大規模な組合せ最適化問題に対する発見的解法

大阪大学 大学院情報科学研究科
科学技術振興機構
梅谷　俊治

2014年3月12日
数学協働プログラムチュートリアル
「ビッググラフと最適化」

講演の概要

- かつてはITインフラの整備，今は大規模データの解析，将来は効率的な計画の立案・運用が重要な課題になる？
- 実世界から収集された大規模データに基づく大規模かつ多様な組合せ最適化問題を効率良く解くことが求められる？
- これらの組合せ最適化問題の多くはNP困難で，現場では経験と勘に基づくアルゴリズム設計・開発が主流？
- 狭い範囲の事例にしか適用できないノウハウではなく，広い範囲の事例に適用できるアルゴリズム設計のノウハウは無いのだろうか？

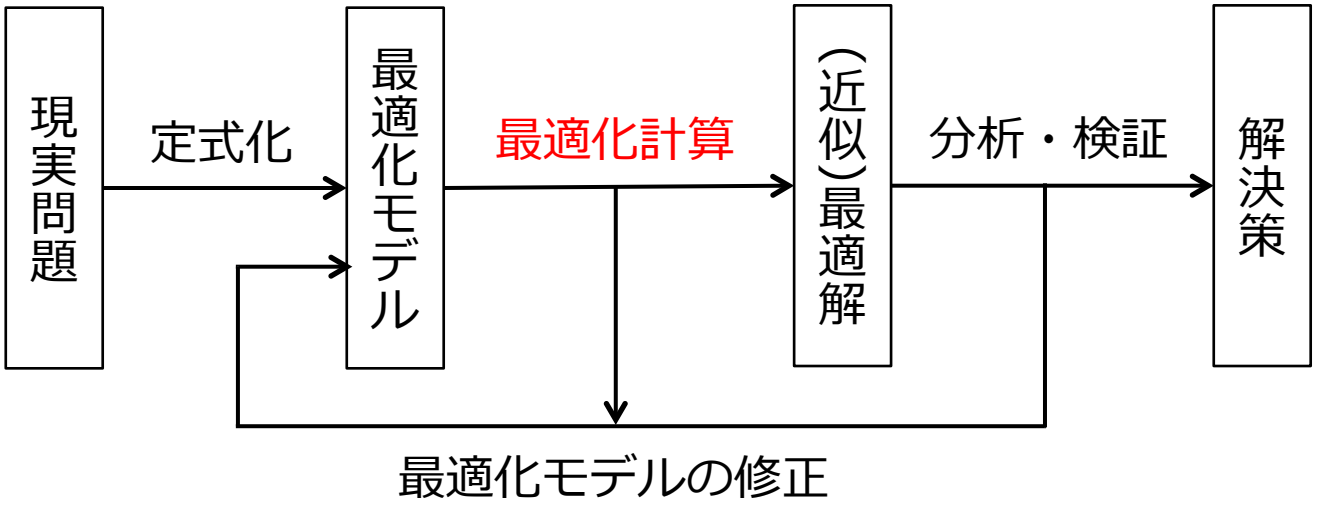
- 本講演では以下の話題を紹介します．
  - 組合せ最適化問題とその応用
  - 計算困難な組合せ最適化問題に対するアプローチ
  - 大規模な組合せ最適化問題に対する発見的解法

講演途中でいくつかスライドを飛ばすかも知れませんが，本講演のスライドはウェブ上で公開する予定なのでご安心下さい．

組合せ最適化問題とその応用

最適化手法による問題解決アプローチ

- 最適化は意思決定・問題解決のための一つの手段.
- 最適化問題に定式化＋最適化計算＋(近似)最適解の分析・検証＋最適化モデルの修正.



最適化問題
制約条件を満たす解の中で目的関数を最小(最大)にする解を求める問題.

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & x \in F \end{aligned}$$

最適化問題に定式化する

生産計画問題

あるシャトーでは3種類のブドウ，カルベネ，メルロー，セミヨンを原料として，3種類のワイン，赤ワイン，白ワイン，ロゼワインを製造している．収益が最大となる各ワインの1日当たりの製造量を求めよ．

種類	赤ワイン	白ワイン	ロゼワイン	最大供給量
カルベネ	2	0	0	4(t/日)
メルロー	1	0	2	8(t/日)
セミヨン	0	3	1	6(t/日)
収益	3	4	2	(百万円/日)

$$
\begin{aligned}
\text{maximize} \quad & 3x_1 + 4x_2 + 2x_3 && \rightarrow \text{収益を最大化} \\
\text{subject to} \quad & 2x_1 \leq 4, && \rightarrow \text{カルベネの使用量は4t/日以内} \\
& x_1 + 2x_3 \leq 8, && \rightarrow \text{メルローの使用量は8t/日以内} \\
& 3x_2 + x_3 \leq 6, && \rightarrow \text{セミヨンの使用量は6t/日以内} \\
& x_1, x_2, x_3 \geq 0. && \rightarrow \text{各ワインの製造量は非負}
\end{aligned}
$$

線形計画問題と整数計画問題

- 線形計画問題(Linear Program; LP)
  - 線形制約の下で線形関数を最小化（最大化）

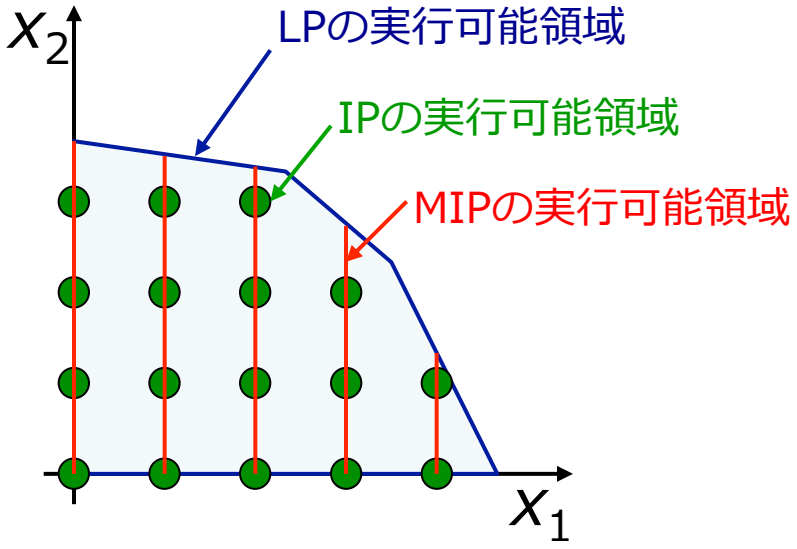
- 整数計画問題(Integer Program; IP)
  - 広義：変数に整数条件の付いた最適化問題
  - 狭義：線形計画問題＋変数の整数条件

連続変数と整数変数が混在する問題は，混合整数計画問題(Mixed Integer Program; MIP)

$$\text{min.} \quad \sum_{j=1}^{n} c_j x_j$$

$$\text{s.t.} \quad \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = 1, \ldots, m,$$

$$x_j \geq 0, \quad j = 1, \ldots, n.$$



$\min\{\mathbf{c}^T\mathbf{x} \mid \mathbf{Ax} \geq \mathbf{b}, \mathbf{x} \geq 0\}$ のようにベクトルと行列を用いて書く場合も多い.

組合せ最適化問題

- 探索空間が離散的であるもしくは離散的なものに減らせる最適化問題，解が集合，順序，割当て，グラフ，論理，整数などの離散構造で記述される場合が多い．
- 多くの組合せ最適化問題は整数計画問題として定式化できる．

組合せ最適化問題の例

- 最短路問題(カーナビのルート検索，乗換案内など)
- ネットワーク設計問題(ライフライン，交通・通信網，石油・ガスのパイプライン網の設計など)
- 配送計画問題(宅配便，店舗・工場への製品配送，ゴミ収集など)
- 施設配置問題(工場，店舗，公共施設など)
- スケジューリング問題(工場の操業計画，乗務員・看護師などの勤務表，スポーツの対戦表・日程表，時間割の作成など)

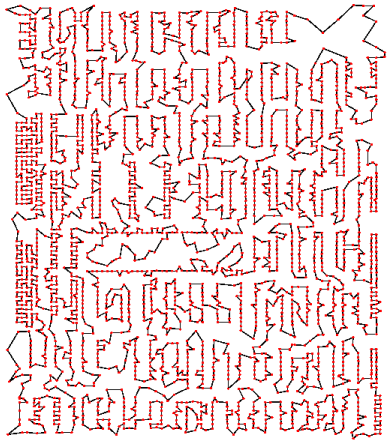
現実問題の多くが組合せ最適化問題として定式化できる！

整数計画問題に定式化した方が解き易い場合もあれば，そうでない場合もある．

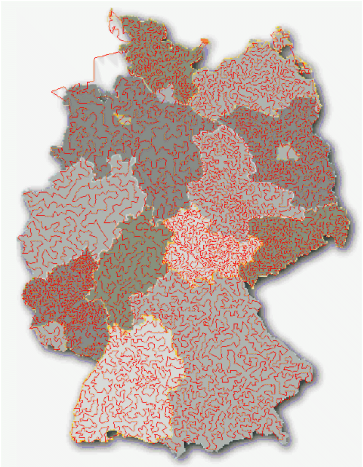
巡回セールスマン問題

- 都市の集合と各2都市間の移動コスト(例えば距離)が与えられたとき，全ての都市をちょうど一度ずつ訪問し，出発地に戻る巡回路の中で総移動コストが最小のものを求める問題.

Newsweek, July 26, 1954



PCB3038



D15112

d_{ij} : 点(i,j)間の距離(定数)

x_{ij} : 枝(i,j)を通る$\rightarrow x_{ij}=1$,

通らない$\rightarrow x_{ij}=0$(変数)

そのまま解くには制約式が多過ぎる

$$\text{min.} \quad \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_{ij}$$

$$\text{s.t.} \quad \sum_{j:j\neq i} x_{ij} = 1, \quad i = 1,\dots,n,$$

$$\sum_{j:j\neq i} x_{ji} = 1, \quad i = 1,\dots,n,$$

$$\sum_{i\in S}\sum_{j\notin S} x_{ij} \geq 1, S \subset \{1,\dots,n\}, S \neq \emptyset,$$

$$x_{ij} \in \{0,1\}, \quad i = 1,\dots,n, j = 1,\dots,n.$$

出典: The Traveling Salesman Problem (http://www.math.uwaterloo.ca/tsp/index.html)

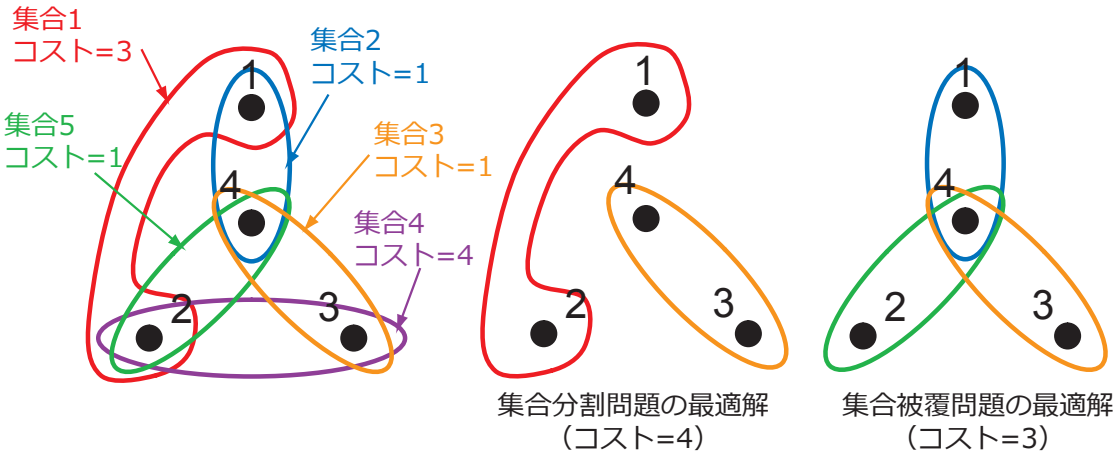
集合分割問題

- m個の要素からなる集合Mと，n個の部分集合 $S_j \subseteq M$ が与えられたとき，集合Mの全ての要素 $i \in M$ をちょうど1回ずつ含む部分集合の組み合わせの中で，コストの総和が最小になるものは？

a_{ij}：部分集合S_jが要素iを含む→$a_{ij}=1$，含まない→$a_{ij}=0$(定数)

c_j：部分集合S_jのコスト(定数)

x_j：部分集合S_jを選ぶ→$x_j=1$，選ばない→$x_j=0$(変数)



$$\min \sum_{j \in N} c_j x_j$$

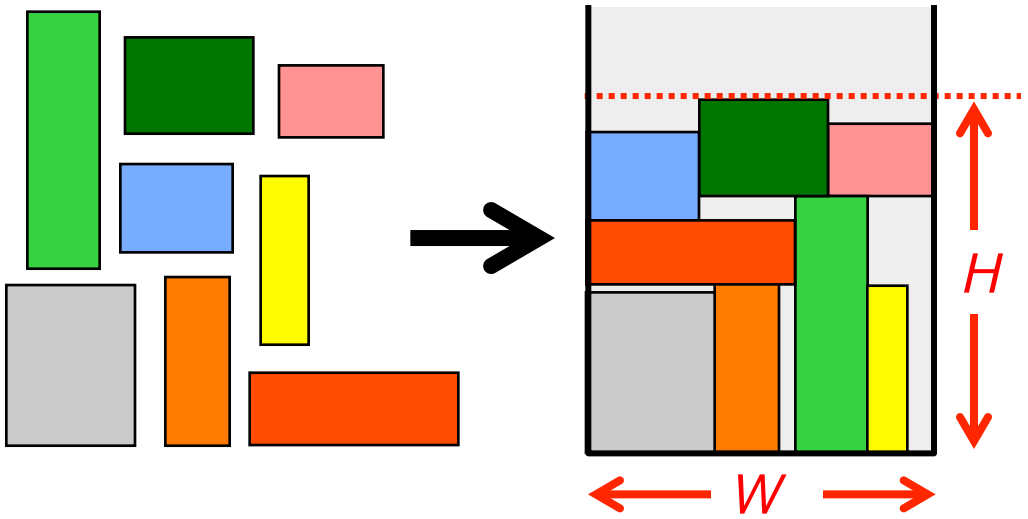
$$\text{s.t.} \sum_{j \in N} a_{ij} x_j = 1, \quad i \in M,$$

$$x_j \in \{0, 1\}, \quad j \in N.$$

集合分割問題は，配送計画，シフトスケジューリング，施設配置，選挙区割り，データ解析など多くの現実問題を応用に持つ．

長方形詰込み問題

- 幅が固定で十分な高さのある長方形の容器とn個の長方形の荷物が与えられる.
- 荷物を互いに重ならないように容器内に配置する制約の下で必要な容器の高さを最小にするには？



$\{1, \ldots, n\}$： 長方形の集合
w_i： 長方形iの幅
h_i： 長方形iの高さ
W： 容器の幅
H： 容器の高さ

※荷物の自由な回転は考えない

詰込み・切出し問題は，長方形詰込み問題，円詰込み問題，コンテナ詰込み問題，多角形詰込み問題，など図形の次元や形状によりさまざまなバリエーションを持つ.

長方形詰込み問題

- (x_i, y_i)を長方形iの左下隅の座標とする.
- 制約1: 長方形iは容器内に配置される.

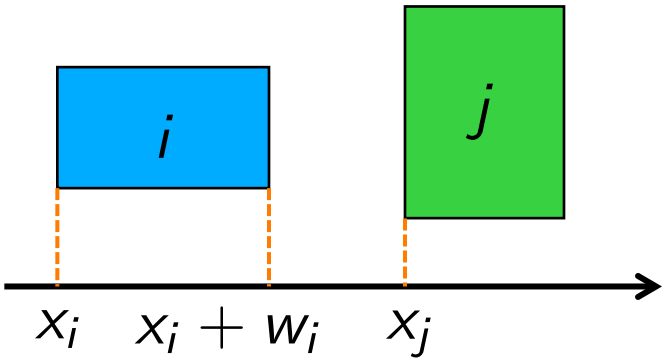
$$0 \le x_i \le W - w_i$$

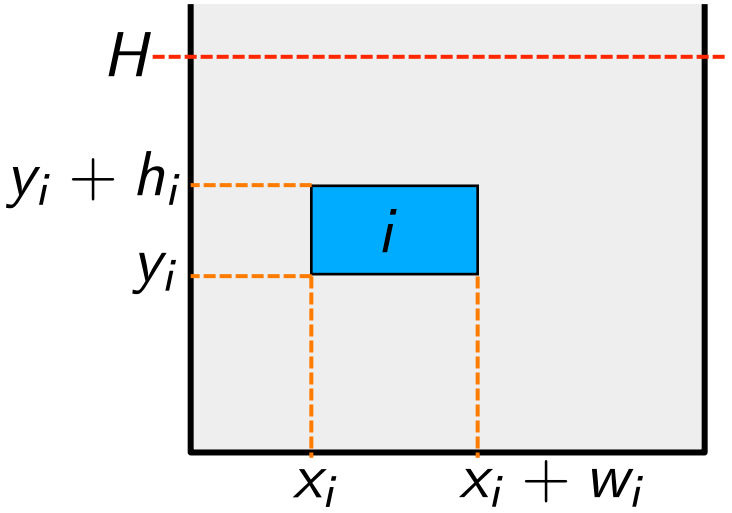
$$0 \le y_i \le H - h_i$$

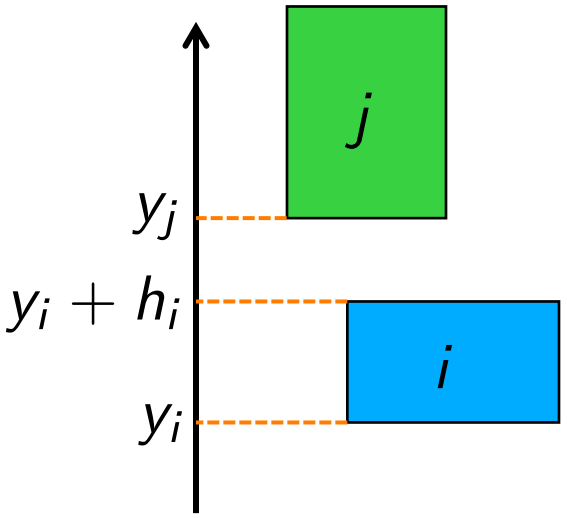
- 制約2: 長方形i,jは互いに重ならない.

$$(x_i + w_i \le x_j) \vee (x_j + w_j \le x_i)$$

$$\vee (y_i + h_i \le y_j) \vee (y_j + h_j \le y_i)$$







長方形詰込み問題

- $z_{ij}^{left}, z_{ij}^{right}, z_{ij}^{lower}, z_{ij}^{upper}$ をそれぞれ長方形iが長方形jの左，右，下，上にあるならば1，そうでなければ0を取る変数とする．

$$\min H$$

$$\text{s.t. } 0 \le x_i \le W - w_i,\ i = 1, \dots, n,$$

$$0 \le y_i \le H - h_i,\ i = 1, \dots, n,$$

そのまま解くには制約式が多過ぎる

$$x_i + w_i \le x_j + M(1 - z_{ij}^{left}),\ i = 1, \dots, n,\ j \ne i,$$

$$x_j + w_j \le x_i + M(1 - z_{ij}^{right}),\ i = 1, \dots, n,\ j \ne i,$$

$$y_i + h_i \le y_j + M(1 - z_{ij}^{lower}),\ i = 1, \dots, n,\ j \ne i,$$

$$y_j + h_j \le y_i + M(1 - z_{ij}^{upper}),\ i = 1, \dots, n,\ j \ne i,$$

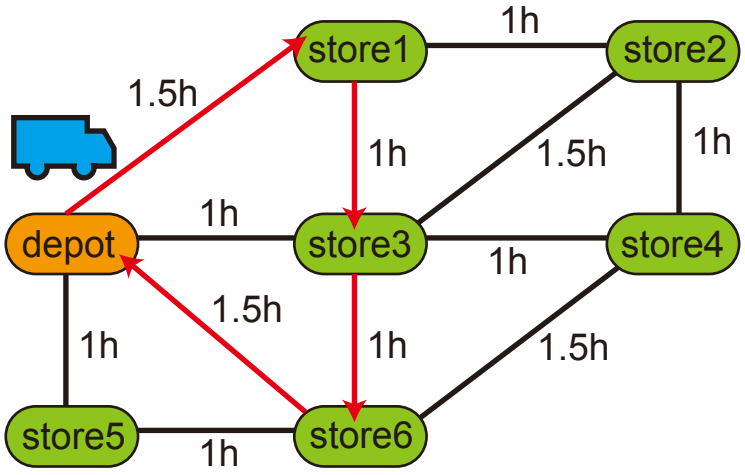
$$z_{ij}^{left} + z_{ij}^{right} + z_{ij}^{lower} + z_{ij}^{upper} = 1,\ i = 1, \dots, n,\ j \ne i,$$

$$z_{ij}^{left}, z_{ij}^{right}, z_{ij}^{lower}, z_{ij}^{upper} \in \{0, 1\},\ i = 1, \dots, n,\ j \ne i.$$

※Mは十分に大きな定数

応用：配送計画

- mカ所の店舗に商品を配送するのに必要なトラックの台数を最小にするような配送ルートの組み合わせは？
- １台のトラックが時間内に配送できるルートを全て列挙すると集合分割問題として定式化できる.



6時間で配送可能な経路を全て列挙

$$\boldsymbol{c} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

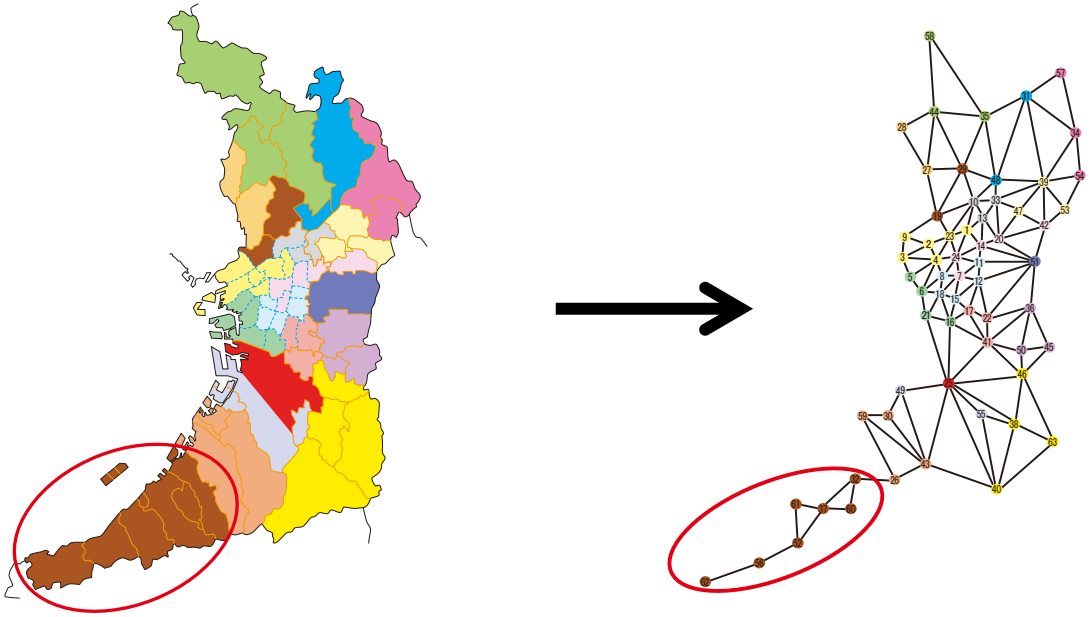
$x_j = 1$

各列が対応するルートを表す

勤務スケジュール作成問題や時間割問題も定式化できる！

応用：選挙区割り

- m個の市区郡を一票の格差が最小になるようにn個の選挙区に分割するには？(飛び地を持つ選挙区を作ってはならない)
- 市区郡を頂点とするグラフを描いて，連結な頂点の部分集合を全て列挙すると集合分割問題として定式化できる.

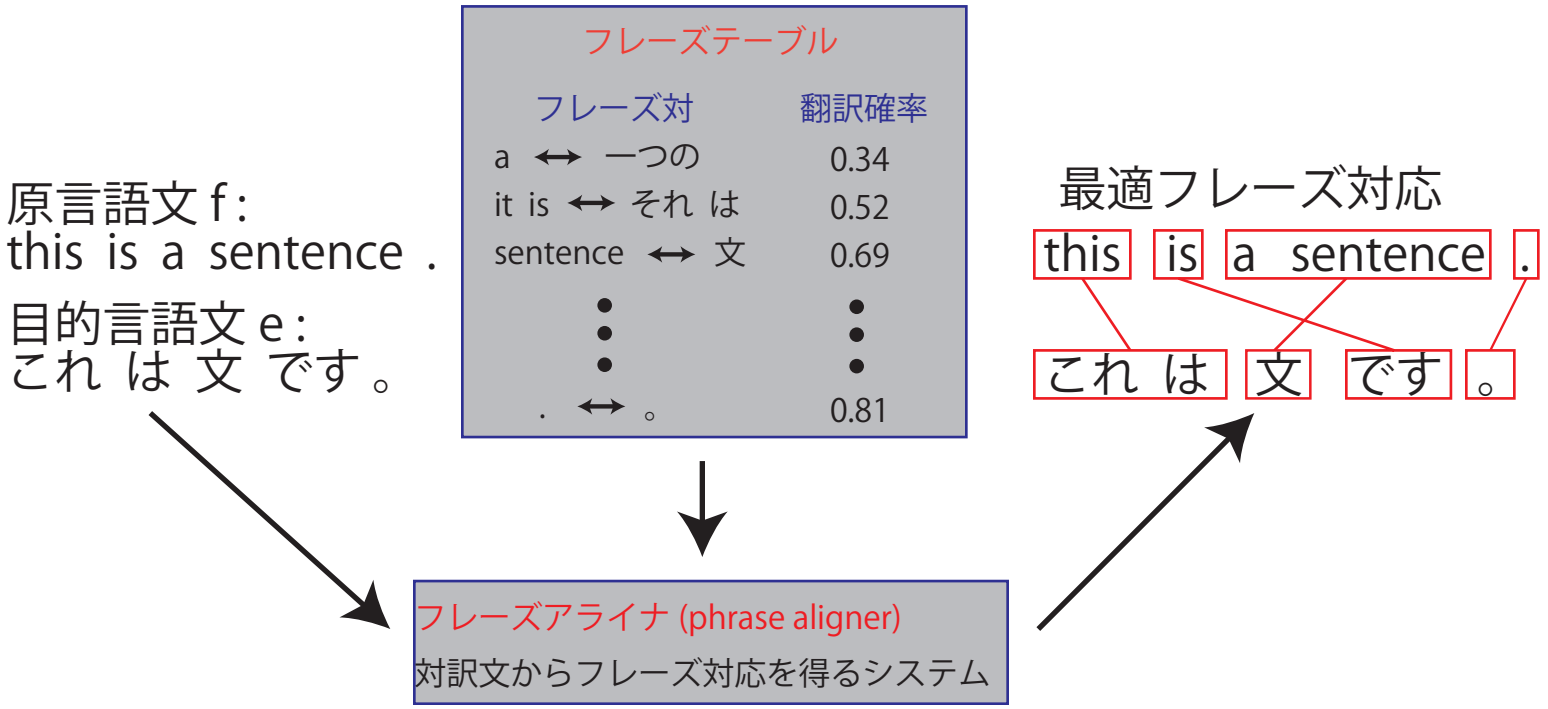


施設配置問題やネットワーク設計問題も定式化できる！

根本俊男，堀田敬介，区割画定問題のモデル化と最適区割の導出，オペレーションズ・リサーチ，**48**(2003),300-306.

応用：機械翻訳におけるフレーズ対応

- 大量の対訳文から確率モデルに基づく翻訳規則を学習する.
- 連続する単語列(フレーズ)を最小単位として翻訳する.
- 確率が最大となる入力文のフレーズ区切りと翻訳前後の訳語関係(フレーズ対応) を求める.



越川 満, 内山 将夫, 梅谷 俊治, 松井 知己, 山本 幹雄: 統計的機械翻訳におけるフレーズ対応最適化を利用したN-best翻訳候補のリランキング, 情報処理学会論文誌, **51** (2010), 1443-1451.

応用：機械翻訳におけるフレーズ対応

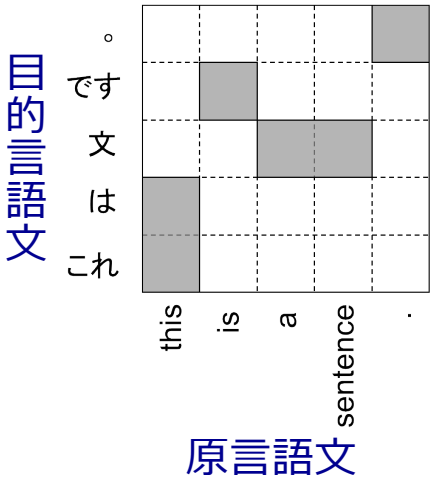
- 塗りつぶした矩形の縦横に重複が起きないよう全単語を被覆する.
- フレーズ対応問題は集合分割問題として定式化できる.

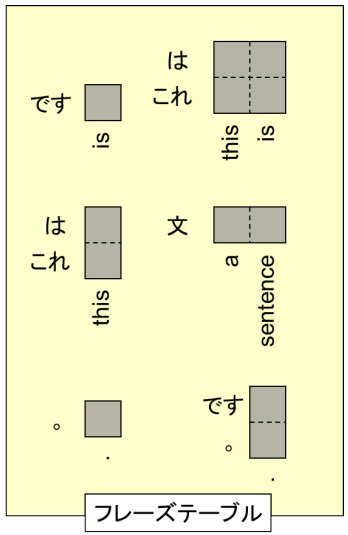
t_k：フレーズ対kの翻訳スコア(定数)

f_{ki}：フレーズ対kが原言語文のi番目の単語を被覆(定数)

e_{kj}：フレーズ対kが目的言語文のj番目の単語を被覆(定数)

x_k：フレーズ対kを使用する→$x_k=1$，使用しない→$x_k=0$ (変数)





$$\max. \quad \sum_{k \in K} t_k x_k$$

$$\text{s.t.} \quad \sum_{k \in K} f_{ki} x_k = 1, \quad i \in I,$$

$$\sum_{k \in K} e_{kj} x_k = 1, \quad j \in J,$$

$$x_k \in \{0,1\}, \quad k \in K.$$

文書要約問題も定式化できる！

応用：推薦商品の最適化

- 顧客の過去の購入履歴からその嗜好を推測はできるが，実際には実務上の様々な制約の下で，顧客に商品を推薦する必要がある．

C_{ij} :顧客iに商品jを推薦したときの期待費用(定数)
G_{ij} :顧客iに商品jを推薦したときの期待利得(定数)
P_j :商品jの総期待利得の下限(定数)
$C_{\max}$:総期待費用の上限(定数)
N :顧客1人に推薦する商品数(定数)
x_{ij} :顧客iに商品jを推薦する→x_{ij}=1，しない→x_{ij}=0(変数)

$$\max \quad \sum_{i\in I}\sum_{j\in J} G_{ij}x_{ij}$$

$$\text{s.t.} \quad \sum_{i\in I}\sum_{j\in J} C_{ij}x_{ij} \le C_{\max}$$

$$\sum_{i\in I} G_{ij}x_{ij} \ge P_j, j \in J,$$

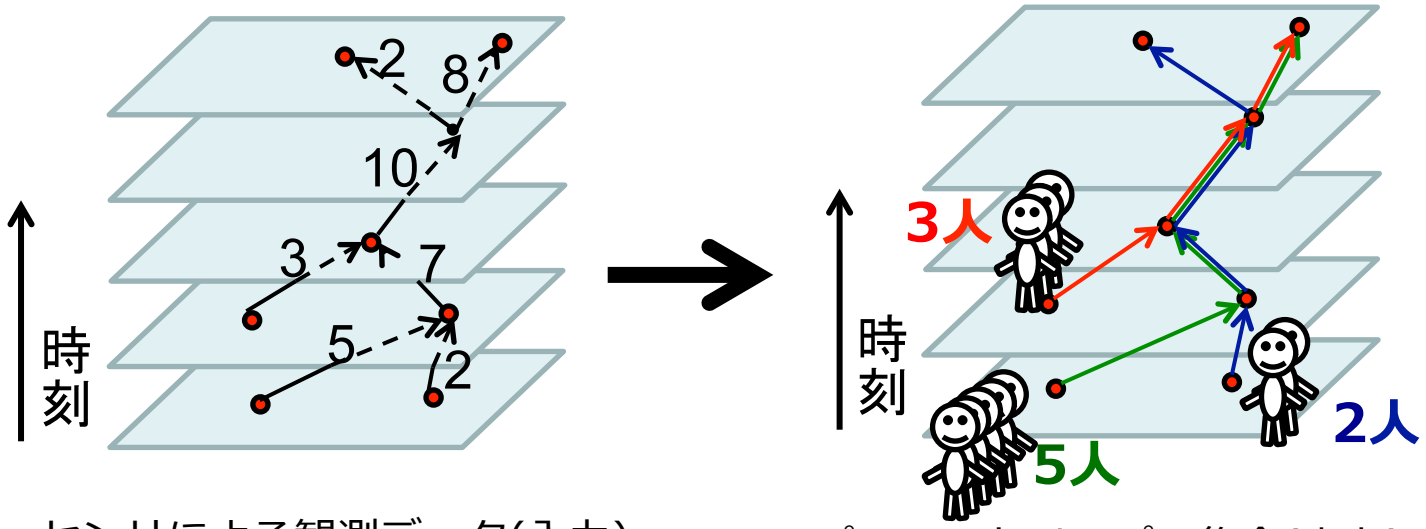
$$\sum_{j\in J} x_{ij} = N, i \in I,$$

$$x_{ij} \in \{0,1\}, i \in I, j \in J.$$

土谷拓人，西村直樹，鮭川矩義，高野祐一，中田和秀，松本健:大規模な推薦商品最適化問題に対する確率的勾配法，日本オペレーションズ・リサーチ学会春季研究発表会，2014年3月6日.

応用：人の移動履歴の推定

- 人の移動履歴を収集・加工したデータベースを用いて時々刻々と変化する人の流れを解析する研究や調査が盛ん.
- 各自治体のパーソントリップ調査やSuicaなどのデータがあるが任意の目的に利用することは困難.
- 地域内の各地点に配置されたセンサーで観測される通過人数の履歴から人の移動履歴を推定できないだろうか？

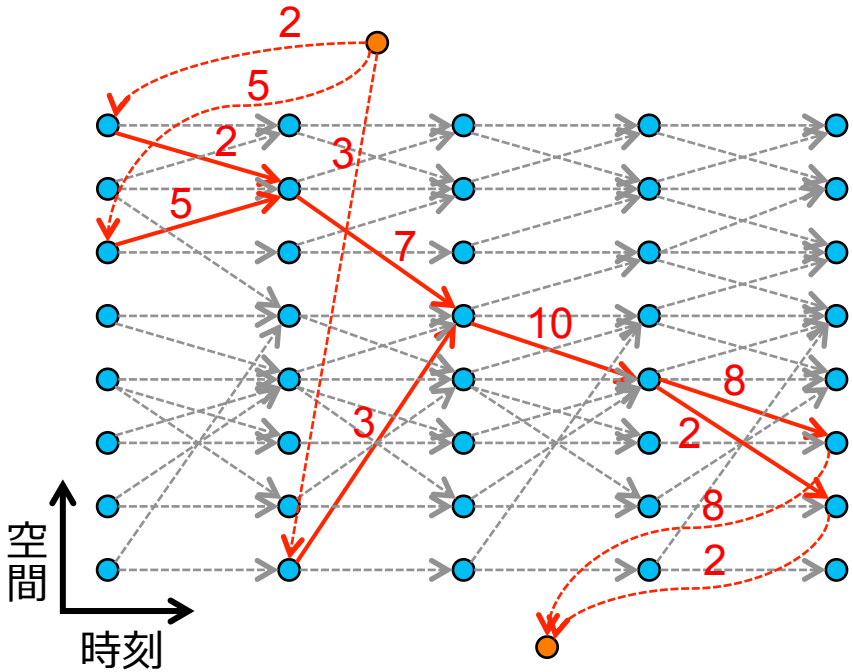


センサによる観測データ(入力)　　パーソントリップの集合(出力)

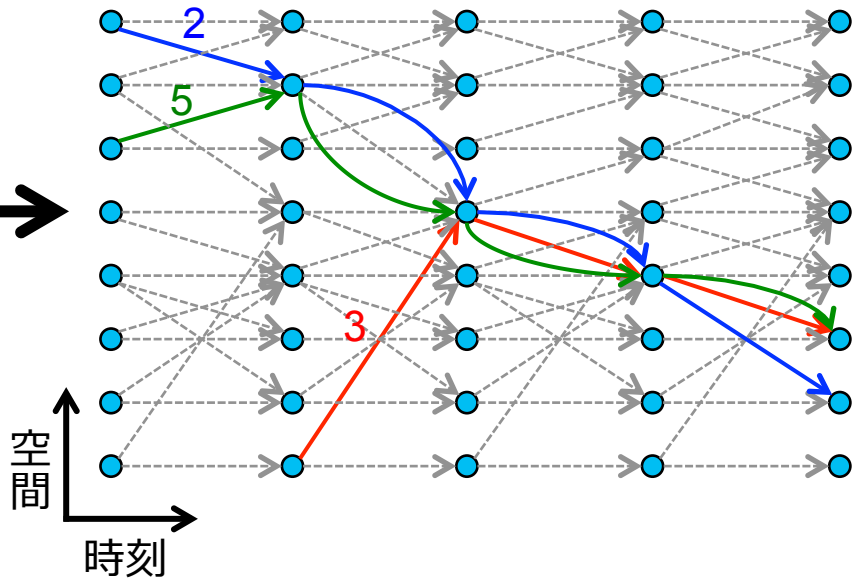
熊野徹，蓮池隆，梅谷俊治，森田浩:観測データに基づく人の移動履歴の推定，日本オペレーションズ・リサーチ学会春季研究発表会，2014年3月6日.

応用：人の移動履歴の推定

- 入力：ネットワーク外部との間の流入出者数，各枝の移動人数，パス候補の集合P
- 出力：各パスの利用者数 $x_j\ (j \in P)$
- 観測データと出力の誤差を最小化



観測データ(入力)

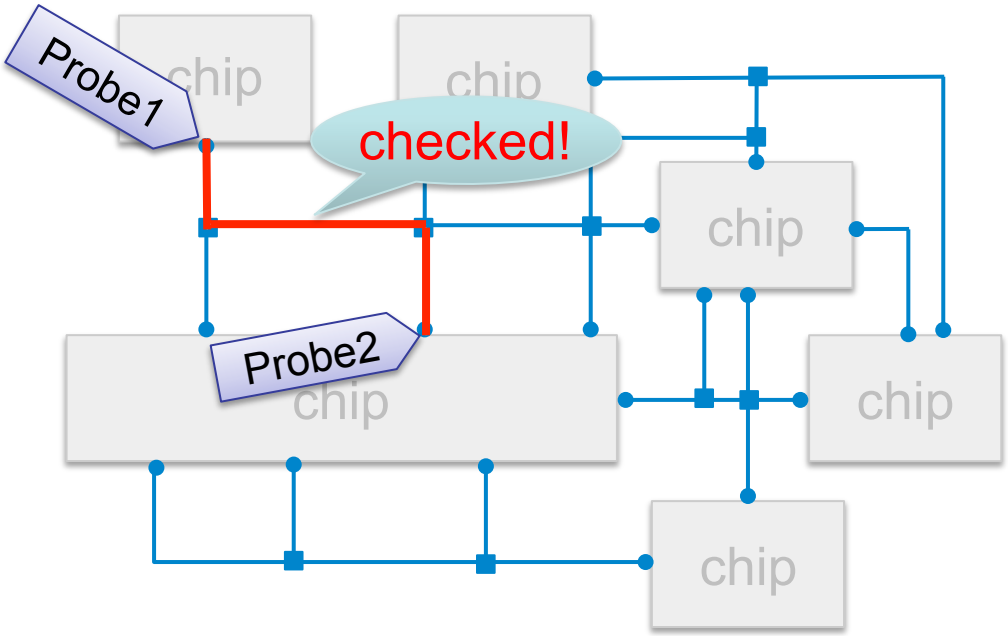


人の移動履歴(出力)

multi-target tracking と呼ばれる多くの分野に現れる問題

応用：基板検査プローブの経路計画

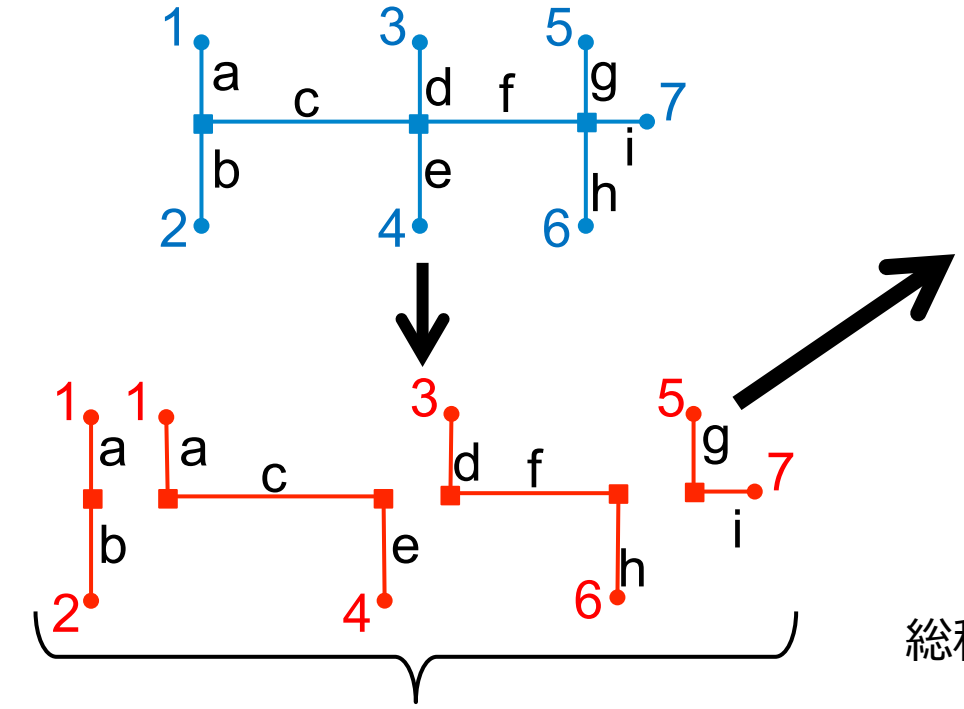
- 2本の移動型プローブを用いてプリント基板の導通を検査.
- 全ての導線の導通検査にかかる時間を最小化.
- 全ての導線を導通検査するのに必要な2本のプローブの訪問先と訪問順を決定.



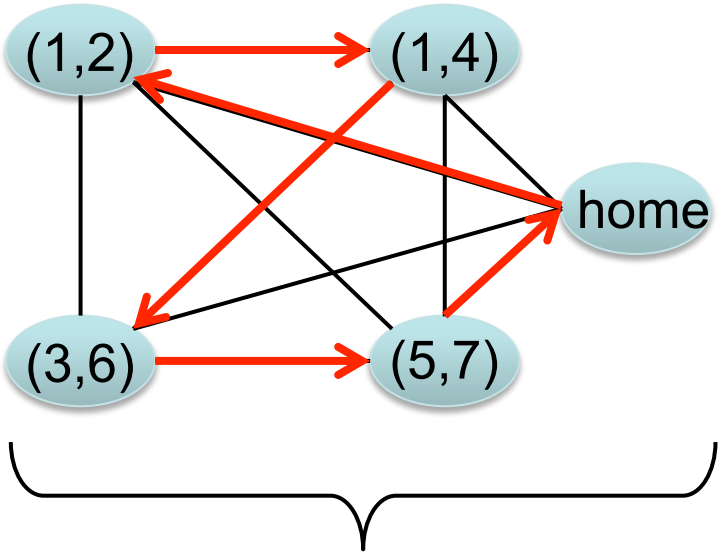
K.Murakami, Formulation and heuristic algorithms for multi-chip module substrate testing, *Computers and Electrical Engineering*, **39** (2013), 1049-1060.

応用：基板検査プローブの経路計画

1. 制約付き最短路問題に定式化 → ラベリング法
2. 集合被覆問題＋巡回セールスマン問題の2段解法
   - 集合被覆問題：全ての導線の導通を検査するプローブの訪問先(端子ペアの集合)を決定
   - 巡回セールスマン問題：総移動距離が最短となるプローブの訪問順を決定



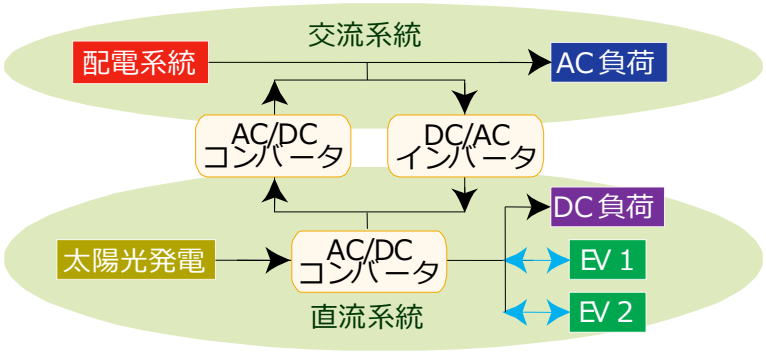
全ての導線を検査するプローブの訪問先

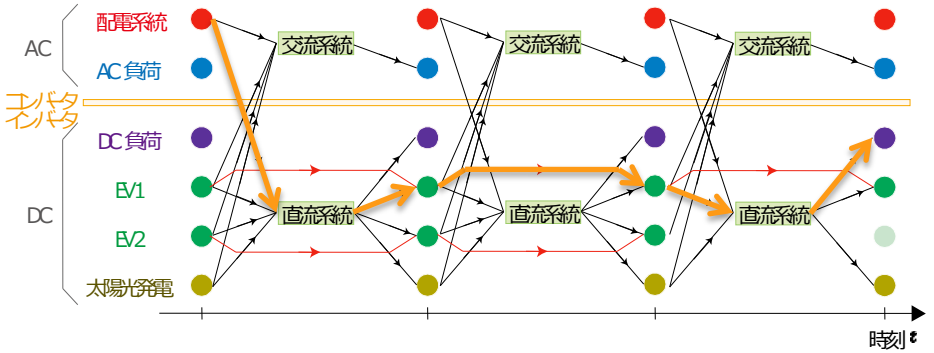


総移動距離が最小になるプローブの訪問順

応用：電気自動車の充放電計画

- 電気自動車に充放電させて蓄電池の代替として利用する.
- 電気自動車を適切なタイミングで充放電させることで，ピーク時間帯での配電系統からの受電量の平準化と低減を実現する.
- 電気自動車は放電時には直流電源，充電時には負荷となり，出庫時には電力ネットワークから離脱する.
- 電力ネットワークを時間軸方向に拡張して，各枝のフロー量を決定する(混合)整数計画問題として定式化する.





森田浩，小林美佐世: 電気自動車インフラの効率的な運用と設計，システム/制御/情報，**56** (2012)，427-432.

計算困難な組合せ最適化問題に対するアプローチ

巡回セールスマン問題の難しさ

- *n*都市に対して(*n*-1)!/2通りの巡回路を列挙すれば巡回セールスマン問題は解ける？
- 実際に計算機で数え上げるとどれぐらい時間がかかる？

都市数	巡回路の総数	計算時間	
6	60	4.32×10^{-10}秒	
8	2520	3.23×10^{-8}秒	
10	1.81×10^{5}	3.63×10^{-6}秒	
15	4.36×10^{10}	1.96秒	
20	6.08×10^{16}	4.87×10^{6}秒	→約56日
25	3.10×10^{23}	3.88×10^{13}秒	→約122万年
30	4.42×10^{30}	7.96×10^{20}秒	→約25233億年

100TFlops(1秒間に10^{13}回四則演算できる)のコンピュータの場合

フカシギの数え方@YouTube

小規模の問題なら良いデータ構造を使えば数え上げできますが…

全ての解を列挙することは現実的には不可能（組合せ爆発）

出典：JST ERATO 湊離散構造処理系プロジェクト(http://www-erato.ist.hokudai.ac.jp/)

組合せ最適化問題の難しさ

- 易しい問題
  - 最悪の場合でも入力サイズに対する多項式時間で厳密な最適解を求めるアルゴリズムが存在する問題.
  - 割当問題→ハンガリー法，最小全域木問題→クラスカル法，最短路問題→ダイクストラ法など.

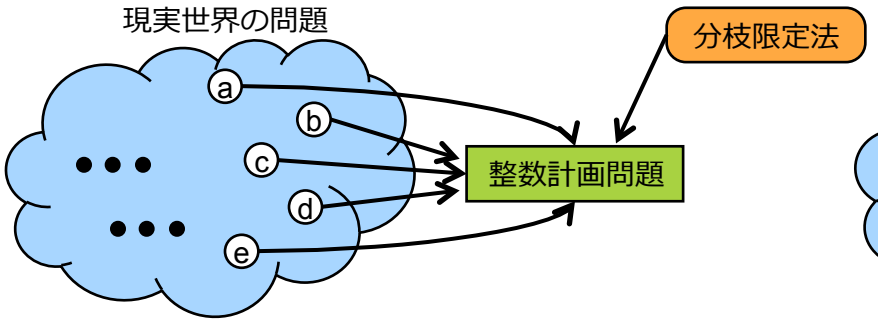
- 難しい問題
  - 厳密な最適解を求めるのに最悪の場合に入力サイズに対して指数時間を要すると多くの研究者が考えている問題.
  - NP困難問題と呼ばれる.
  - ナップサック問題，巡回セールスマン問題，整数計画問題など.

現実社会にあらわれる組合せ最適化問題の多くはNP困難

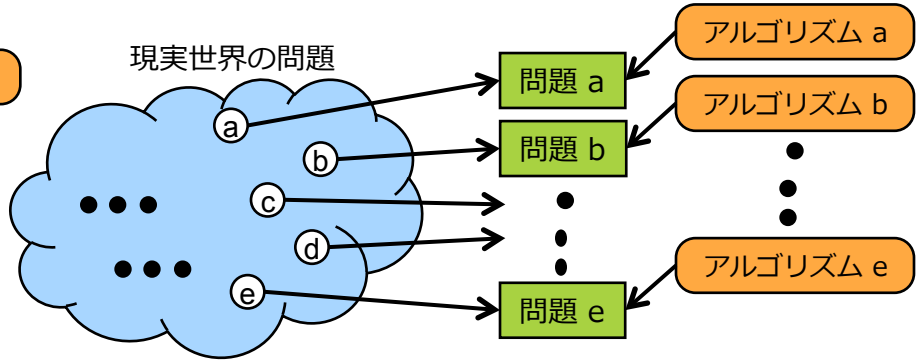
Yes/Noを出力する決定問題ではNP完全問題と呼ばれる.

計算困難な問題に対するアプローチ

- 厳密解法と近似解法
  - 厳密解法：最悪の場合に指数時間かかっても最適解を求める.
  - 近似解法：現実的な計算時間で良い実行可能解を求める.
- 汎用解法と専用解法
  - 汎用解法：整数計画問題や制約充足問題などに定式化して汎用ソルバーを適用する.
  - 専用解法：問題特有の性質を利用した専用ソルバーを適用もしくは開発する.



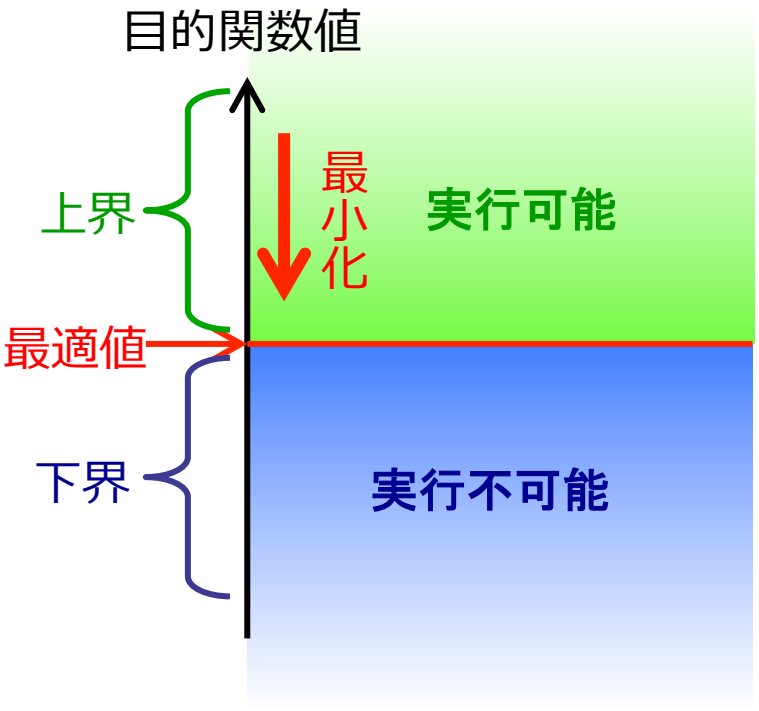
多様な問題に適用可能な
汎用性の高いアルゴリズム

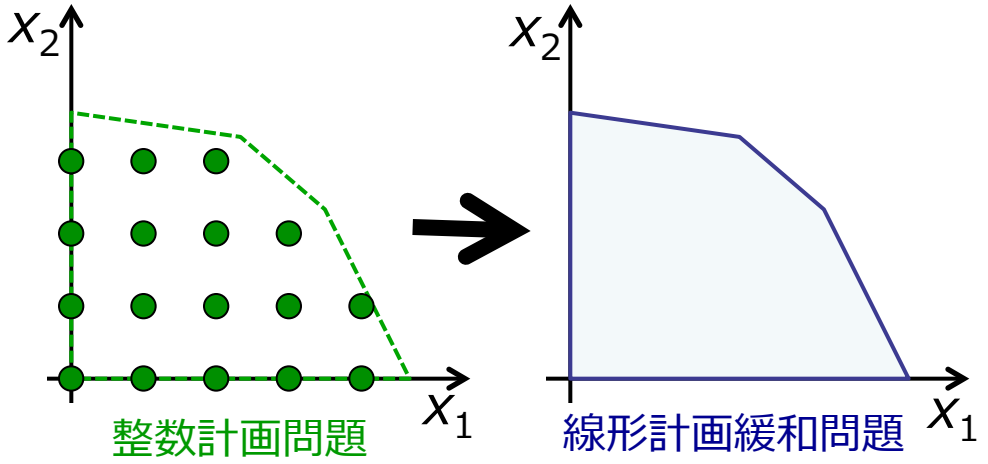


個々の問題の特徴を利用した
高性能なアルゴリズム

精度保証付き近似解法

- 同時に緩和問題を解いて下界も求めるアルゴリズムを設計できれば，任意の問題例に対して精度を保証できる場合がある．
- 緩和問題：原問題の一部の制約条件を緩めた問題
- 線形計画緩和問題：整数変数の整数条件を外す→線形計画問題
- ただし，独立に緩和問題を解いて下界を求めれば，得られた解の精度を事後的に評価することは可能．





MIPの実行可能領域⊆LPの実行可能領域

現実問題に対するアプローチ

- 既存の汎用ソルバーを利用
  - 現実問題が既知の最適化問題と完全に一致することは稀.
  - 専用ソルバーが利用可能な状態で公開されていることは稀.
  - (混合)整数計画問題に定式化して厳密解法に基づく汎用ソルバーを適用する.
  - 実用的に解ける問題の規模が限られる場合が多い.

厳密解法：整数計画ソルバーを使いこなす

- 自分で専用ソルバーを開発
  - (混合)整数計画ソルバーでは解けない大規模・複雑な問題.
  - 問題構造を上手く利用すれば効率的なソルバーが開発可能.
  - 十分な知識・技術と開発期間が必要.

近似解法：メタヒューリスティクスを設計する

問題の分割や定式化の工夫などで対応できる場合も多い

最適化モデルの正確さを犠牲にして易しい問題に定式化する方法も良く採られる.

大規模な組合せ最適化問題に対する発見的解法

なぜ専用ソルバーを開発するか？

- 整数計画問題の記述が困難もしくは多くの変数や制約が必要
  - 順列を決定するタイプの問題は変数・制約が非常に多く，汎用ソルバーの単純な適用では大規模な問題例が解けないことが多い.
  - 多角形詰込み問題のように定式化自体が困難な問題も多い.
- 汎用ソルバーで対応できないほど大規模 and/or 難しい
  - 汎用ソルバーの進歩は目覚ましいものの対応できない問題はまだ多い.
- 整数計画問題の記述に現れない有用な情報がある
  - 入力データに偏りがあることが分かっている場合，汎用ソルバーでは実行可能解すら求まらないのに，簡単なアルゴリズムで良い近似解が求まることがある.
- 高性能な汎用ソルバーは有償のため営利目的での利用が困難
  - 1製品ごとに1ライセンスを利用すると大赤字になってしまう.
  - 大規模プロジェクトで少数のライセンスを利用する場合はOK.
- 動作原理の説明が難しい
  - 最適解が期待していた解と異なる場合に理由の説明が難しい.

汎用ソルバーに含まれるライブラリを使って大規模問題に対応できるアルゴリズムを開発することも可能です.

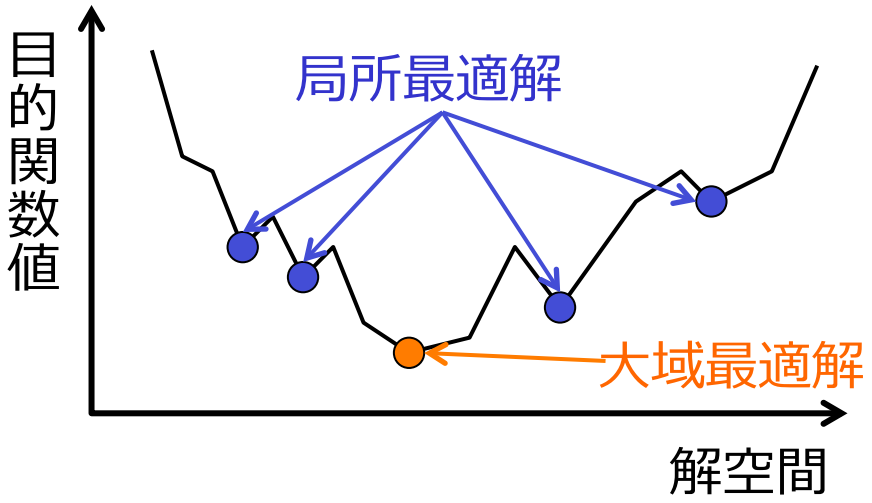
メタヒューリスティクス

- ヒューリスティクス
  - もともとは発見的法則，経験則に基づく問題解決の方法を指す．それが転じて，最適化問題における最適性の保証のないアルゴリズムの総称として呼ばれる．
  - 結果が非常に悪くなるごく少数の反例が存在する場合や，アルゴリズムが複雑過ぎて解析的な評価ができない場合が多い．
- メタヒューリスティクス
  - 最適化問題に対する実用的なアルゴリズムを設計するための一般的な枠組み(レシピ集)，もしくはそのような考え方に従って設計された様々なアルゴリズムの総称として呼ばれる．
  - アニーリング法，タブー探索法，遺伝アルゴリズム，進化計算，アント法，粒子群最適化など多くの探索戦略がある．
  - 自然現象にアナロジーを持つ手法が多いが，自然現象を模倣すればアルゴリズムの性能が良くなるわけではない．

解析的に評価できなくてもそれなりに説明できるノウハウはあります

メタヒューリスティクス

- 多くの最適化問題は近接最適性(良い解の近くに良い解がある)と多峰性(局所最適解が多数存在する)を持つ.
- 近似解法の基本戦略は貪欲法(構築法)と局所探索法(改善法).
- 貪欲法や局所探索法などの基本的な手法に様々な手法を組み合わせて，探索の集中化と多様化をバランス良く実現する.
- 計算の効率化，探索空間の削減，問題構造の利用など，アルゴリズム，最適化，離散数学の知識を活用すれば，大規模な組合せ最適化問題に対して高性能なメタヒューリスティクスを開発できる.



メタヒューリスティクスの設計

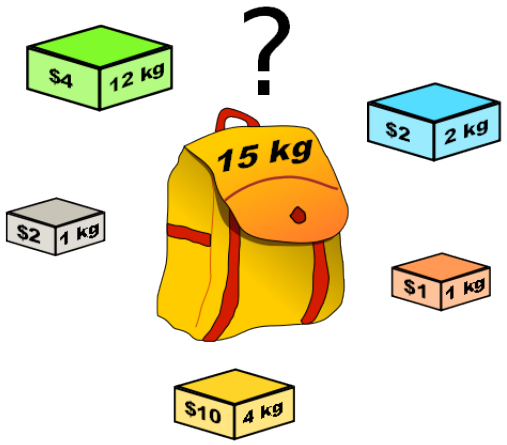
- メタヒューリスティクスが向いている場合
  - ① 厳密な最適解を求めるのが困難である
    (効率的な解法が知られている or 問題規模が十分に小さい場合はダメ)
  - ② 解の評価が高速にできる
    (評価関数がblack-boxで評価にシミュレーションや実験を要する場合はダメ)
  - ③ 近接最適性が成り立つ
    (微小な変形で評価関数の値が大きく変わる場合はダメ)
- メタヒューリスティクスの多くは局所探索法の一般化なので，まずは高性能な局所探索法を設計することが重要.
- 局所探索法では，探索空間，解の評価，近傍の定義，移動戦略などの各要素を注意深く設計することで高性能なアルゴリズムを実現できる.

申し訳ありませんが今回は多点探索に関わる部分は説明しません

多点探索を含むメタヒューリスティクス全般については，オペレーションズ・リサーチ **58**(2013)の特集「はじめようメタヒューリスティクス」をご参照下さい.

貪欲法

- 目的関数への貢献度を示す局所的な評価値に基づいて実行可能解を直接構成する方法.
- ナップサック問題の例
  - 各荷物 i の重さ w_i と価値 p_i に対して，w_i / p_i は単位重さ当たりの価値と解釈できるので，これの大きい順に詰めて行く.
  - この貪欲法を少し修正すると，常に最適値の1/2以上の近似値が得られる精度保証付き近似解法になる.



$$\max \quad \sum_{i=1}^{n} p_i x_i$$

荷物 i の価値

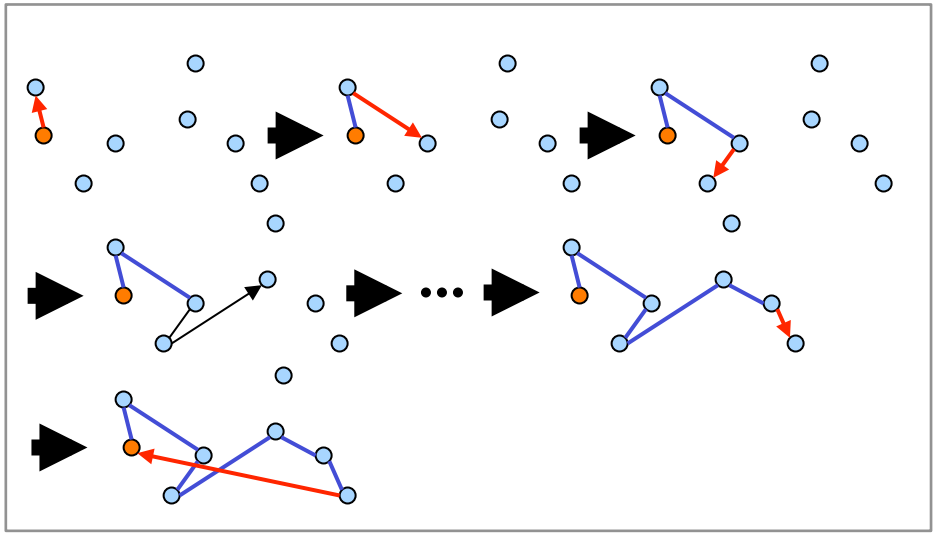
$$\text{s.t.} \quad \sum_{i=1}^{n} w_i x_i \leq c,$$

荷物 i の重さ

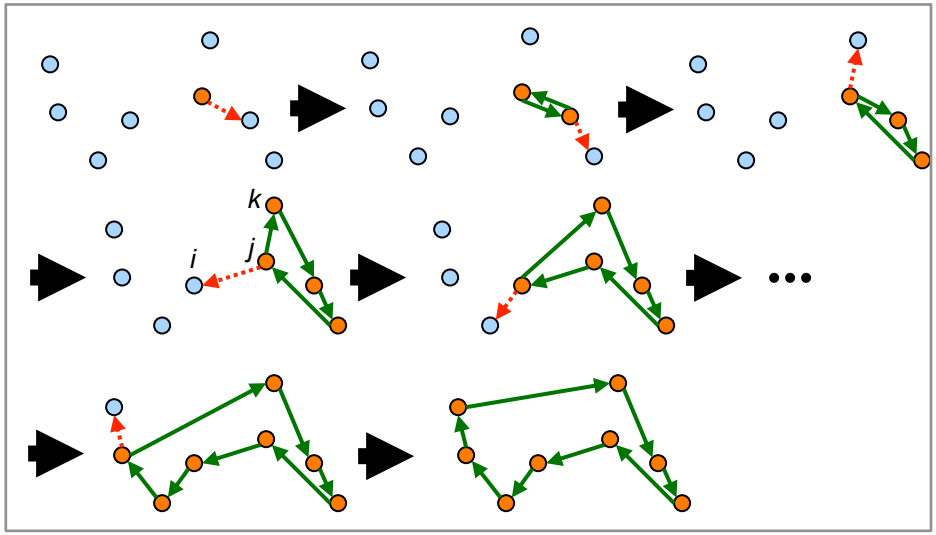
$$x_i \in \{0, 1\}, \ i = 1, \ldots, n.$$

貪欲法

- 巡回セールスマン問題に対する最近近傍法
  - 現在いる都市から最も近い未訪問の都市に移動する操作を繰り返す.
  - 全ての都市を訪問したら出発した都市に戻る.
- 巡回セールスマン問題に対する最近追加法
  - 部分巡回路Tに都市を1つずつ加える操作を繰り返す.
  - 部分巡回路Tからの距離$d(T,i)$が最小となる都市iを選ぶ.
  - 都市jと隣接する都市kの間の枝(j,k)を繋ぎ替えて新たな部分巡回路を作る.



最近近傍法の例

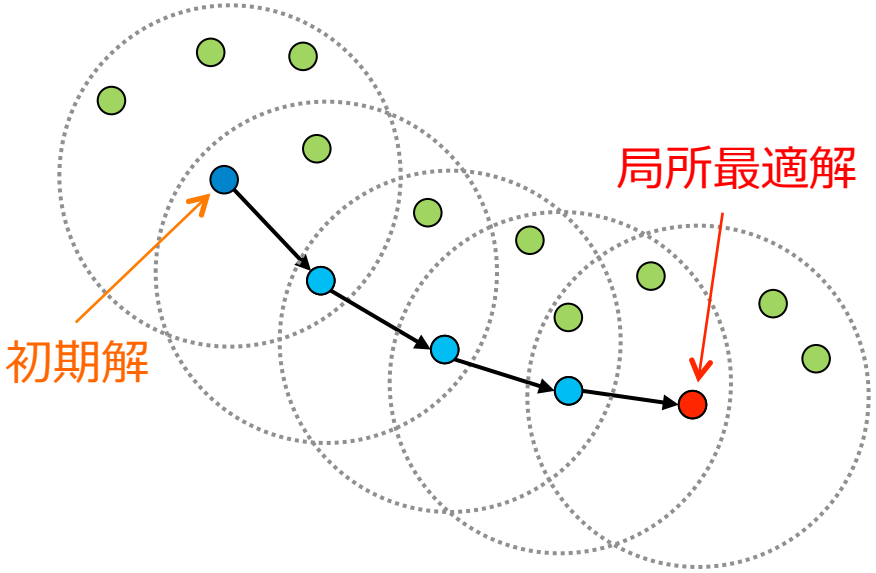


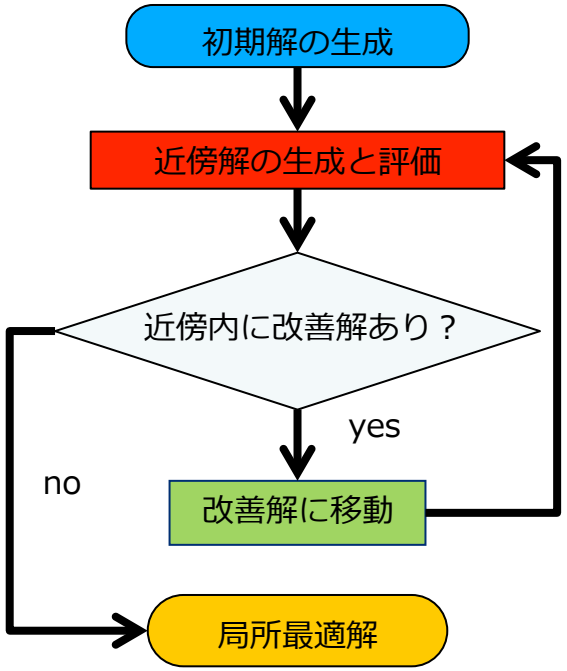
最近追加法の例

最近追加法は最適巡回路の2倍以内の長さの巡回路を常に出力する.

局所探索法

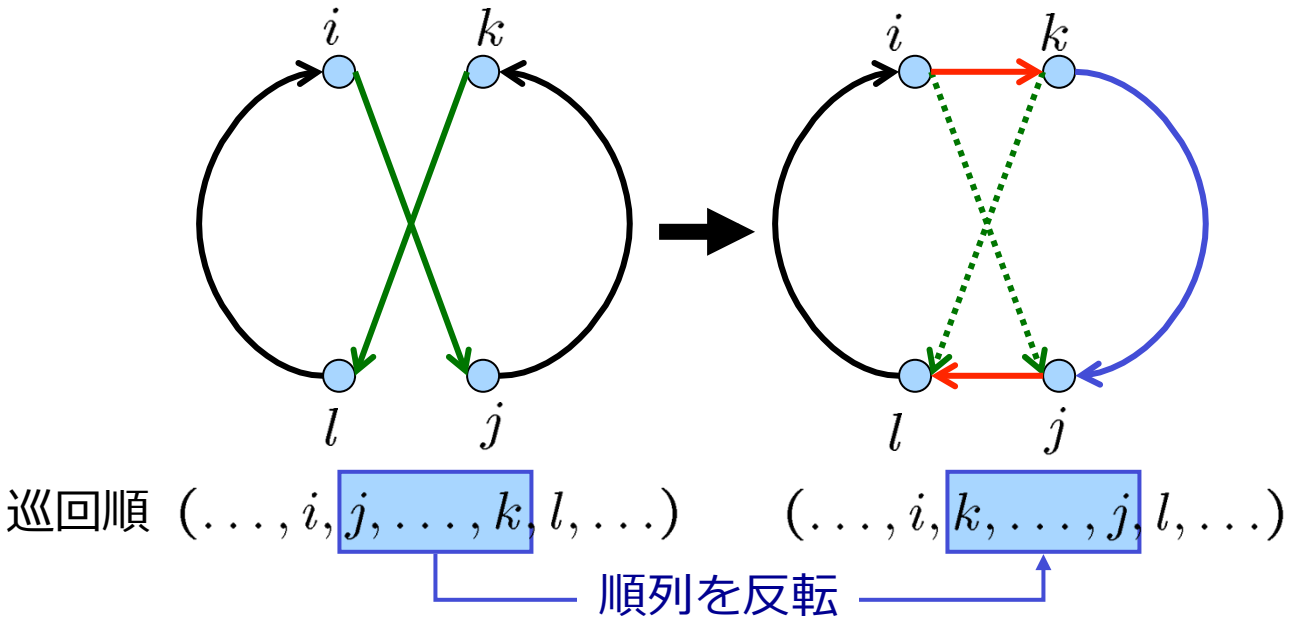
- 適当な初期解 x から始めて，現在の解 x の近傍 $x' \in NB(x)$に改善解があれば移動する(近傍探索).
- 勾配法では微分$\nabla f(x)$を用いるが，局所探索法では差分 $\Delta f(x) = f(x') - f(x)$を用いて探索方向を決定する.
- 集合分割問題の近傍の例
  - 1-flip近傍：x_j=0→1 (or 1→0)
  - 2-flip近傍：x_{j1}=1→0, x_{j2}=0→1





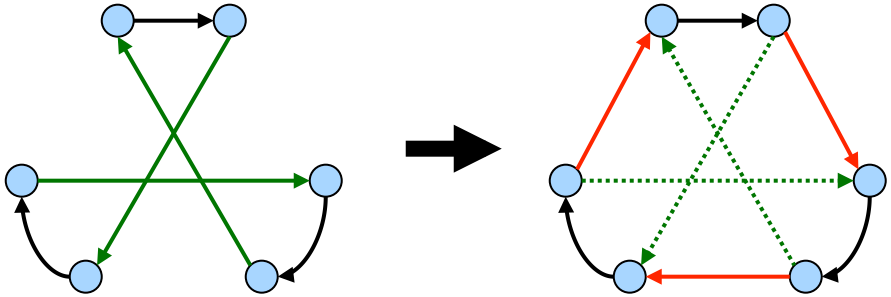
局所探索法

- 整数計画問題によらない自然な定式化で扱える.
- 微小な変形(近傍操作)による評価関数の変化量のみ計算すれば良いので，短時間に多くの解が探索可能.
- 巡回セールスマン問題に対する2-opt近傍
  - 現在の巡回路から辺を高々2本交換する
  - 巡回路を順列で記述している場合は一部を反転する操作に対応する.

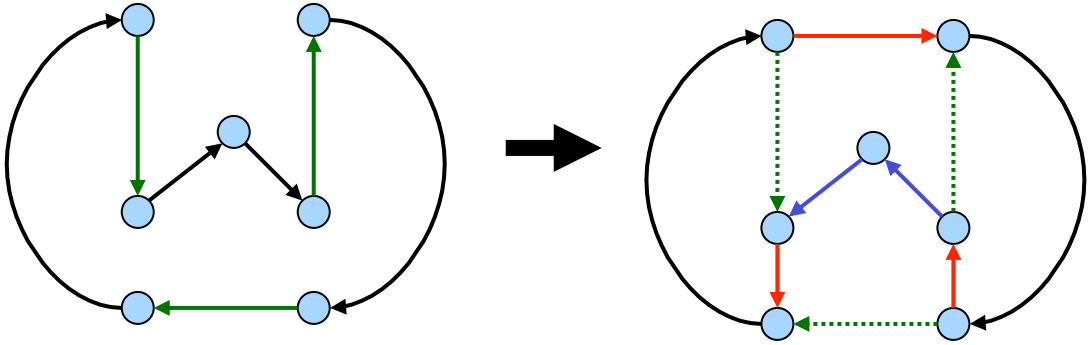


局所探索法

- 自由な発想に基づいて様々な近傍を定義できる.
- 巡回セールスマン問題に対する3-opt近傍，Or-opt近傍
  - 枝を3本付け替えて得られる巡回路，部分路を別の場所に挿入して得られる巡回路.



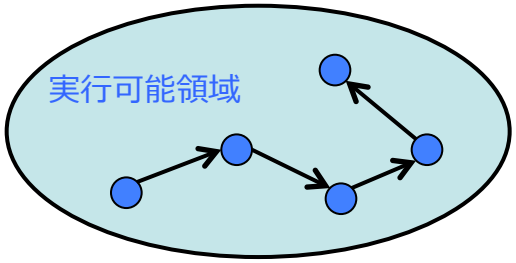
3-opt近傍の例



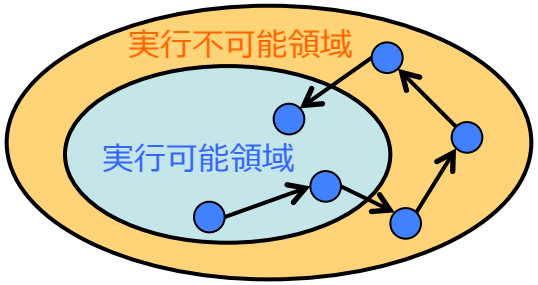
Or-opt近傍の例

探索空間

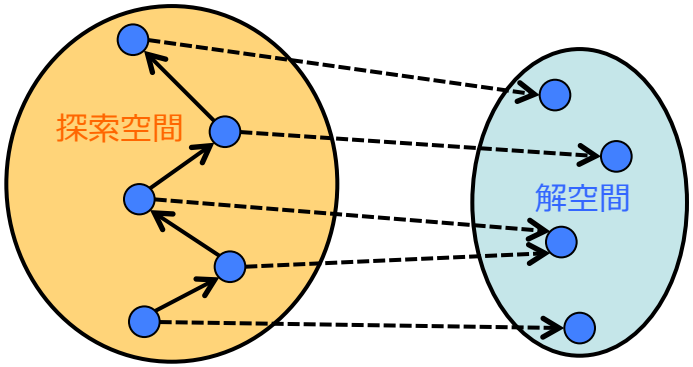
- 問題によっては実行可能領域とは異なる探索空間を定義する方が有効な場合も多い.
- 探索空間の定義は大きく以下の3通りに分類できる.
  a) 実行可能解のみ探索する.
  b) 実行不可能解も含めて探索する →実行不可能解の評価が必要
  c) 解空間とは異なる探索空間をを導入して，探索空間から解空間への写像を用いて探索する.



a)実行可能解のみ探索



b)実行不可能解も探索



c)解空間と異なる探索空間を用意

実行不可能解も探索

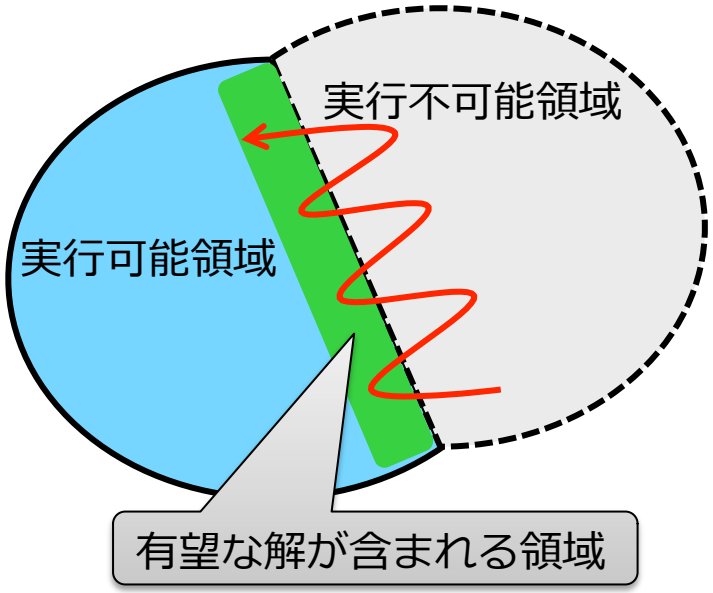
- 集合分割問題の例
  - 実行可能解を一つ見付けることも非常に難しい.
  - 制約条件を緩和して実行不可能解も探索空間に含める.
  - 各制約iに対する違反度に重みw_iを掛けた値をペナルティとして目的関数に加える.
  - ペナルティ重みw_iの設定は容易ではないので適応的に調整する.
    (実行可能領域の境界近辺を集中的に探索できると嬉しい)

$$z(\mathbf{x}) = \sum_{j \in N} c_j x_j + \sum_{i \in M} w_i \left| \sum_{j \in N} a_{ij} x_j - 1 \right|$$

ペナルティ重み

各制約式の重みパラメータを適応的に増減しつつ探索する

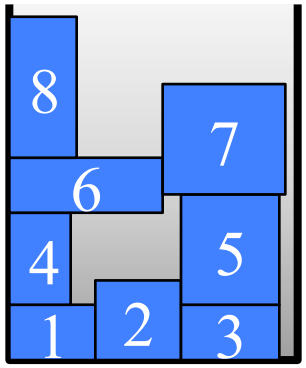
違反度



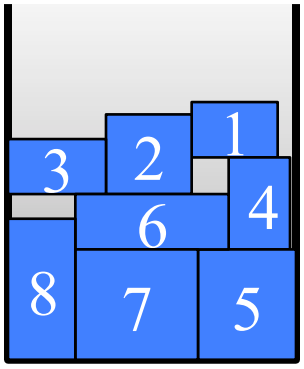
解空間と異なる探索空間

- 長方形詰込み問題の例
  - BL法(bottom-left法)：順列σの順にBL点に長方形を配置する.
  - BL点：配置可能な最も低い位置の中で最も左の位置.
  - 順列σ全ての集合を探索空間として，対応する配置の目的関数でσを評価する.

σ：1 2 3 4 5 6 7 8 の場合



σ：8 7 6 5 4 3 2 1 の場合

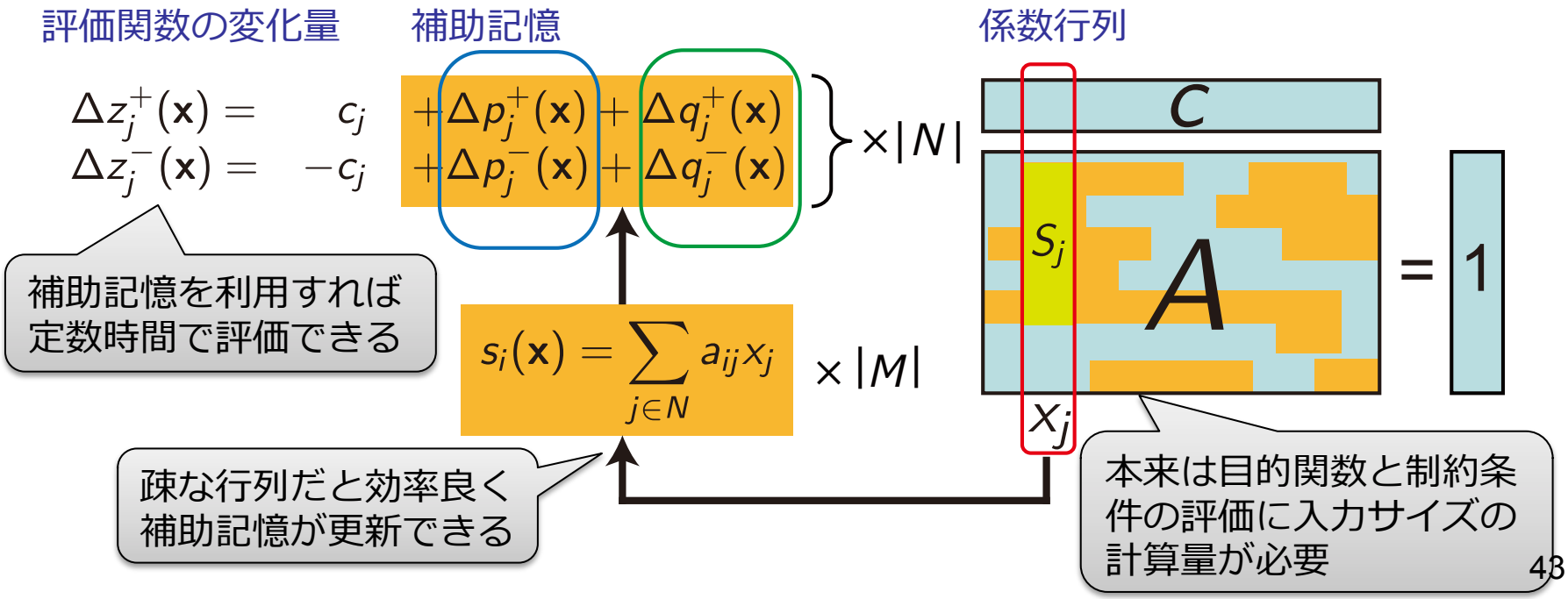


近傍探索の効率化

- 局所探索法では計算時間の大部分が近傍探索に費やされるので，近傍探索の効率化はアルゴリズム全体の高速化に直結する．
- 近傍探索の高速化により，同程度の計算時間でより大きな近傍を探索できるようになり，解の精度が向上する効果も期待できる．
- 近傍探索を効率化する方法は大きく以下の2通りに分類できる．
  a) 解の評価値の計算を高速化する．
  b) 改善の可能性のない解の探索を省略する．

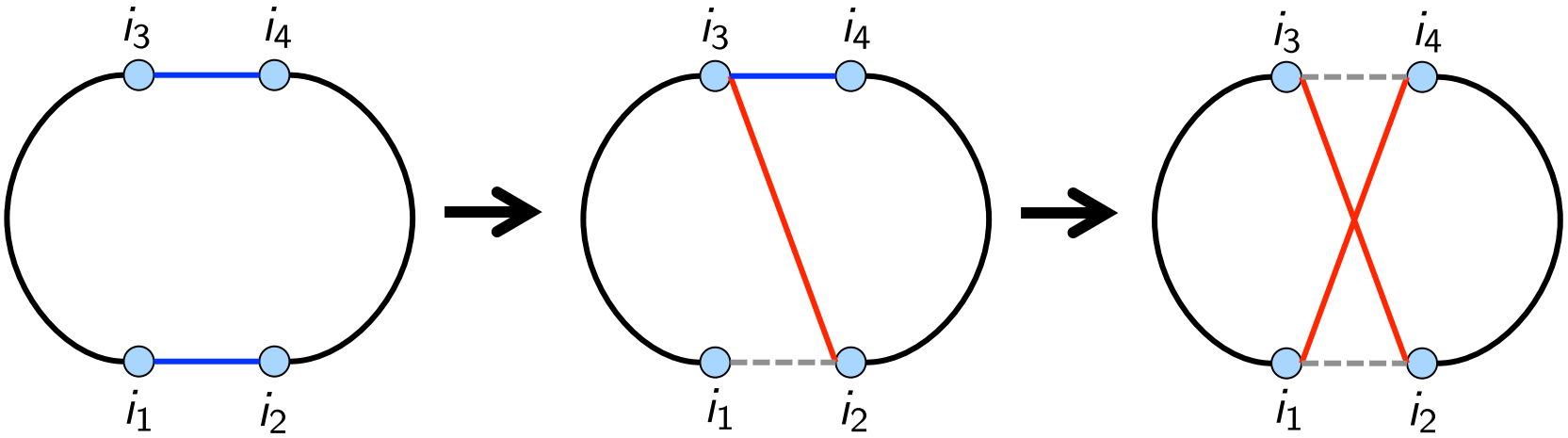
解の評価値の計算の高速化

- 近傍操作ではごく少数の変数のみ値が変化するため，現在の解$\boldsymbol{x}$と近傍解 $\boldsymbol{x}' \in \mathrm{NB}(x)$ の間で値が変化した変数に関わる部分のみ再計算すれば，評価関数値の変化量 $\Delta z(\boldsymbol{x}) = z(\boldsymbol{x}') - z(\boldsymbol{x})$ を高速に計算できる場合が多い．
- 局所探索法では，近傍内の解を評価する回数に比べて，現在の解が移動する回数ははるかに少ないので，補助記憶の更新に多少時間がかかっても，全体では十分な高速化が実現できる．



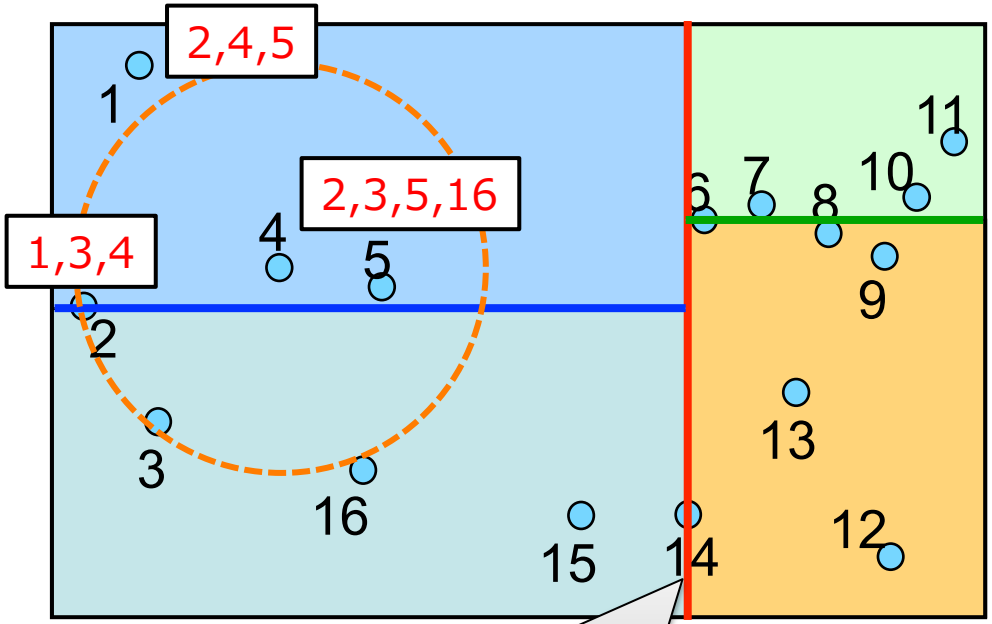
近傍の縮小

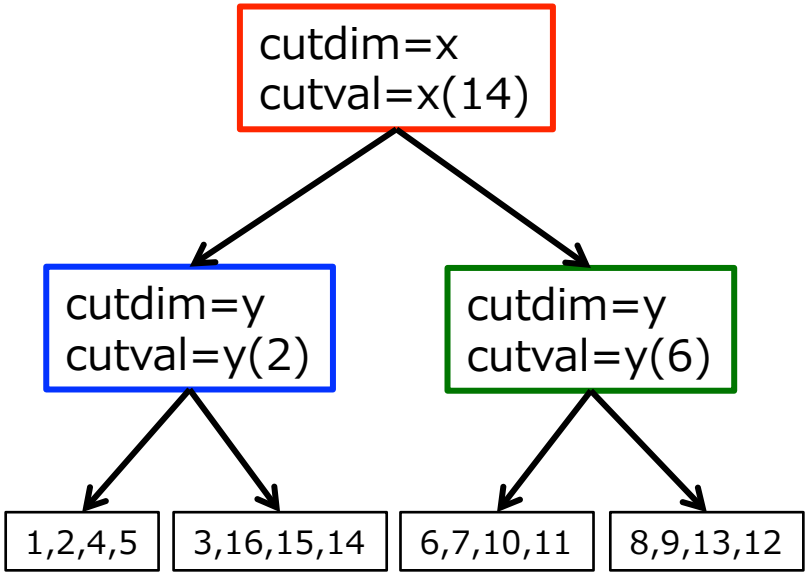
- 巡回セールスマン問題の例：
  - 現在の巡回路から辺$\{i_1,i_2\}$と$\{i_3,i_4\}$を除き，辺$\{i_2,i_3\}$と$\{i_4,i_1\}$を加える2-opt近傍の操作を考える．
  - (1)辺$\{i_1,i_2\}$を削除して$\{i_2,i_3\}$を追加し，(2)辺$\{i_3,i_4\}$を削除して$\{i_1,i_4\}$を追加すると2段階に分ける．
  - $d_{i_2,i_3}+d_{i_4,i_1}<d_{i_1,i_2}+d_{i_3,i_4}$が成り立つためには$d_{i_2,i_3}<d_{i_1,i_2}$，$d_{i_4,i_1}<d_{i_3,i_4}$の少なくとも一方が成立しているはず．
  - 削除する辺$\{i_1,i_2\}$を決めたら追加する辺$\{i_2,i_3\}$は$d_{i_2,i_3}<d_{i_1,i_2}$を満たす辺に限定する．



近傍の縮小

- 最適巡回路は近隣の都市間を繋ぐ枝を含む場合が多い.
- 各都市に対して近隣にある都市を列挙して隣接リストに保持する.
- 平面上の巡回セールスマン問題であれば，k-d木のデータ構造を用いて近隣にある都市を高速に検索できる.
- ただし，高次元空間における類似ベクトルの検索には向かない.

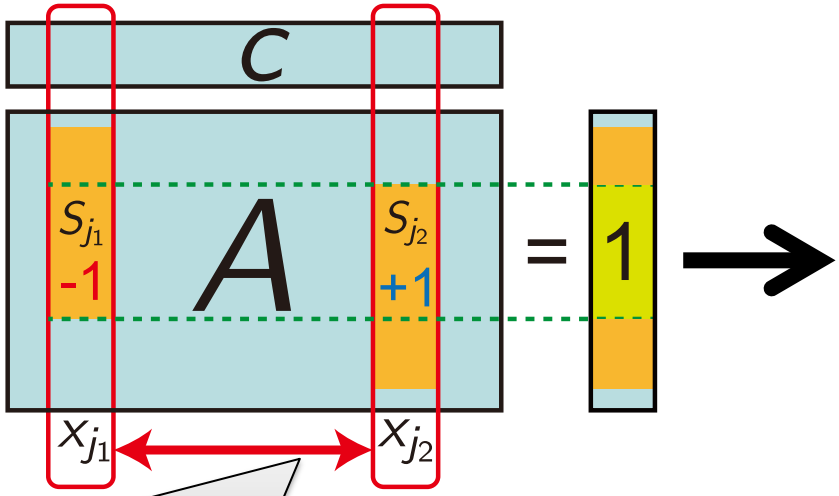




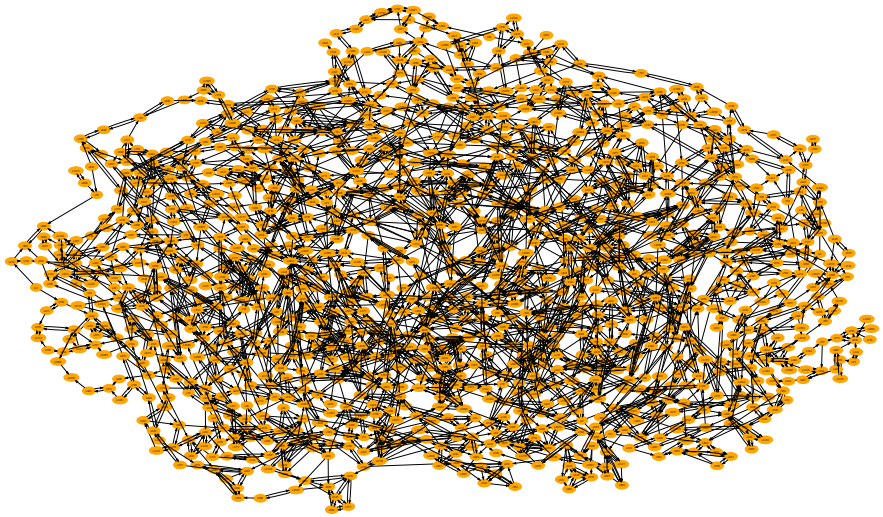
生成されたk-d木

変数間の関係による問題の縮小

- 集合分割問題における2-flip近傍の例：
  - 現在の解$\boldsymbol{x}$が1-flip近傍の局所最適解ならば，x_{j1}=1→0，x_{j2}=0→１と反転させる交換近傍のみ考慮すれば良い．
  - 同時に反転すると改善解が得られる可能性の高い変数の組み合わせを効率良く見付けたい．
  - 入力データから特殊な制約式に同時に現れる変数の組み合わせを解析してネットワークを生成する．



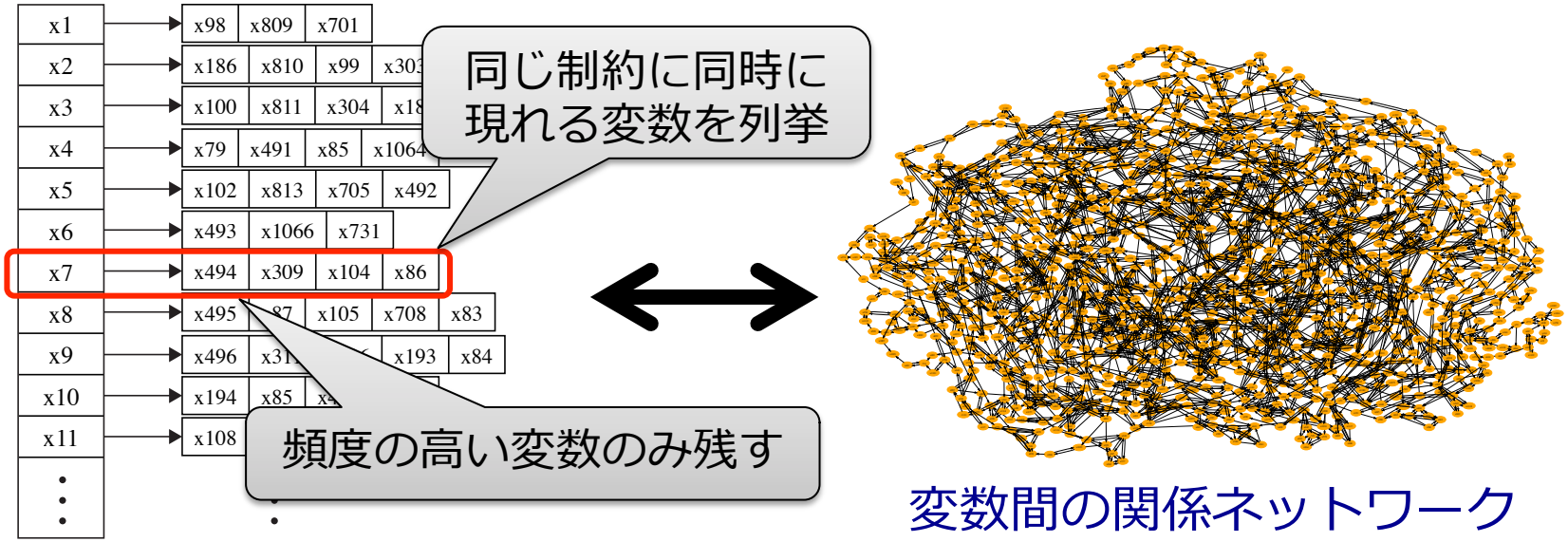
x_{j1}=1→0，x_{j2}=0→１と同時に反転するとペナルティが打ち消される



変数間の関係ネットワーク

変数間の関係による問題の縮小

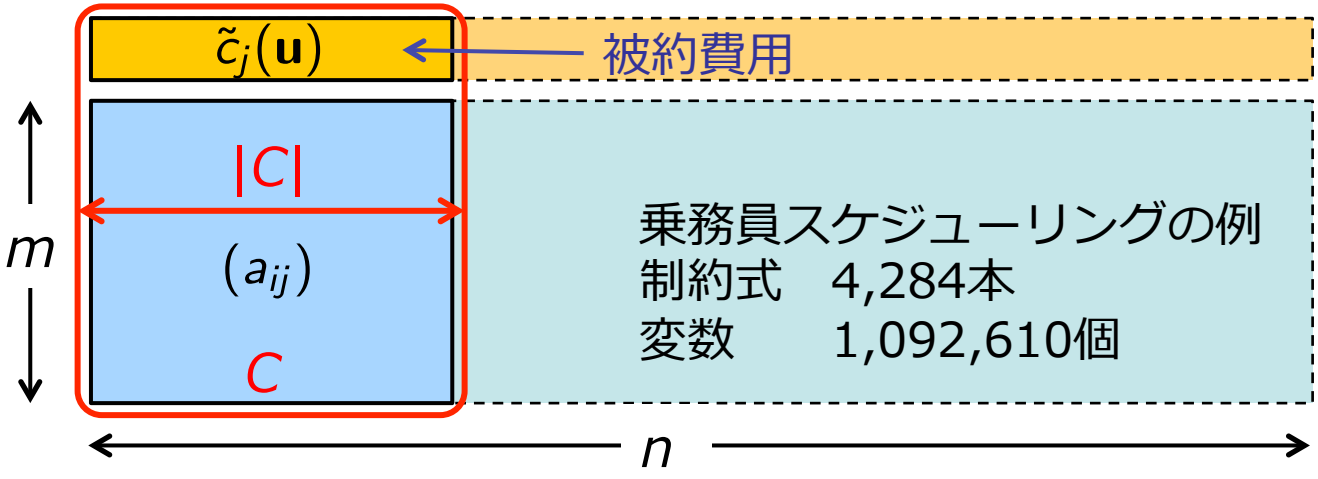
- 各変数x_jについて同じ割当制約に同時に現れる変数を列挙.
- 変数が非常に多い問題例は少なくない.
- 組合せの数がn^2(nは変数の数)となるので，x_jと同時に現れる頻度の高い変数のみ(上位数%)残してネットワークを構成.
- 多くの計算時間を要するので，変数x_jを反転する際にネットワークの必要な部分のみ遅延生成する.



遅延生成すれば高次元空間でも隣接リストを利用できる

線形計画法による問題の縮小

- 集合被覆問題の例：
  - 変数の数nが非常に多い事例がしばしば現れる.
  - x_j=1となる変数は高々制約式の本数となる.
  - 有望な変数のみを用いた制限された集合被覆問題を解く.
  - コスト係数c_jだけでは有望な変数が特定できない.
  - 線形計画緩和問題(各変数の整数制約を緩和した問題)の双対解 u_i を利用して各変数を評価 → 価格法，列生成法



機械学習でもオンライン最適化やLPブースティングなど類似した手法が多く知られる.

線形計画法による問題の縮小

- LP緩和問題ではその最適値と双対問題の最適値が一致する.
- 主問題に変数を追加＝双対問題に制約式を追加.
- 部分問題 $C \subset N$ のLP緩和問題を解いた後に，残りの変数の被約費用 $\tilde{c}_j(\mathbf{u})$ $(j \in N \setminus C)$ を計算し，$\tilde{c}_j(\mathbf{u}) < 0$ となる変数を部分問題Cに追加.

主LP緩和問題：

$$\min \sum_{j \in C} c_j x_j$$

$$\text{s.t.} \sum_{j \in C} a_{ij} x_j \geq 1, \ i \in M,$$

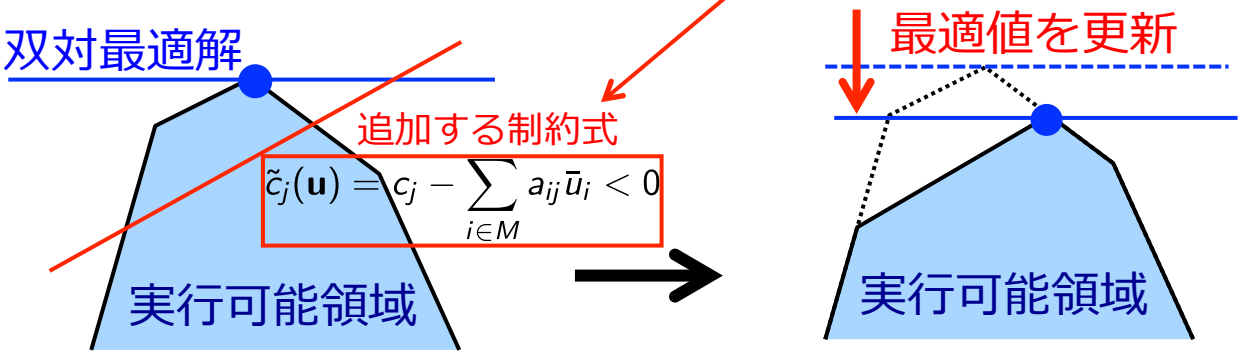
$$x_j \geq 0, \ j \in N.$$

双対LP緩和問題：

$$\max \sum_{i \in M} u_i$$

$$\text{s.t.} \sum_{i \in M} a_{ij} u_i \leq c_j, \ j \in C,$$

$$u_i \geq 0, \ i \in M.$$



まとめ

- 組合せ最適化問題とその応用
  - 巡回セールスマン問題，集合分割問題，長方形詰込み問題
  - 配送計画，選挙区割り，機械翻訳におけるフレーズ対応，推薦商品の最適化，人の移動履歴の推定，基板検査プローブの経路計画，電気自動車の充放電計画
- 計算困難な組合せ最適化問題に対するアプローチ
  - 易しい問題と難しい問題
  - 厳密解法と近似解法，汎用解法と専用解法
- 大規模な組合せ最適化問題に対する発見的解法
  - メタヒューリスティクス，貪欲法，局所探索法
  - 探索空間，近傍探索の効率化
  - 解の評価値の計算の高速化，近傍の縮小，変数間の関係による問題の縮小，線形計画法による問題の縮小

アルゴリズム設計の技術を上手く取り入れることで
大規模な組合せ最適化問題に対する効率的なアルゴリズムは実現できる

さらには，データ，インターフェース，モデル，アルゴリズムなど全体を俯瞰した上で最良の解決策を目指すことが大事だとは思います．

参考文献

- 組合せ最適化問題とその応用
  - H.P.Williams, Model Building in Mathematical Programming (4th edition),Wiley，1999. (前田英次郎 監訳，小林英三 訳，数理計画モデルの作成，産業図書，1995)
- 計算困難な組合せ最適化問題に対するアプローチ
  - 宮代隆平，整数計画ソルバー入門，オペレーションズ・リサーチ，**57**(2012)，183-189.
  - 藤江哲也，整数計画法による定式化入門，オペレーションズ・リサーチ，**57**(2012)，190-197.
  - 久保幹雄，J.P.ペドロソ，村松正和，A.レイス，あたらしい数理最適化―Python言語とGurobiで解く―，近代科学社，2012.
  - T.Berthold, A.M.Gleixner, S.Heinz, T.Koch, 品野勇治, SCIP Optimization Suite を利用した 混合整数 (線形/非線形) 計画問題の解法, ZIB-Report 12-24, 2012.
- 大規模な組合せ最適化問題に対する発見的解法
  - 柳浦睦憲，茨木俊秀，組合せ最適化 ― メタ戦略を中心として ―，朝倉書店，2001.
  - 久保幹雄，J.P.ペドロソ，メタヒューリスティクスの数理，共立出版，2009.
  - 梅谷俊治，柳浦睦憲，メタヒューリスティクス事始め：まずは局所探索法から，オペレーションズ・リサーチ，**58**(2013)，689-694.
  - 今堀慎治，柳浦睦憲，概説メタヒューリスティクス，オペレーションズ・リサーチ，**58**(2013)，695-702.
  - 梅谷俊治，問題構造の解析に基づく組合せ最適化アルゴリズムの自動構成，オペレーションズ・リサーチ，**59**(2014)，20-25.
  - S.Umetani, M.Yagiura, Relaxation heuristics for the set covering problem, *Journal of Operations Research Society of Japan,* **50**(2007), 350-375.