right)\right)$$

だいたい $N! \approx N^N$ ということ。

- 入力サイズ: N , 計算時間: N^N

☞ 指数時間アルゴリズム (exponential time algorithm)



効率的でない \iff 計算量が指数関数的に増大する

TSPの解の総数

N	$(N - 1)!/2$	
4	3	
5	12	
6	60	
7	360	
8	2,520	
9	20,160	
10	181,440	18 万
11	1,814,400	180 万
12	19,958,400	1995 万
13	239,500,800	2 億
14	3,113,510,400	31 億
15	43,589,145,600	435 億
16	653,837,184,000	6538 億
17	10,461,394,944,000	10 兆
18	177,834,714,048,000	177 兆
19	3,201,185,852,864,000	3201 兆
20	60,822,550,204,416,000	6 京
⋮	⋮	
30	4,420,880,996,869,850,977,271,808,000,000	44 穰
⋮	⋮	
53	40,329,087,585,471,939,285,830,318,428,201,883,487,644,752,720,441,638,912,000,000,000,000	4 無量大数
100	4666310772197207634084961942813335024535798413219081073429648194760879999661495780447073198807825914312684896041361187912559260545843200000000000000000000	
⋮	⋮	
1000	$2.011936300385468 \times 10^{2564}$	
10000	$1.423129840458527 \times 10^{35655}$	

スパコン級の計算機だと

入力 サイズ N	計算時間					
	多項式			指数関数		
	N^2	N^3	N^5	2^N	3^N	$N!$
10	0.1 μ 秒	1 μ 秒	100 μ 秒	1 μ 秒	59 μ 秒	3.6 m 秒
20	0.4 μ 秒	8 μ 秒	3.2m 秒	1 m 秒	3.49 秒	77.1 年
30	0.9 μ 秒	27 μ 秒	24.3m 秒	1.07 秒	57.2 時間	5.6×10^5 宙齡
40	1.6 μ 秒	64 μ 秒	0.102 秒	18.3 分	386 年	1.72×10^{21} 宙齡
50	2.5 μ 秒	125 μ 秒	0.313 秒	313 時間	23 万世紀	6.43×10^{37} 宙齡
100	10 μ 秒	1 m 秒	10 秒	2679.8 宙齡	1.1×10^{21} 宙齡	1.97×10^{131} 宙齡

f : femto (10^{-15}), p : pico (10^{-12}), n : nano (10^{-9}), μ : micro (10^{-6}), P : Peta (10^{15}), T : Tera (10^{12}), G : Giga (10^9)

- ❑ 一巡回路を 1×10^{-9} (ナノ) 秒で求めることができるくらい高速なコンピュータ
- ❑ 宙齡 = 150 億年
- ❑ $N = 20$ 程度で殆んど不可能
- ❑ 現実的な時間で計算を終了させるには, $N = 15 \sim$ 程度
- ❑ 通常のパソコンなら $N = 12$ 程度が限界
⇒ 厳密さを諦めて, 短い時間で精度の良い近似解を見つけることが必要!

出典: 久保幹雄, 組み合わせ最適化とアルゴリズム, 共立出版, 2000

それでは TSP を如何にして解くか?

- ❑ 最初に思いつきそうな解法は? \Rightarrow 列挙法
 - 全ての巡回路の長さを出す.
 - 最も短い巡回路長を与える訪問順を解とする.
- ❑ ところが, この方法は全く実用的でない.
 - せいぜい $N = 13$ ぐらいまで
 - \therefore 巡回路の総数は $(N - 1)!/2$
 - $\Rightarrow N$ の増加とともに総巡回路数が指数関数的に増大

👉 もし今日の話聞いて, TSP の解法に興味を持ったとしても, より
大きいサイズの問題 (うまく作っても 程度) を列挙法のアルゴリズム
で絶対に解いてはいけない!

✌️ もちろん, その問題に対して列挙法でない を見つけ
ることが出来れば, それを用いればよい.

👉 ある問題を解くための
を考える必要もある.

それでは TSP を如何にして解くか?

- 最初に思いつきそうな解法は? \Rightarrow 列挙法
 - 全ての巡回路の長さを出す.
 - 最も短い巡回路長を与える訪問順を解とする.
- ところが, この方法は全く実用的でない.
 - せいぜい $N = 13$ ぐらいまで
 - \therefore 巡回路の総数は $(N - 1)!/2$
 $\Rightarrow N$ の増加とともに総巡回路数が指数関数的に増大
- ☞ もし今日の話聞いて, TSP の解法に興味を持ったとしても, $N = 13$ より大きいサイズの問題 (うまく作っても $N = 17$ 程度) を列挙法のアルゴリズムで絶対に解いてはいけない!
- ☞ もちろん, その問題に対して列挙法でない を見つけ
ることが出来れば, それを用いればよい.
- ☞

ある問題を解くための
を考える必要もある.

それでは TSP を如何にして解くか?

- 最初に思いつきそうな解法は? ⇒ 列挙法
 - 全ての巡回路の長さを出す .
 - 最も短い巡回路長を与える訪問順を解とする .
- ところが, この方法は全く実用的でない .
 - せいぜい $N = 13$ ぐらいまで
 - ∴ 巡回路の総数は $(N - 1)!/2$
 - ⇒ N の増加とともに総巡回路数が指数関数的に増大
- ☞ もし今日の話聞いて, TSP の解法に興味を持ったとしても, $N = 13$ より大きいサイズの問題 (うまく作っても $N = 17$ 程度) を列挙法のアルゴリズムで絶対に解いてはいけない!
- ☞ もちろん, その問題に対して列挙法でない「良い」アルゴリズム を見つけることが出来れば, それを用いればよい .

☞ ある問題を解くための
を考える必要もある .

それでは TSP を如何にして解くか?

- 最初に思いつきそうな解法は? ⇒ 列挙法
 - 全ての巡回路の長さを出す.
 - 最も短い巡回路長を与える訪問順を解とする.
- ところが, この方法は全く実用的でない.
 - せいぜい $N = 13$ ぐらいまで
 - ∴ 巡回路の総数は $(N - 1)!/2$
 - ⇒ N の増加とともに総巡回路数が指数関数的に増大
- ☞ もし今日の話聞いて, TSP の解法に興味を持ったとしても, $N = 13$ より大きいサイズの問題 (うまく作っても $N = 17$ 程度) を列挙法のアルゴリズムで絶対に解いてはいけない!
- ☞ もちろん, その問題に対して列挙法でない「良い」アルゴリズムを見つけることが出来れば, それを用いればよい.
- ☞

ある問題を解くための「良い」アルゴリズムがあるのか無いのかを考える必要もある.

アルゴリズム (algorithm)

- ❑ **直観的な説明**
 - 「ある問題を解くための手順，算法」
 - 「命題の真偽を確かめるための手続き」
- ❑ **語源**

ペルシャの数学者 アル・クワーリズミ
(Abu Ja'far Mohammed ibn Mûsâ al-Khowârizmi)
cf. 代数学 (algebra)
- ❑ **最古のアルゴリズム**

ユークリッドの互助法 (最大公約数を求めるアルゴリズム)
- ❑ **英語では，可算名詞 (countable)**

an algorithm for ... ing

アルゴリズムと計算量



アルゴリズム

(Polynomial time algorithm)

問題のサイズ N に対して、その問題の解を求めるための計算量が、 N の となるアルゴリズム。

⇒ アルゴリズム



アルゴリズム

(exponential time algorithm)

問題のサイズ N に対して、その問題の解を求めるための計算量が、 N の となるアルゴリズム。

⇒ アルゴリズム



ある問題を解くために作ったアルゴリズムが
ことも必要

アルゴリズムと計算量

□ 多項式時間 アルゴリズム

(Polynomial time algorithm)

問題のサイズ N に対して、その問題の解を求めるための計算量が、 N の 多項式 となるアルゴリズム。

⇒ アルゴリズム

□ アルゴリズム

(exponential time algorithm)

問題のサイズ N に対して、その問題の解を求めるための計算量が、 N の となるアルゴリズム。

⇒ アルゴリズム

 ある問題を解くために作ったアルゴリズムが
ことも必要

アルゴリズムと計算量

□ 多項式時間 アルゴリズム

(Polynomial time algorithm)

問題のサイズ N に対して、その問題の解を求めるための計算量が、 N の 多項式 となるアルゴリズム。

⇒ 効率的な良い アルゴリズム

□ アルゴリズム

(exponential time algorithm)

問題のサイズ N に対して、その問題の解を求めるための計算量が、 N の となるアルゴリズム。

⇒ アルゴリズム

 ある問題を解くために作ったアルゴリズムが
ことも必要

アルゴリズムと計算量

□ 多項式時間 アルゴリズム

(Polynomial time algorithm)

問題のサイズ N に対して、その問題の解を求めるための計算量が、 N の 多項式 となるアルゴリズム。

⇒ 効率的な良い アルゴリズム

□ 指数時間 アルゴリズム

(exponential time algorithm)

問題のサイズ N に対して、その問題の解を求めるための計算量が、 N の となるアルゴリズム。

⇒ アルゴリズム

 ある問題を解くために作ったアルゴリズムが
ことも必要

アルゴリズムと計算量

□ 多項式時間 アルゴリズム

(Polynomial time algorithm)

問題のサイズ N に対して、その問題の解を求めるための計算量が、 N の 多項式 となるアルゴリズム。

⇒ 効率的な良い アルゴリズム

□ 指数時間 アルゴリズム

(exponential time algorithm)

問題のサイズ N に対して、その問題の解を求めるための計算量が、 N の 指数関数 となるアルゴリズム。

⇒ アルゴリズム

 ある問題を解くために作ったアルゴリズムが
ことも必要

アルゴリズムと計算量

□ 多項式時間 アルゴリズム

(Polynomial time algorithm)

問題のサイズ N に対して、その問題の解を求めるための計算量が、 N の 多項式 となるアルゴリズム。

⇒ 効率的な良い アルゴリズム

□ 指数時間 アルゴリズム

(exponential time algorithm)

問題のサイズ N に対して、その問題の解を求めるための計算量が、 N の 指数関数 となるアルゴリズム。

⇒ 効率的でない悪い アルゴリズム

 ある問題を解くために作ったアルゴリズムが
ことも必要

アルゴリズムと計算量

□ 多項式時間 アルゴリズム

(Polynomial time algorithm)

問題のサイズ N に対して、その問題の解を求めるための計算量が、 N の 多項式 となるアルゴリズム。

⇒ 効率的な良い アルゴリズム

□ 指数時間 アルゴリズム

(exponential time algorithm)

問題のサイズ N に対して、その問題の解を求めるための計算量が、 N の 指数関数 となるアルゴリズム。

⇒ 効率的でない悪い アルゴリズム

 ある問題を解くために作ったアルゴリズムが 良いのか悪いのか を考える ことも必要

計算量と問題のクラス

□ 計算量 (=問題を解くときの時間) での分類

☞ 多項式時間アルゴリズム (polynomial time algorithm)

☞ 指数時間アルゴリズム (exponential time algorithm)

⇒ 効率的でない悪いアルゴリズム

□ 問題のクラス分け (の一つ)

— :

決定性アルゴリズムを用いて多項式時間で解ける問題のクラス
(Polynomial Time Solvable)

— :

非決定性アルゴリズムを用いて多項式時間で解ける問題のクラス
(Nondeterministic Polynomial Time Solvable)

注 1 Time Solvable ではない。

注 2 に属する問題は、その解が与えられた時、それが正しいことは多項式時間で分かる

注 3 列挙法で解ける問題は に属すると考えて、ほぼ間違いない

計算量と問題のクラス

□ 計算量 (=問題を解くときの時間) での分類

☞ 多項式時間アルゴリズム (polynomial time algorithm)

☞ 指数時間アルゴリズム (exponential time algorithm)

⇒ 効率的でない悪いアルゴリズム

□ 問題のクラス分け (の一つ)

– クラス \mathcal{P} :

決定性アルゴリズムを用いて多項式時間で解ける問題のクラス
(Polynomial Time Solvable)

– クラス \mathcal{NP} :

非決定性アルゴリズムを用いて多項式時間で解ける問題のクラス
(Nondeterministic Polynomial Time Solvable)

注 1 \mathcal{P} は \mathcal{NP} の部分集合であるが、 \mathcal{NP} は \mathcal{P} ではない。

注 2 \mathcal{NP} に属する問題は、その解が与えられた時、それが正しいことは多項式時間で分かる

注 3 列挙法で解ける問題は \mathcal{NP} に属すると考えて、ほぼ間違いない

計算量と問題のクラス

□ 計算量 (=問題を解くときの時間) での分類

☞ 多項式時間アルゴリズム (polynomial time algorithm)

☞ 指数時間アルゴリズム (exponential time algorithm)

⇒ 効率的でない悪いアルゴリズム

□ 問題のクラス分け (の一つ)

- クラス \mathcal{P} :

決定性アルゴリズムを用いて多項式時間で解ける問題のクラス
(Polynomial Time Solvable)

- クラス NP :

非決定性アルゴリズムを用いて多項式時間で解ける問題のクラス
(Nondeterministic Polynomial Time Solvable)

注 1 Time Solvable ではない。

注 2 に属する問題は，その解が与えられた時，それが正しい
ことは多項式時間で分かる

注 3 列挙法で解ける問題は に属すると考えて，ほぼ間違いない

計算量と問題のクラス

□ 計算量 (=問題を解くときの時間) での分類

☞ 多項式時間アルゴリズム (polynomial time algorithm)

☞ 指数時間アルゴリズム (exponential time algorithm)

⇒ 効率的でない悪いアルゴリズム

□ 問題のクラス分け (の一つ)

- クラス \mathcal{P} :

決定性アルゴリズムを用いて多項式時間で解ける問題のクラス
(Polynomial Time Solvable)

- クラス NP :

非決定性アルゴリズムを用いて多項式時間で解ける問題のクラス
(Nondeterministic Polynomial Time Solvable)

注 1 Not-Polynomial Time Solvable ではない。

注 2 　　　　　に属する問題は，その解が与えられた時，それが正しい
ことは多項式時間で分かる

注 3 列挙法で解ける問題は 　　　　　に属すると考えて，ほぼ間違いない

計算量と問題のクラス

□ 計算量 (=問題を解くときの時間) での分類

☞ 多項式時間アルゴリズム (polynomial time algorithm)

☞ 指数時間アルゴリズム (exponential time algorithm)

⇒ 効率的でない悪いアルゴリズム

□ 問題のクラス分け (の一つ)

– クラス \mathcal{P} :

決定性アルゴリズムを用いて多項式時間で解ける問題のクラス
(Polynomial Time Solvable)

– クラス NP :

非決定性アルゴリズムを用いて多項式時間で解ける問題のクラス
(Nondeterministic Polynomial Time Solvable)

注 1 Not-Polynomial Time Solvable ではない。

注 2 クラス NP に属する問題は, その解が与えられた時, それが正しいこととは多項式時間で分かる

注 3 列挙法で解ける問題は クラス NP に属すると考えて, ほぼ間違いない

決定性，非決定性

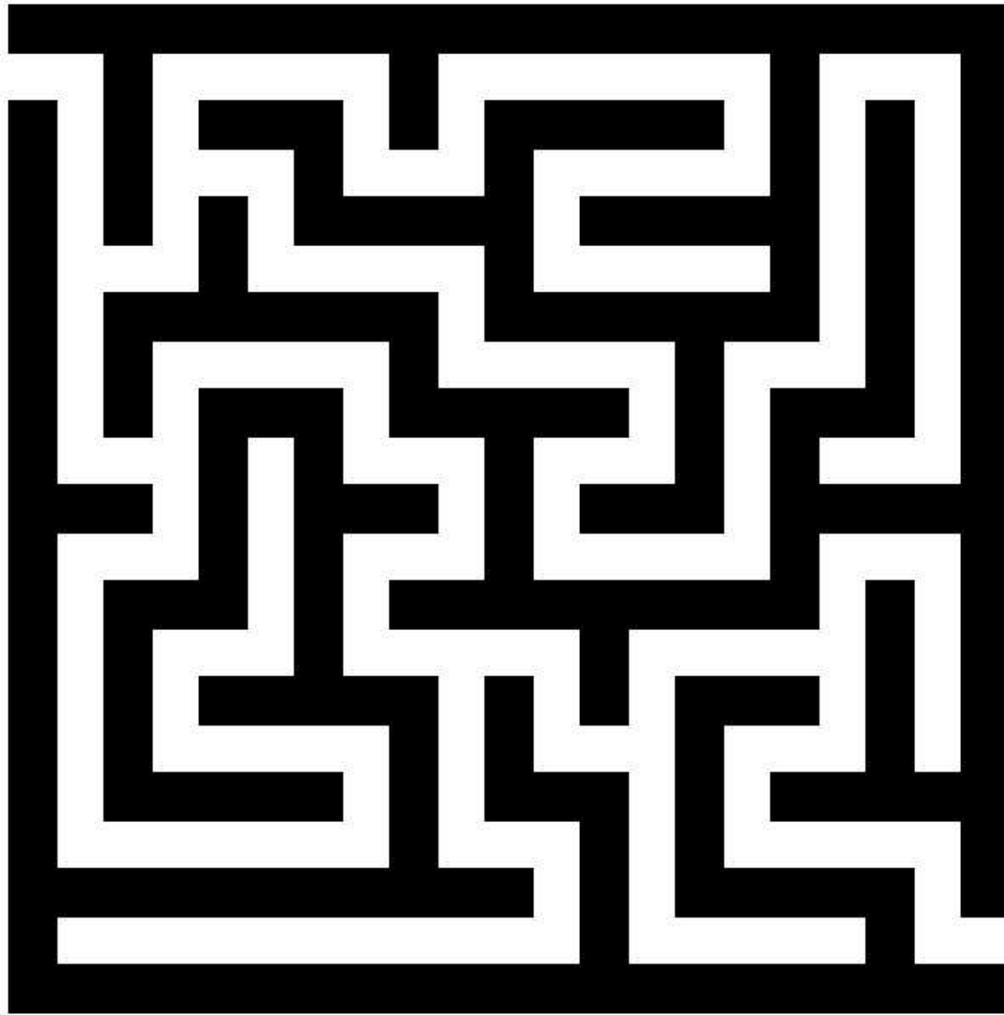
1. 決定性アルゴリズム (deterministic algorithm)

- ❑ アルゴリズムが動いているどの時点でも，次の動作が完全に決まっているアルゴリズム
- ❑ 決定性チューリングマシン
- ❑ 現在使われているコンピュータのこと

2. 非決定性アルゴリズム (nondeterministic algorithm)

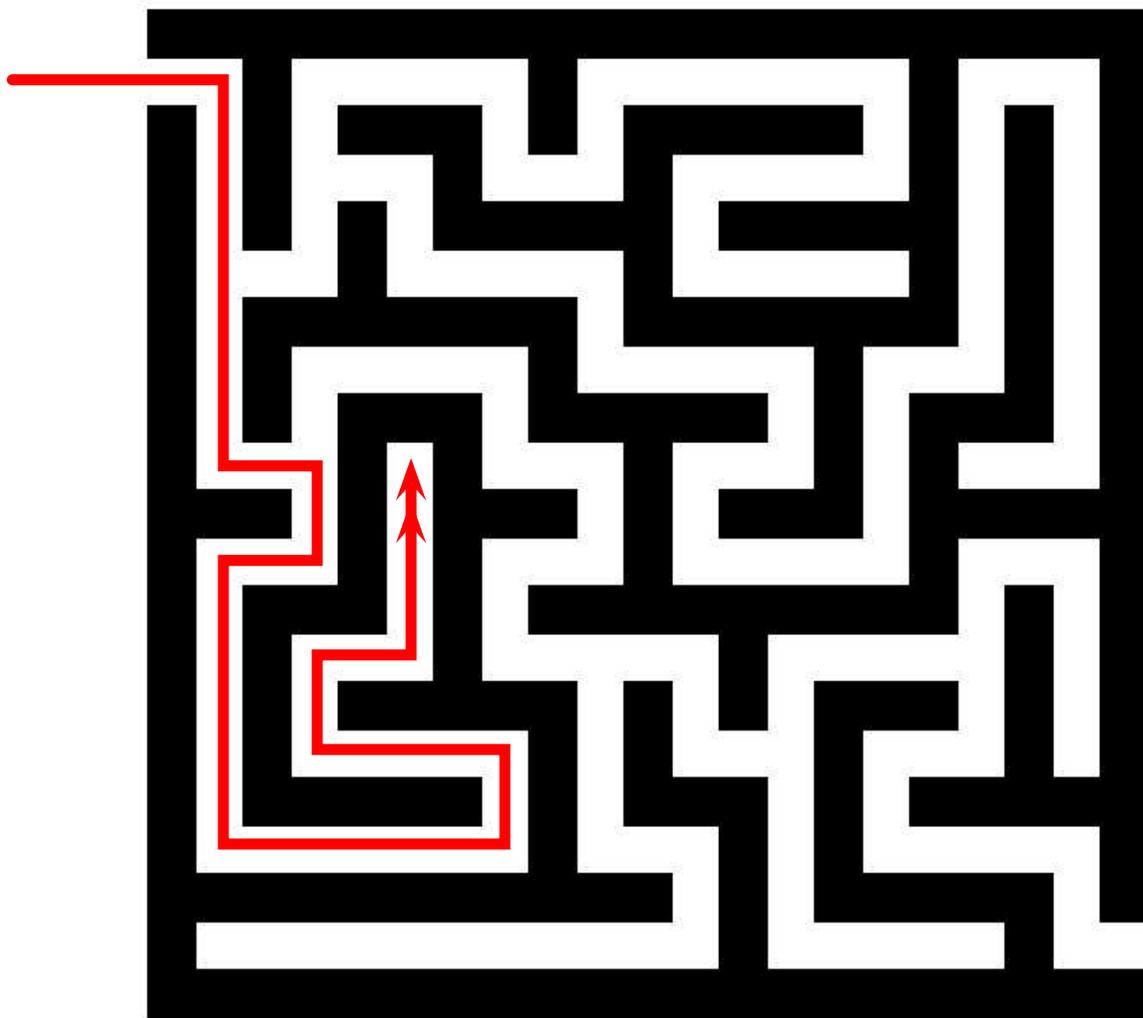
- ❑ 次の動作が必ずしも決まっていないアルゴリズム
- ❑ 非決定性チューリングマシン
- ❑ 山勘が鋭い計算機
- ❑ 占師—ただし未来が必ず当たる!
- ❑ もちろん，実在しない

迷路



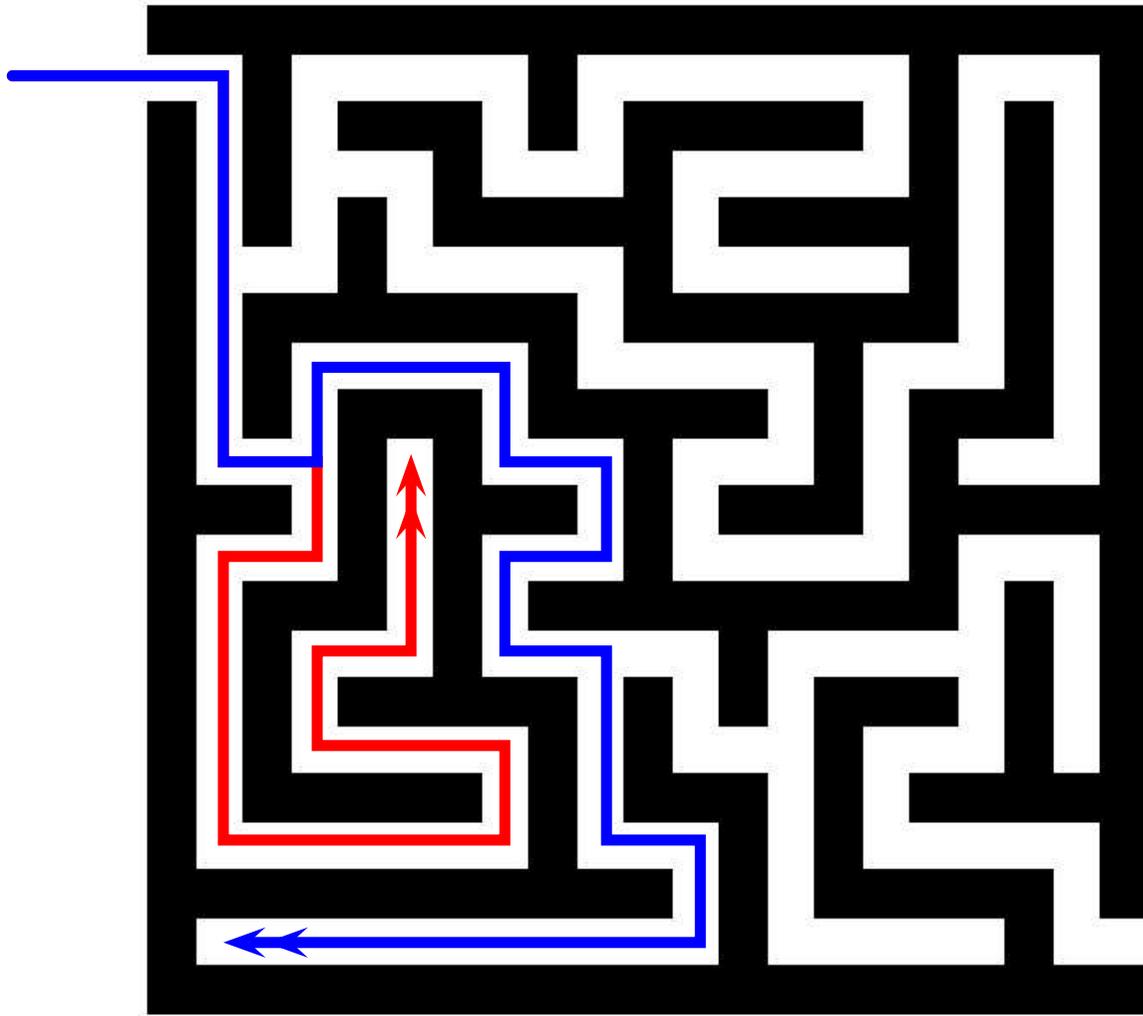
<http://karai.vis.ne.jp/>

迷路



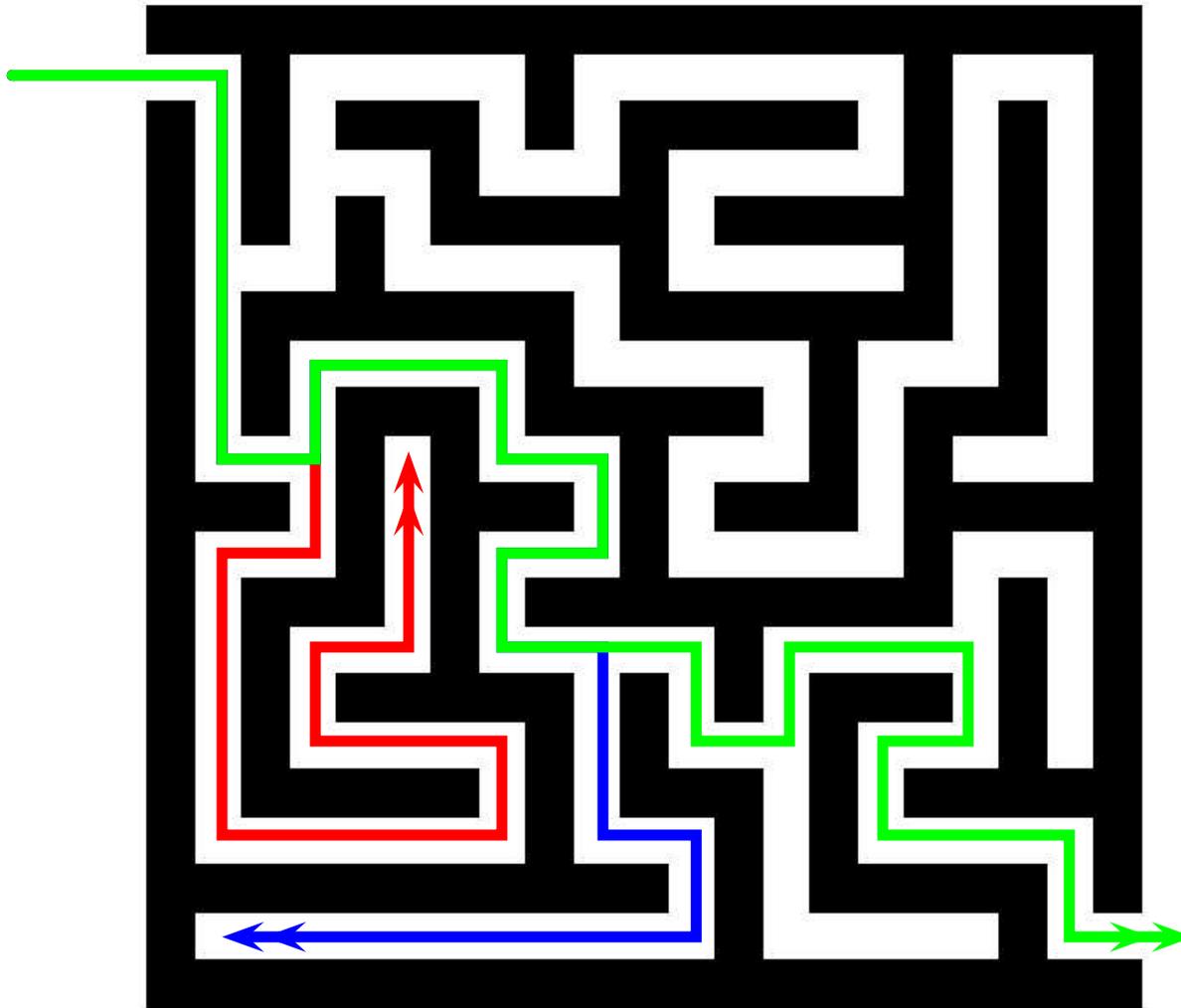
<http://karai.vis.ne.jp/>

迷路



<http://karai.vis.ne.jp/>

迷路



<http://karai.vis.ne.jp/>

なぜ P ? なぜ NP ?

ある問題が易しいか難しいか。



その問題を解くための「良い」アルゴリズムがあるのか無いのか。

1. もしその問題の解を求めるための多項式時間アルゴリズムが見つければ、その問題を解くことは容易 (つまりその問題は P に属する)
2. もしその問題の解を求めるための多項式時間アルゴリズムが見つからなければ ...

⇒

∴ 多項式時間アルゴリズムが見つからないからと言って、
その問題が とは言えない。

⇒ {

は一般に難しい
cf. 小泉前首相のイラク大量破壊兵器に関する発言
から

なぜ P ? なぜ NP ?

ある問題が易しいか難しいか。



その問題を解くための「良い」アルゴリズムがあるのか無いのか。

1. もしその問題の解を求めるための多項式時間アルゴリズムが見つければ、その問題を解くことは容易 (つまりその問題は P に属する)
 2. もしその問題の解を求めるための多項式時間アルゴリズムが見つからなければ ...
⇒ 実は、こちらはちょっとややこしい
- ∴ 多項式時間アルゴリズムが見つからないからと言って、
その問題が とは言えない。

⇒ { は一般に難しい
cf. 小泉前首相のイラク大量破壊兵器に関する発言
から

なぜ P ? なぜ NP ?

ある問題が易しいか難しいか。



その問題を解くための「良い」アルゴリズムがあるのか無いのか。

1. もしその問題の解を求めるための多項式時間アルゴリズムが見つければ、その問題を解くことは容易 (つまりその問題は P に属する)
 2. もしその問題の解を求めるための多項式時間アルゴリズムが見つからなければ ...
⇒ 実は、こちらはちょっとややこしい
- ∴ 多項式時間アルゴリズムが見つからないからと言って、その問題が 容易でない (P に属さない) とは言えない。

⇒ { は一般に難しい
cf. 小泉前首相のイラク大量破壊兵器に関する発言
から

なぜ P ? なぜ NP ?

ある問題が易しいか難しいか。



その問題を解くための「良い」アルゴリズムがあるのか無いのか。

1. もしその問題の解を求めるための多項式時間アルゴリズムが見つければ、その問題を解くことは容易 (つまりその問題は P に属する)
 2. もしその問題の解を求めるための多項式時間アルゴリズムが見つからなければ ...
⇒ 実は、こちらはちょっとややこしい
- ∴ 多項式時間アルゴリズムが見つからないからと言って、その問題が 容易でない (P に属さない) とは言えない。

⇒ { 不可能性の証明 は一般に難しい
cf. 小泉前首相のイラク大量破壊兵器に関する発言
から

なぜ P ? なぜ NP ?

ある問題が易しいか難しいか。



その問題を解くための「良い」アルゴリズムがあるのか無いのか。

1. もしその問題の解を求めるための多項式時間アルゴリズムが見つければ、その問題を解くことは容易 (つまりその問題は P に属する)
 2. もしその問題の解を求めるための多項式時間アルゴリズムが見つからなければ ...
⇒ 実は、こちらはちょっとややこしい
- ∴ 多項式時間アルゴリズムが見つからないからと言って、その問題が 容易でない (P に属さない) とは言えない。

⇒ { 不可能性の証明 は一般に難しい
cf. 小泉前首相のイラク大量破壊兵器に関する発言
努力が足りないだけかもしれない から

$P = NP$ か $P \neq NP$ か , それが問題だ

$P = NP$ か $P \neq NP$ かは未解決である .

注 1

ではある .

- 非決定性アルゴリズムに決定性アルゴリズムは含まれる .
- 直感で解く方 (非決定性アルゴリズム) が優れているはず .

注 2 多分

と信じられている .

- この世には , 解くことが難しいと考えられる問題が沢山存在する .
- これらの問題に対して , P のオーダのアルゴリズムが見つかっていない .
☞ 不可能性の証明は難しい .
- 但し , 信じられているだけで証明されていない .

注 3 従って、解決すれば有名になれる (ドル) .

- 否定的 , 肯定的どちらでも良い .
- クレイ数学研究所 (Clay Mathematics Institute)
ミレニアム懸賞問題 (Millennium Problems) の一つ

$\mathcal{P} = \text{NP}$ か $\mathcal{P} \neq \text{NP}$ か , それが問題だ

$\mathcal{P} = \text{NP}$ か $\mathcal{P} \neq \text{NP}$ かは未解決である .

注1 $\mathcal{P} \subseteq \text{NP}$ ではある .

- 非決定性アルゴリズムに決定性アルゴリズムは含まれる .
- 直感で解く方 (非決定性アルゴリズム) が優れているはず .

注2 多分 と信じられている .

- この世には , 解くことが難しいと考えられる問題が沢山存在する .
- これらの問題に対して , \mathcal{P} のオーダのアルゴリズムが見つかっていない .
☞ 不可能性の証明は難しい .
- 但し , 信じられているだけで証明されていない .

注3 従って、解決すれば有名になれる (ドル) .

- 否定的 , 肯定的どちらでも良い .
- クレイ数学研究所 (Clay Mathematics Institute)
ミレニアム懸賞問題 (Millennium Problems) の一つ

$\mathcal{P} = NP$ か $\mathcal{P} \neq NP$ か , それが問題だ

$\mathcal{P} = NP$ か $\mathcal{P} \neq NP$ かは未解決である .

注1 $\mathcal{P} \subseteq NP$ ではある .

- 非決定性アルゴリズムに決定性アルゴリズムは含まれる .
- 直感で解く方 (非決定性アルゴリズム) が優れているはず .

注2 多分 $\mathcal{P} \neq NP$ と信じられている .

- この世には , 解くことが難しいと考えられる問題が沢山存在する .
- これらの問題に対して , \mathcal{P} のオーダのアルゴリズムが見つかっていない .
☞ 不可能性の証明は難しい .
- 但し , 信じられているだけで証明されていない .

注3 従って、解決すれば有名になれる (ドル) .

- 否定的 , 肯定的どちらでも良い .
- クレイ数学研究所 (Clay Mathematics Institute)
ミレニアム懸賞問題 (Millennium Problems) の一つ

$\mathcal{P} = NP$ か $\mathcal{P} \neq NP$ か , それが問題だ

$\mathcal{P} = NP$ か $\mathcal{P} \neq NP$ かは未解決である .

注1 $\mathcal{P} \subseteq NP$ ではある .

- 非決定性アルゴリズムに決定性アルゴリズムは含まれる .
- 直感で解く方 (非決定性アルゴリズム) が優れているはず .

注2 多分 $\mathcal{P} \neq NP$ と信じられている .

- この世には , 解くことが難しいと考えられる問題が沢山存在する .
- これらの問題に対して , \mathcal{P} のオーダのアルゴリズムが見つかっていない .
☞ 不可能性の証明は難しい .
- 但し , 信じられているだけで証明されていない .

注3 従って、解決すれば有名になれる (+100万 ドル) .

- 否定的 , 肯定的どちらでも良い .
- クレイ数学研究所 (Clay Mathematics Institute)
ミレニアム懸賞問題 (Millennium Problems) の一つ

Millennium Problems

In order to celebrate mathematics in the new millennium, the Clay Mathematics Institute of Cambridge, Massachusetts (CMI) has named seven Prize Problems. The Scientific Advisory Board of CMI selected these problems, focusing on important classic questions that have resisted solution over the years. The Board of Directors of CMI designated a 7 million prize fund for the solution to these problems, with 1 million allocated to each. During the Millennium Meeting held on May 24, 2000 at the Collège de France, Timothy Gowers presented a lecture entitled The Importance of Mathematics, aimed for the general public, while John Tate and Michael Atiyah spoke on the problems. The CMI invited specialists to formulate each problem.

1. Birch and Swinnerton-Dyer Conjecture
2. Hodge Conjecture
3. Navier-Stokes Equations
4. P vs NP
5. Poincaré Conjecture
6. Riemann Hypothesis
7. Yang-Mills Theory

組み合わせ最適化問題の解法

- 厳密解法—(もちろん列挙法じゃない)
例: 分枝限定法など
膨大な計算量を要する．現実的な時間内では適用困難．
sw24978@TSP → CPU 時間で約 8 年分
- 近似解法—出来るだけ良い解を，出来るだけ短い時間で
 1. 精度保証がある
最適性の保証はないけれども，得られた解がどれくらい最適解から悪いのかということは保証されている．
 2. 精度保証が無い
最適性・精度の保証はないが，実際にやってみると解くスピードや性能が非常に良く，世の中から受け入れられている．
 - 構築法: 最遠挿入法
 - 逐次的改善法
2-opt, 3-opt, λ -opt , Lin-Kernighan
 - ローカルミニマムからの脱出
シミュレーテッド・アニーリング, ジェネティック・アルゴリズム,
ニューラルネットワーク, タブーサーチ, **カオスサーチ**

今日のお話はそろそろおしまい…

1. 組み合わせ最適化問題の例
 - (a) 巡回セールスマン問題
2. 組み合わせ最適化問題の解法
 - (a) 列挙法, 列挙木
 - (b) アルゴリズム
 - (c) 計算量
 - (d) 問題のクラス – \mathcal{P} と NP –
3. 組み合わせ最適化問題の解法の簡単な紹介
 - (a) 厳密解法
 - (b) 近似解法

最後に-皆さんに伝えたいこと-

- ➡ 中途半端なコンピュータの知識なら無くても良い。
コンピュータの OS といえば Windows である。
Windows で setup をダブルクリックできるのでバッチリだ。
UNIX? Makefile? configure? 何じゃそりゃ?
- ➡ 必要なのは数学力 (論理力), 物理的なセンス
 - ・ゲーム, CG, ネットワーク, どんなものも時間の推移と共に変化!
 - ・アルゴリズムの考案には論理力が必須

微分積分が出来ても, 情報システム工学科では意味あらへんわ...

ソフトウェアを書くのに, 物理なんか関係ないじゃん...
- ➡ 語学力も必要
 - ・最低限, 日本語と英語. 出来れば第二外国語.
 - ・外国語の習得には論理力も必要.

自動翻訳機がそのうちに出るから英語などできなくても良い.

辞書は goo とか excite などの辞書を使えば十分だ.

単語の意味を調べて, あとは適当に雰囲気です...

最後に-皆さんに伝えたいこと-

- ➡ 中途半端なコンピュータの知識なら無くても良い。
 - × コンピュータの OS といえば Windows である。
 - × Windows で setup をダブルクリックできるのでバッチリだ。
 - × UNIX? Makefile? configure? 何じゃそりゃ?
- ➡ 必要なのは数学力 (論理力), 物理的なセンス
 - ・ ゲーム, CG, ネットワーク, どんなものも時間の推移と共に変化!
 - ・ アルゴリズムの考案には論理力が必須

微分積分が出来ても, 情報システム工学科では意味あらへんわ...

ソフトウェアを書くのに, 物理なんか関係ないじゃん...
- ➡ 語学力も必要
 - ・ 最低限, 日本語と英語. 出来れば第二外国語.
 - ・ 外国語の習得には論理力も必要.

自動翻訳機がそのうちに出るから英語などできなくても良い.

辞書は goo とか excite などの辞書を使えば十分だ.

単語の意味を調べて, あとは適当に雰囲気です...

最後に-皆さんに伝えたいこと-

- ☞ 中途半端なコンピュータの知識なら無くても良い。
 - × コンピュータの OS といえば Windows である。
 - × Windows で setup をダブルクリックできるのでバッチリだ。
 - × UNIX? Makefile? configure? 何じゃそりゃ?
- ☞ 必要なのは数学力 (論理力), 物理的なセンス
 - ・ ゲーム, CG, ネットワーク, どんなものも時間の推移と共に変化!
 - ・ アルゴリズムの考案には論理力が必須
 - × 微分積分が出来ても, 情報システム工学科では意味あらへんわ...
 - × ソフトウェアを書くのに, 物理なんか関係ないじゃん...
- ☞ 語学力も必要
 - ・ 最低限, 日本語と英語. 出来れば第二外国語.
 - ・ 外国語の習得には論理力も必要.

自動翻訳機がそのうちに出るから英語などできなくても良い.
辞書は goo とか excite などの辞書を使えば十分だ.
単語の意味を調べて, あとは適当に雰囲気です...

最後に—皆さんに伝えたいこと—

- ☞ 中途半端なコンピュータの知識なら無くても良い。
 - × コンピュータの OS といえば Windows である。
 - × Windows で setup をダブルクリックできるのでバッチリだ。
 - × UNIX? Makefile? configure? 何じゃそりゃ?
- ☞ 必要なのは数学力 (論理力), 物理的なセンス
 - ・ ゲーム, CG, ネットワーク, どんなものも時間の推移と共に変化!
 - ・ アルゴリズムの考案には論理力が必須
 - × 微分積分が出来ても, 情報システム工学科では意味あらへんわ...
 - × ソフトウェアを書くのに, 物理なんか関係ないじゃん...
- ☞ 語学力も必要
 - ・ 最低限, 日本語と英語。出来れば第二外国語。
 - ・ 外国語の習得には論理力も必要。
 - × 自動翻訳機がそのうちに出るから英語などできなくても良い。
 - × 辞書は goo とか excite などの辞書を使えば十分だ。
 - × 単語の意味を調べて, あとは適当に雰囲気です...

今日の模擬講義に関する質問・コメントは随時受け付けます。

- ❑ **メール** : tohru@nls.ics.saitama-u.ac.jp
- ❑ **居室** : 埼玉大学 総合研究棟 5F 506 号室
- ❑ **電話** : 048-858-3577
- ❑ **今日の授業で使ったスライドの PDF ファイルは , 池口の講義サポートページにあります。**

<http://www.nls.ics.saitama-u.ac.jp/~tohru/Lectures/>
から , どうぞ .

ユーザ: saidai

パスワード: ics

