簡単そうで難しい組合せ最適化

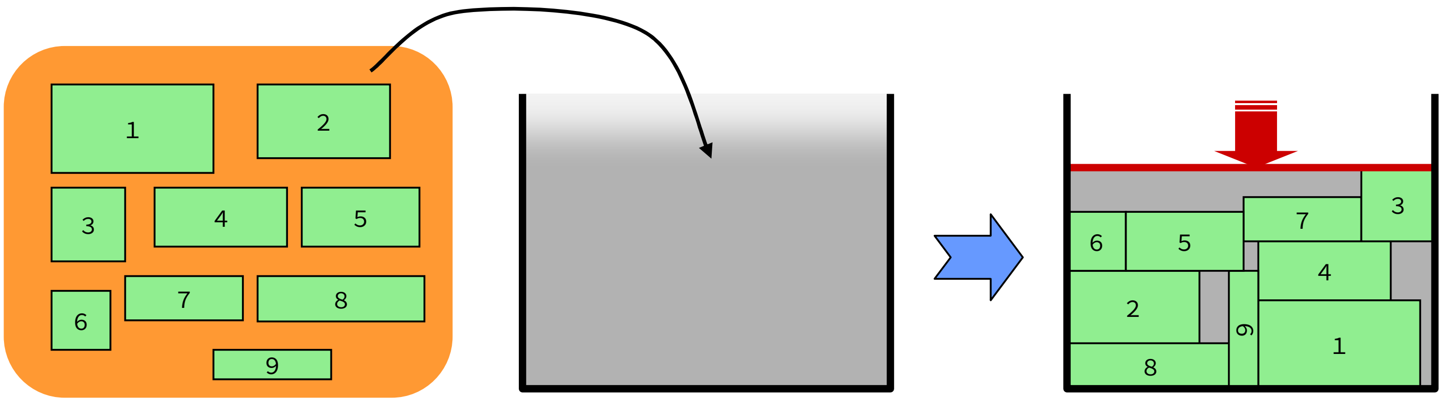
高校生，高専生，大学学部生の皆さん

私たちの研究室では，組合せ最適化（離散最適化）というものを研究の対象にしています．これは離散数学の問題ですが，私たちの身近なところにも現れています．ここでは組合せ最適化問題の例を挙げて，その解決に向けた研究についで説明いたします

京都大学工学部情報学科　数理工学コース
京都大学大学院情報学研究科　数理工学専攻　離散数理分野

長方形詰め込み問題

最初にパズルのような問題を紹介しましょう．左の図のようにいくつかの長方形が与えられ，これらを入れ物に重ならないように詰めます．このとき，右の図のように詰めた結果の高さをできるだけ低くすることがこの問題の目的です．



与えられた長方形　入れ物　詰めた結果

実はこの問題には産業において多くの応用があります（例えば，ロール鉄板から指定の形の部材を切り出すときに無駄になる部分が少なくなるように配置を決める問題など）．この問題については当研究室ホームページのパッキングパズルの項目もご覧ください．

組合せ最適化問題

まず，
　最適化問題とは「条件を満たす解の中で一番よいものを求める問題」
を指します．さらに，
　組合せ（離散）最適化とは「解が順序や割当のように
組合せ的な構造を持つ最適化問題」
のことを言います．

長方形詰め込み問題では，重ならないように入れ物に詰めることが満たすべき条件で，この条件を満たす解（詰め込み方）の中で高さが一番低いものが一番良い解となります．実社会では組合せ最適化として捉えることのできる課題が数多くあります．例えば

- 配送計画（コンビニへの商品配達、宅配）
- 工場での製品の機械への割当
- カーナビのルート探索
- スポーツの対戦表の作成
- 病院の看護師の勤務表作成

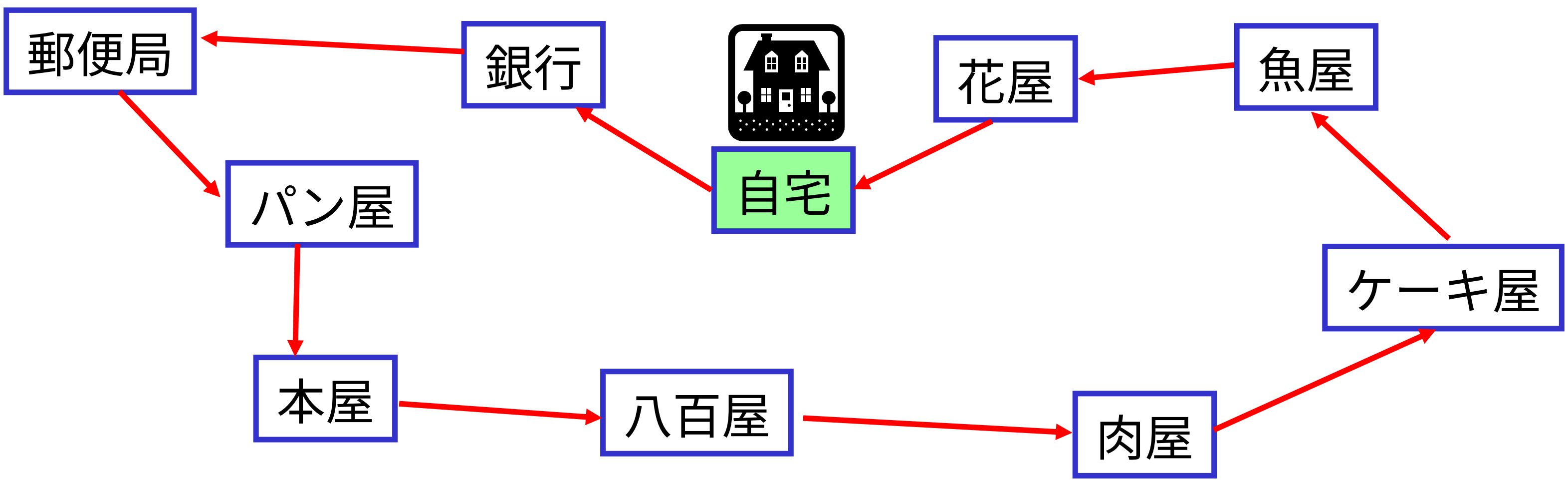
などなど…

以下身近な例を使いながら具体的に説明しましょう．

身近な例その１：買い物のルート

あなたは，郵便局、銀行、魚屋、本屋、八百屋、肉屋、パン屋、ケーキ屋、花屋に用事があるとします。どの順番で回ると無駄が少ないだろうか？

まずはお金をおろして，生ものは帰宅直前に買って，あそこのパン屋は早く行かないと閉まるから，というように訪問順に満たすべき条件があって，できるだけ移動距離や交通費を小さくしたいものです.



訪問する順番が変わると移動距離が違っています.
そこでこの点だけを問う次の問題の登場です.

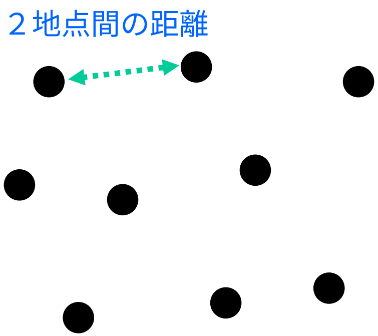
巡回セールスマン問題

セールスマンが都市を一度ずつ訪問する巡回路を求める話から，この問題には巡回セールスマンの名がついています．

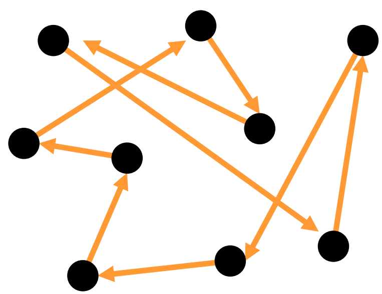
与えられるデータ： n 個の地点と２地点間の距離

条件： すべての地点を１度ずつ通り元の地点に戻る
（そういう移動ルートを巡回路と呼ぶ）

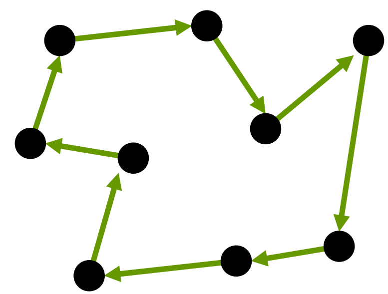
目標： 総距離をできるだけ短くしたい



n個の地点



ある巡回路



別の順回路
（こちらのほうが総距離が短い）

巡回路の選び方は $n! = n(n\text{-}1)(n\text{-}2)\cdots2\cdot1$ 通りです．すべての巡回路を列挙して，総距離が最小であるものを調べたら原理的には解ける．

今度は別の身近な例を見てみましょう．

身近な例その２：おやつの選択

遠足に持っていくおやつの合計金額は３００円までと
決められている。何を買って持っていけばいいだろう？

以下の6種類のお菓子からどれを買うか決めるとします．
それぞれに対する自分の満足度と値段が以下の通りであるとします．

	満足度	値段
ポテトチップス１袋	５点	１００円
チョコレート１枚	７点	１３０円
マシュマロ１袋	４点	８０円
飴１個	２点	５０円
ガム１箱	３点	７０円
せんべい１袋	６点	１１０円

- マシュマロとガムとせんべいを買うなら → 満足度１３点、値段２６０円
- ポテトチップスとチョコレートとガム → 満足度１５点、値段３００円
- ポテトチップスとチョコレートとせんべい → これは予算オーバー

おやつの選択の話をもう少し一般的に設定してみます．

ナップサック問題

ナップサックに食料，用具等の品物を詰める際，ナップサックの容量の範囲内でできるだけ有用な品物を選ぶところからこの問題にはナップサック問題の名がついています．

与えられるデータ： n 個の商品の満足度と値段および予算上限

条件： 選んだ商品の値段の合計が予算上限を超えない

目標： 選んだ商品の満足度合計をできるだけ大きくしたい

各商品に対して選ぶか否かの 2 通りの選択があるので
商品 a,b,c,... の組合せの数は

$n = 1$ ならば 2 通り（なし, a）
$n = 2$ ならば $2 \times 2 = 4$ 通り（なし, a, b, ab）
$n = 3$ ならば $2 \times 2 \times 2 = 8$ 通り（なし, a, b, c, ab, bc, ca, abc）
…

2^n 通りの商品の組合せを列挙して, すべて調べることで原理的には解ける．

列挙すれば解ける？

巡回セールスマン問題もナップサック問題も原理的には列挙すれば答えが見つかることになっています．しかし

可能な巡回路の数は　$n!$ 通り

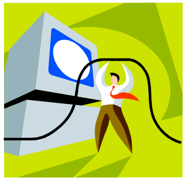
可能な商品の組合せは 2^n 通り．

$n!$ や 2^n は指数関数 と呼ばれる nが大きくなると急激に増加（爆発）する関数の仲間です．増加の具合を表で見てみると

n	$n!$	2^n
10	3,628,800	1,024
20	2.4×10^{18}	1,048,576
30	2.7×10^{32}	1,073,741,824

2.7×10^{32}回の演算を行にはスーパーコンピュータを使っても１００億年はかかります（絶句ですね）．

指数関数の爆発の例：　曾呂利新左衛門の逸話、鼠算など



最適解と近似解

最適解とは，条件を満たす解の中で最も良いもののこと．

➡ 現実的な時間で求めるのは難しい
（ＮＰ困難と呼ばれる難しい問題の仲間がある）

近似解とは，条件を満たす解の中で良いもののこと．

➡ 求める計算法は比較的容易

短時間で良い解を求めたいという要求は多い

必ずしも最適解でなくてもよい場合が多い

> 例： 買い物のルートを決めるのに時間をかけて綿密な計算をする人はいない。問題を解くのにかかる時間も大切な自分の時間だから。

以上から近似解を求める計算方法の研究も重要です．
近似解を求める方法として欲張り法、局所探索法などが知られています．

欲張り法

簡単な規則に基づいて解を構築する方法です．

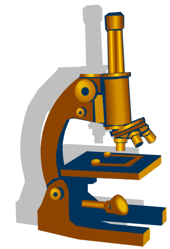
例） 次のナップサック問題（予算３００円）で考えてみます．
まず，値段あたりの満足度を計算し，この率の高い順に並べます．

	満足度	値段	満足度÷値段	順序
せんべい	６点	１１０円	０.０５５	1
チョコレート	７点	１３０円	０.０５４	2
ポテトチップス	５点	１００円	０.０５０	3
マシュマロ	４点	８０円	０.０５０	4
ガム	３点	７０円	０.０４３	5
飴	２点	５０円	０.０４０	6

次に順位の高い順にお菓子を選びます．このとき選んだお菓子を買うと予算オーバーになる場合には次のお菓子の選択に移ります．この例では

- せんべいを買う → 満足度合計６点、残り予算１９０円
- チョコレートを買う → 満足度合計１３点、残り予算６０円
- ポテトチップス、マシュマロ、ガム → いずれも予算オーバー
- 飴を買う → 満足度合計１５点、残り予算１０円となって計算が終了します．

局所探索法



最初に問題の条件を満たす解を持っているとします．
その解の一部（局所的な部分）を修正した解をいろいろ作成し，
その中を探索して現在の解よりもよい解があればそれに移る方法です．
よりよい解が得られ続ける限り反復します．

例） 巡回セールスマン問題における局所探索法

左の図１の巡回路から始め，赤い2本の枝を入れ替えて，真ん中の図２の巡回路を作ると総距離が短くなり，さらに，赤い枝，緑の枝のつなぎ具合を変えて右の図３のよりよい解が得られました．

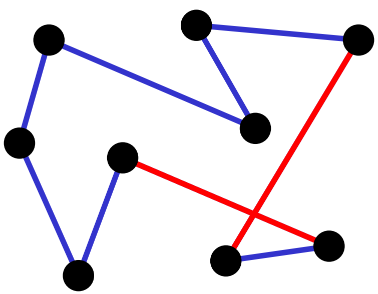


図１

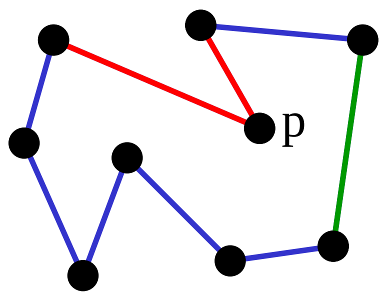


図２

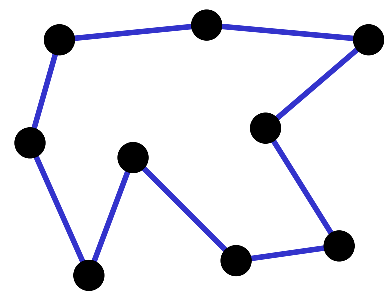


図３

この計算法のデモ，解説について最後の付録１をご覧ください

近似解のよさ

近似解は最適解とどれくらい離れているだろう？
次の近似比は離れ具合を表す尺度です．

近似比：　（近似値 ÷ 最適値）

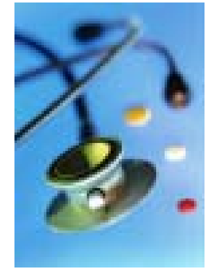
- 最小化問題（目的が値を小さくする問題）なら１以上
  最大化問題（目的が値を大きくする問題）なら１以下
- １に近いほどよい

例）あるナップサック問題の最適値（条件を満たす商品の選び方の満足度合計の最大値）が 100 だったときに，ある方法で求めた近似解の満足度合計が 80 とすると，この近似解の近似比は80/100=0.8となります．

- ナップサック問題に対する欲張り法 &
  巡回セールスマン問題に対する局所探索法
  　→ 近似比が非常に悪くなる例が存在する
  　　（テストの仕方にもよるが，多くの例では近似比はかなり1に近い）

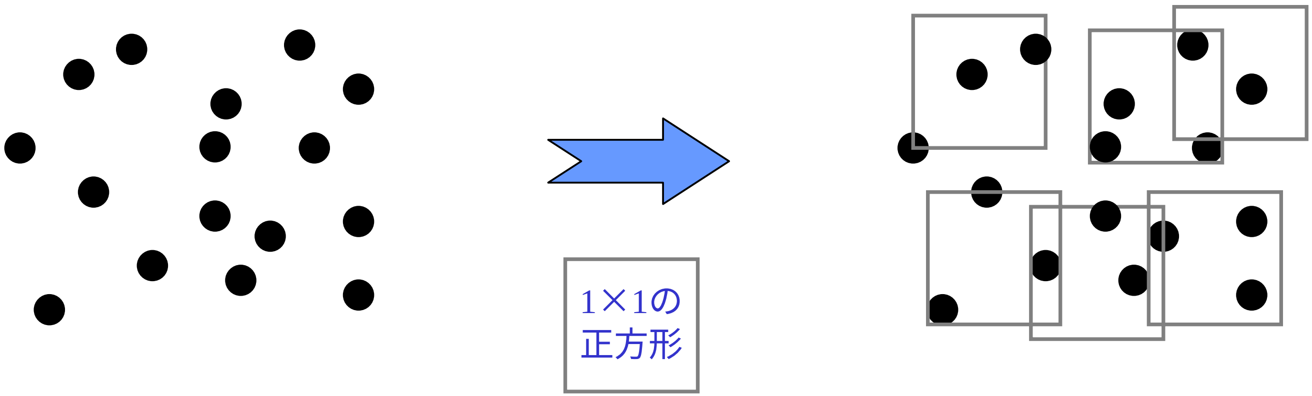
- 近似比が一定の値より悪くならないような近似解を求める計算法が見つかるときもある（これを近似比が保証されるという）．

正方形カバー問題



近似比が理論的に**保証される**方法の例を示しましょう．

平面上に n個の点が与えられ，できるだけ少ない数の正方形ですべての点をカバーしたいという問題を考えてみよう．



ある16点の配置

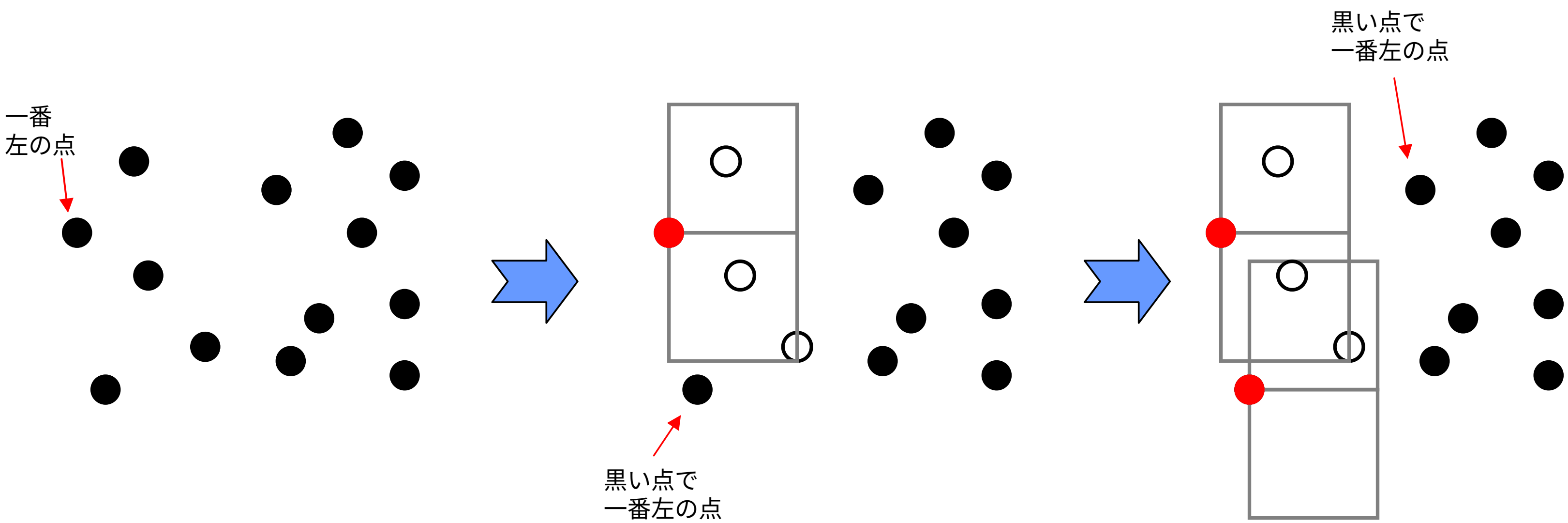
1×1の正方形を6枚で
カバーできている配置例

注： 正方形の辺の上の点はカバーされるとします

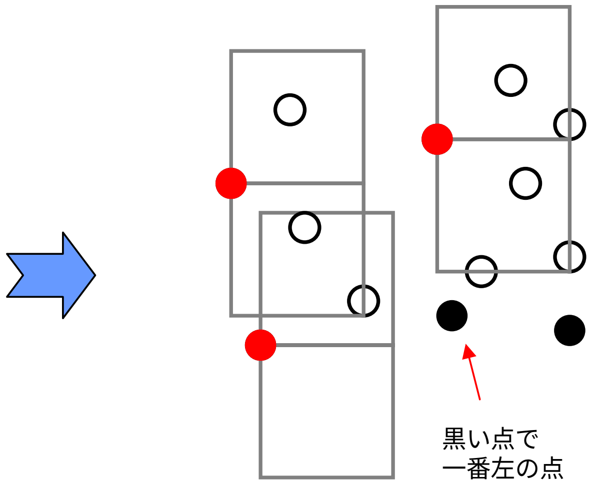
近似比２を保証

次のルールに従って正方形を配置していけば悪くても近似比２の配置が得られます．

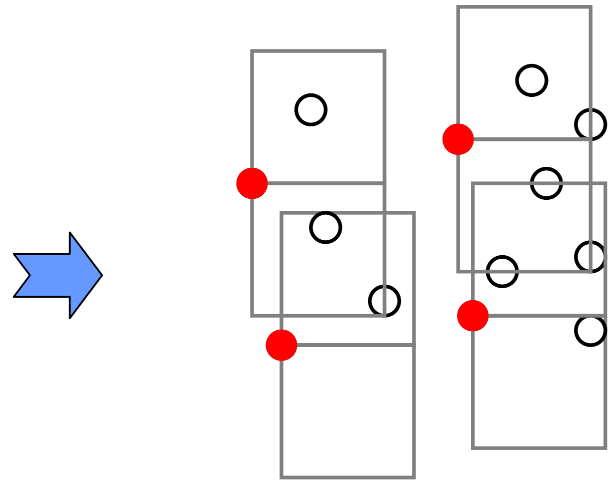
1. 黒い点の中で一番左のものを赤く塗る
2. 赤い点が左下になるように１つ、左上になるように１つ、正方形を置く
3. カバーされた黒い点を白く塗る
4. 黒い点があれば１へ戻る



ある13点の配置　　第１反復後　　第２反復後



第３反復後



第４反復後（終了）
8個の正方形を使った

近似比が２であることの保障

①まず，使った正方形の数は赤い点の数のちょうど２倍ですね．

②計算法のルールから，どの二つの赤い点も正方形１枚ではカバーできません．
このことから，すべての赤い点をカバーするのに同じ数の正方形が必要です．

赤い点の数 ≤ 最適値　ですね．

①，②から，使った正方形の数 =赤い点の数の２倍 ≤ 最適値の２倍.

注： 実際の性能を上げるためにはさらに計算に工夫が施されます

やはり最適解でなければ

これまでの流れをおさらいしておきましょう

- 組合せ最適化問題について
  （例：巡回セールスマン問題、ナップサック問題）
- 列挙法は現実的でなかったですね
- 最適解を求めることは難しいことが多い
- 近似解ならばわりと簡単に計算できる

でしたね．

しかし，近似解ではなく最適解が必要な場合もあります．

以下では，身近な例として 最長しりとり、最長片道切符を取り上げながら最適解計算の最近の成果についてみていきます．

最長しりとり

国語辞典の単語を使ってしりとりを作るとき、
最長で何語までつなげることができるか

テレビ番組「トリビアの泉」のトリビアの種の
コーナーでこの問題は紹介されました（２００４年３月放映）

ある国語辞典の１９２６８７語を用いたときの
最長しりとりの長さは８６７８８語となったそうです．

ここでは説明を省きますが，解き方などの詳細は以下を参照ください

http://al.cs.tuat.ac.jp/~yshinano/shiritori/

数学セミナー ２００４年８月号

最長片道切符

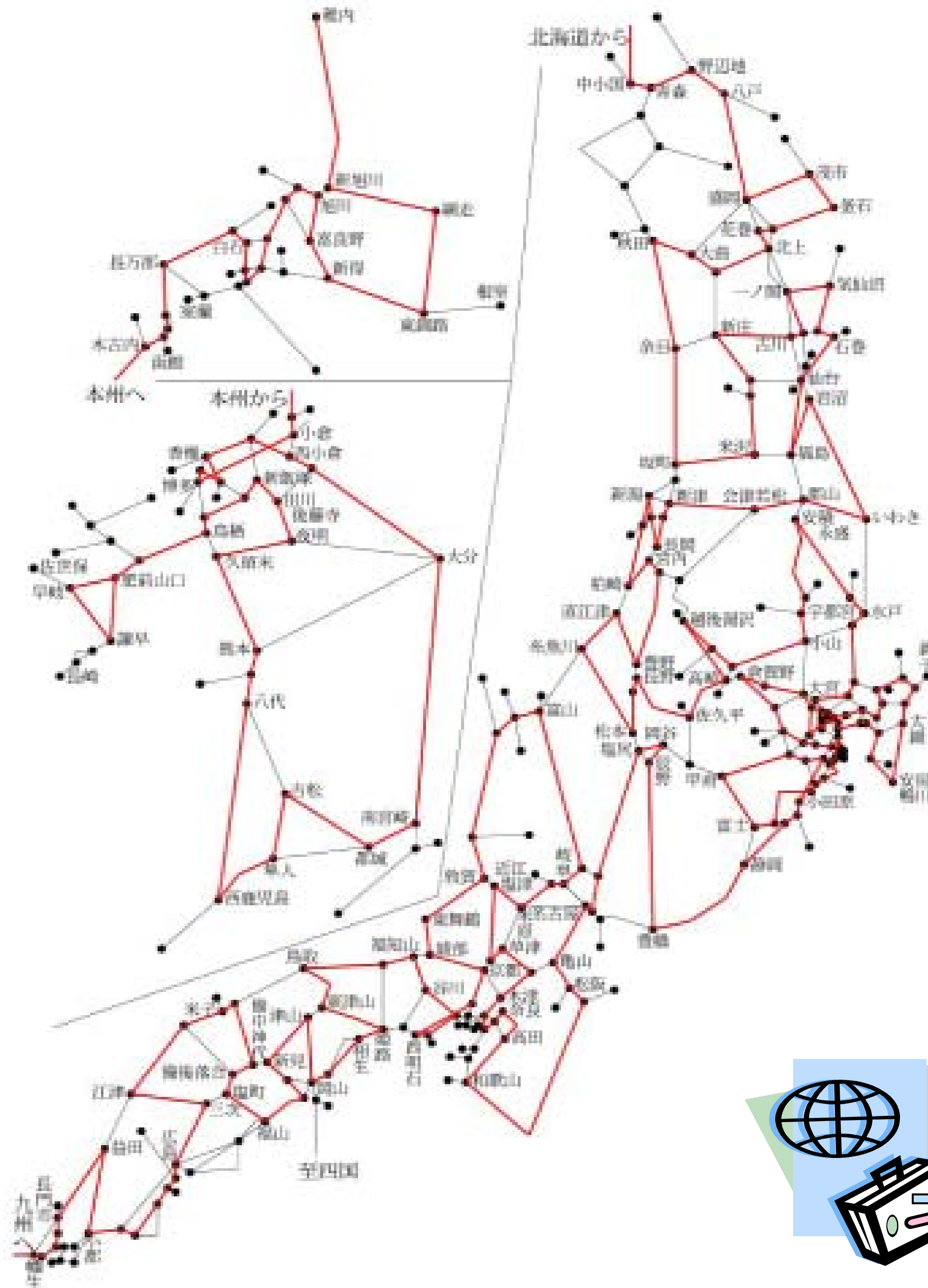
JRの片道切符１枚でできるだけ長い距離を旅行するための経路は？

テレビ番組 「列島縦断
鉄道１２０００ｋｍの旅
最長片道切符で行く４２日」
（２００４年５月－６月放映）

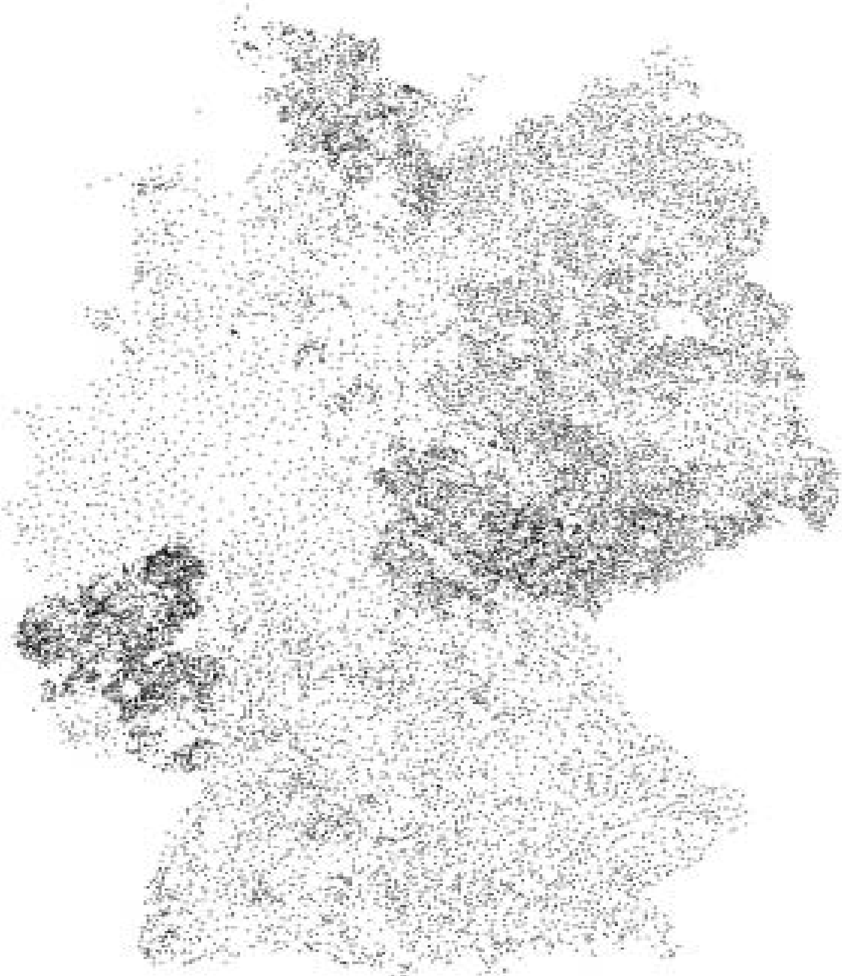
http://www.nhk.or.jp/tabi/

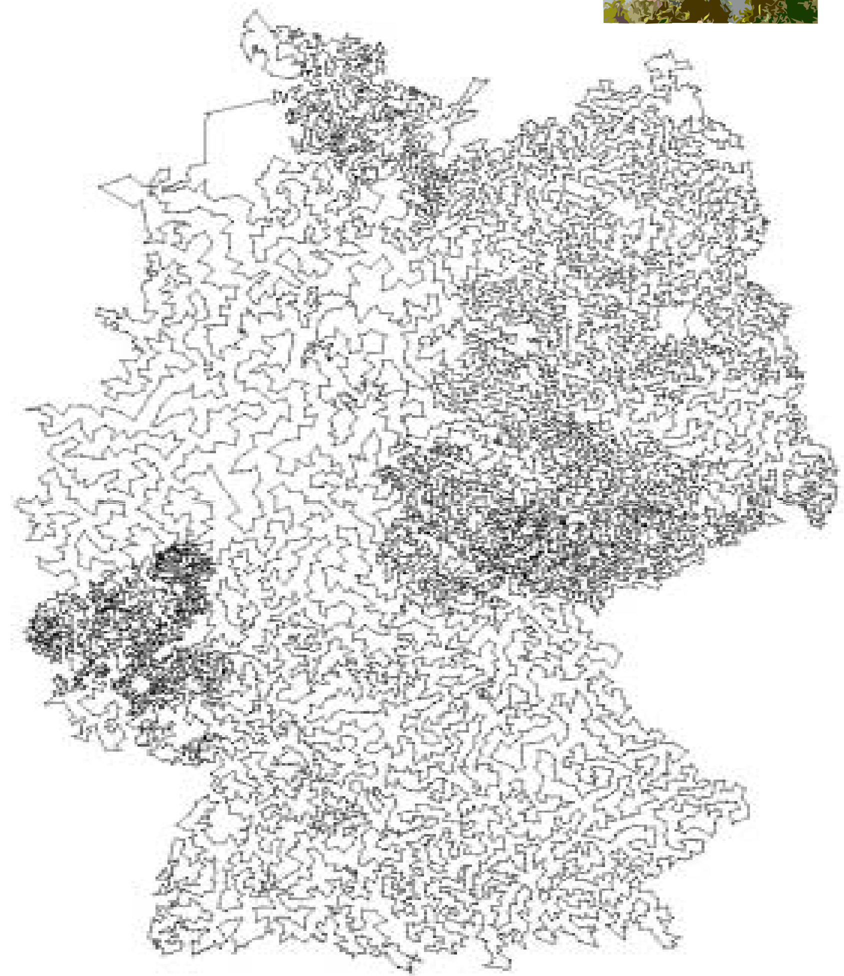
右の路線図で赤色の線が最長の縦断の仕方です．やはり計算法の説明は複雑になるので詳しくは以下をご覧ください．

http://www.swa.gr.jp/lop/
数学セミナー2004年8月号



巡回セールスマン問題の計算例１





ドイツ１５１１２都市の例（図の１点が１都市です）各都市を一度ずつ訪問する順回路で総移動距離が最小であるものは？

上図は最適な巡回路
http://www.tsp.gatech.edu/

巡回セールスマン問題の計算例２

スウェーデンの
２４９７８都市の例

上図は最適な巡回路
http://www.tsp.gatech.edu/

効率的に計算できるものも

これまで出てきた以下の問題は計算量理論の分野ではNP困難問題と
呼ばれる折り紙つきの難しい（最適解を見つけるのが困難な）問題です．

巡回セールスマン問題
ナップサック問題
長方形詰め込み問題
正方形カバー問題
最長しりとり問題
最長片道切符問題

例：スウェーデン２４９７８都市の巡回セールスマン問題

最適解を求めるのに要した計算資源

- PC (Xeon 2.8GHz dual CPU) ９６台
- １つのCPUに換算して８４.８年

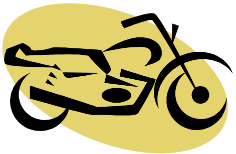
一方で一見難しそうで最適解を効率よく求めることができる
問題もあります．

最短路問題　　文書整形問題

最短路問題

与えられるデータ： n 個の地点と２地点間の距離、および
出発点と目的地

目標： 出発地から目的地までの最短の経路を求める



応用： カーナビのルート検索などたくさんあり

最適解を効率よく求める計算法があります．ダイクストラ法と呼ばれる方法を計算機に組み込めば，百万地点の地図でも瞬時に最短路が得られます．

ダイクストラ法の動作例は，当研究室ホームページのアルゴリズムのデモで確認できます．ぜひ遊んでみてください．
http://www-or.amp.i.kyoto-u.ac.jp/research/demo/

文書整形問題

英文の右端をそろえてきれいに印刷したい

あまり美しくないレイアウト

LAFAYETTE—The Boulder
County Health Department’s La-
fayette office is reducing serv-
ices and hours because of recent state
b u d g e t c u t s t o
public health services.

ハイフンが入った

スペースが広すぎる

良いレイアウト

LAFAYETTE—The Boulder
County Health Department’s
Lafayette office is reducing services
and hours because of recent state
budget cuts to public health services.

ハイフンが入っていない
スペースのバランスがよい

文書整形問題の応用：文書整形ソフト（WORDなどのワープロ）
英字新聞や英語の教科書の印刷

この問題は改行位置の組合せを決めることです．
ハイフンによる単語の分割，
理想的なスペースからのずれ、
などに対するペナルティ値を設定すれば，
総ペナルティ値を最小にする組合せ最適化問題
として扱うことができます．

この問題には自分自身を小さな問題に分解できるという性質があり，動的計画法と呼ばれる方法を使うことで，最適解（最適な改行位置の組合せ）を効率よく求められる ことが知られています．

まとめ

最後にいままで見てきたことをまとめておきます．

- 組合せ最適化問題
  （巡回セールスマン問題、ナップサック問題）
- 列挙法は現実的でない（指数関数の爆発）
- 最適解を求めることは難しい（ＮＰ困難）
- 近似解ならばわりと簡単に計算できる
  （欲張り法、局所探索法）
- 近似解のよさを理論的に保証
  （正方形カバー問題　近似比２）
- 近似解ではなく最適解が必要な場合もある
  （最長しりとり、最長片道切符）
- 最適解を効率よく計算できる問題もある
  （最短路問題、文書整形問題）

おわりに

以上の説明をご覧頂いた皆さん，お疲れ様でした．組合せ最適化や計算の工夫に興味を持って頂けたでしょうか．やや難しい話もあったかもしれませんが，ここで紹介した話が頭に入っていれば組合せ最適化初級コースは卒業です！　ここのホームページにはアルゴリズムのデモも掲載してありますので興味がある方はそちらもご覧ください．

この説明は，２００４年８月の京都大学オープンキャンパスの高校生向け講義を基にして作られています．現在，数理工学コースから高等学校への出前講義が準備されていますが，その一つにこの内容での講義が含まれています．

付録１：巡回セールスマン問題に対する局所探索法

現在の解を元に修正した解をいろいろ作る際，解を修正する度合いが大きいとあまりにたくさんの解を生成することになるので，通常，解のごく一部だけ（局所的な部分）を修正します．

　巡回セールスマン問題では，それぞれ 2-opt, Or-opt と呼ばれる修正の仕方が良く使われます．

2-opt: 2本の枝（線分）を入れ替える修正

Or-opt: 通り道の一部を巡回路の他の部分へ移す修正（例えば，巡回路上の１本の枝や１個の地点を，他の通り道の途中で訪れるように変えること）

本文の例では，図１の巡回路に2-opt修正を適用して，赤い2本の枝を入れ替えて図２の巡回路が得られています．次に，Or-optを適用して地点pを赤い通り道から緑の通り道に移すとことで図３の巡回路が得られています．

巡回セールスマン問題に対する局所探索法の動作例はアルゴリズムのデモのページでご覧いただけます．

http://www-or.amp.i.kyoto-u.ac.jp/research/demo/