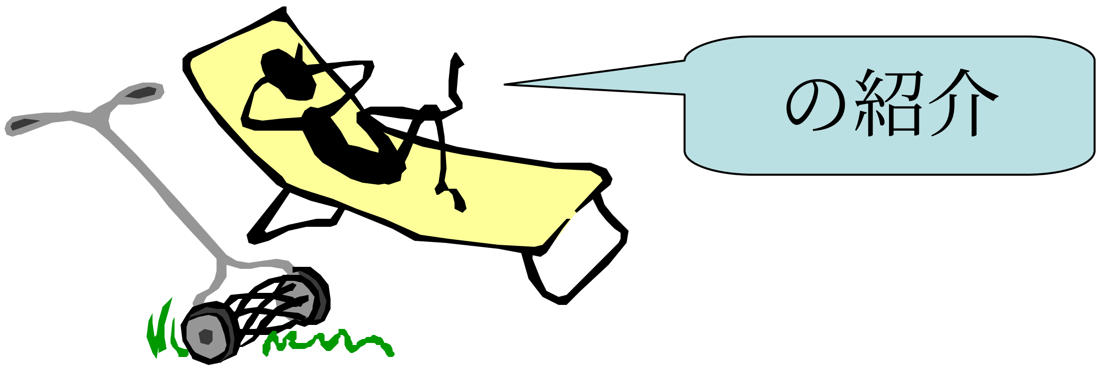
An Optimization Based Heuristic for Political Districting

A.Mehrotra,E.L.Johnson and G.L.Nemhauser

Management Science 44(1998)1100-1114



1.Introduction

- アメリカの選挙事情
- 公正な区割とは?
  - 1人1票の原則⇒1票の価値の平等
  - 隣接性
  - コンパクト性
- 過去の研究
  - ローカルサーチ :Kaiser66,Nagel65
  - 陰列挙法 :George９３(ニュージーランド)
- 貢献 :区割方法の提案, 数理モデル, 解導出

裁判所は最重要視

2. A Mathematical Framework

- 区割：郡の集合を制約を守り議席数に分割
  - 制約1：飛び地がない（隣接性）
  - 制約2：地理的にコンパクト（コンパクト性）
  - 制約3：人口格差が指定幅以内（均等性）

実際は均等が望ましい

- モデル化：グラフ（頂点）分割問題＋コンパクト
  - コンパクトの重みが必要
  - すべてのピースが数え上げられる場合
    →最小重み(総和)集合分割問題

制約1と制約3を守った郡の集合を「ピース」とよぶ

区割画定問題

- 郡の集合{1,...,n}, ピース集合J
- $\delta_{ij}=1$:郡iがピースjに含まれる⇔ $\delta_{ij}=0$
- $x_j=1$:ピースjが選挙区に採用された⇔ $x_j=0$

cjはピースjのコスト

PLAN(J)

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

$$s.t.\ \sum_{j\in J} d_{ij} x_j = 1 \ \text{for}\ i = 1,\dots,n$$

$$\sum_{j\in J} x_j = K$$

$$x_j \in \{0,1\},\ \forall j \in J$$

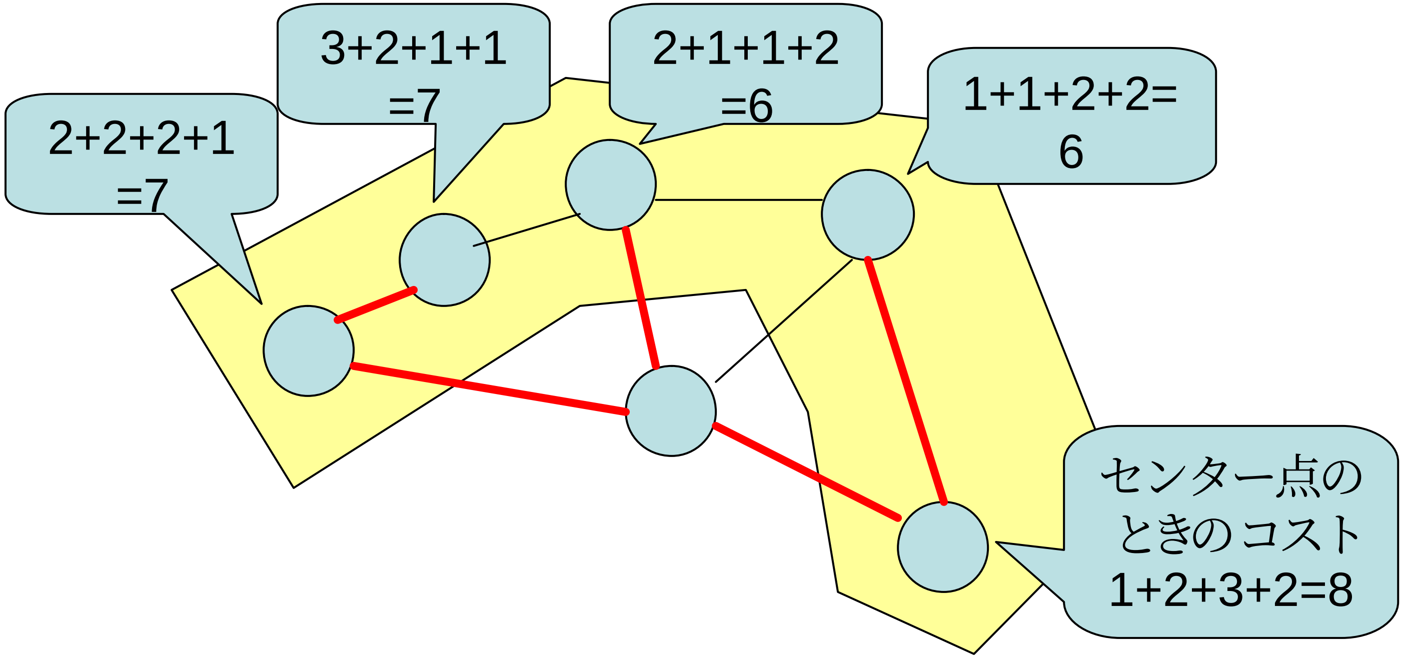
2.1. District Cost

- コンパクト性の定義→一様なものなし
- Young(1998) :8つの計測法の提案
- 『非』コンパクト性のコスト計測法を提案
  - 案1 :ピース内での2点間距離総和 ×
  - 案1修正 :2点人口和で重み付け ×
  - 案2 :2点間移動で通過する郡数 (↔枝数) ◎

案2の詳細

- 隣接グラフ G(V,E)
- ピースG’(V’,E’) :V’で誘導された部分グラフ
- G’の非コンパクト性を表すコスト
  - s_{ij}=点iから点jへの<u>G上での</u>最短パス長（枝数）
  - G’のセンター点u :$\sum\{s_{uj} \mid j\in V'\}$が最小になる点
  - G’のコスト=センター点からの最短パス長の総和
  - Youngの良い計測法の基準にも適合する

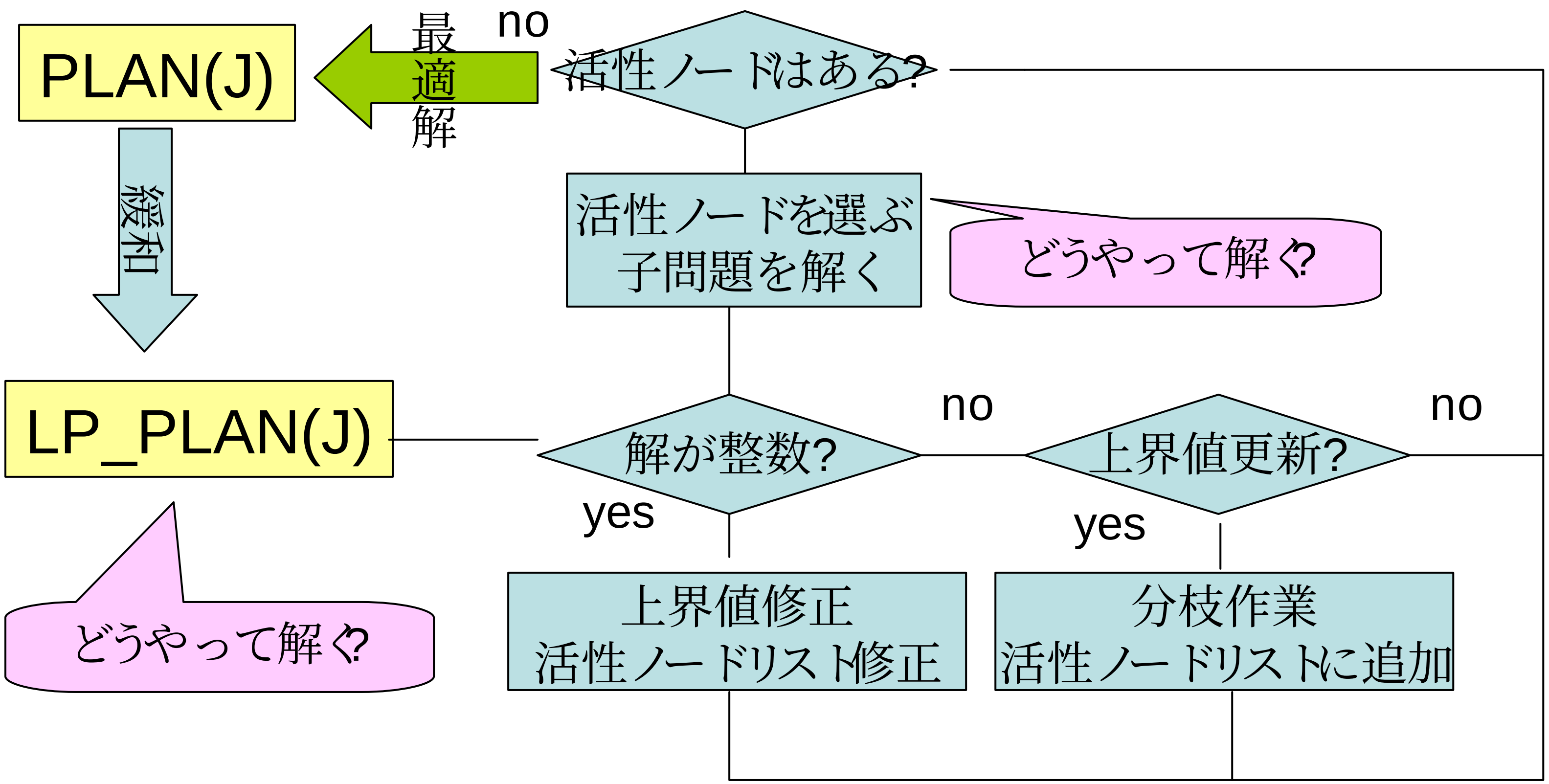
例 非コンパクト性のコスト



→ピースのコストは6

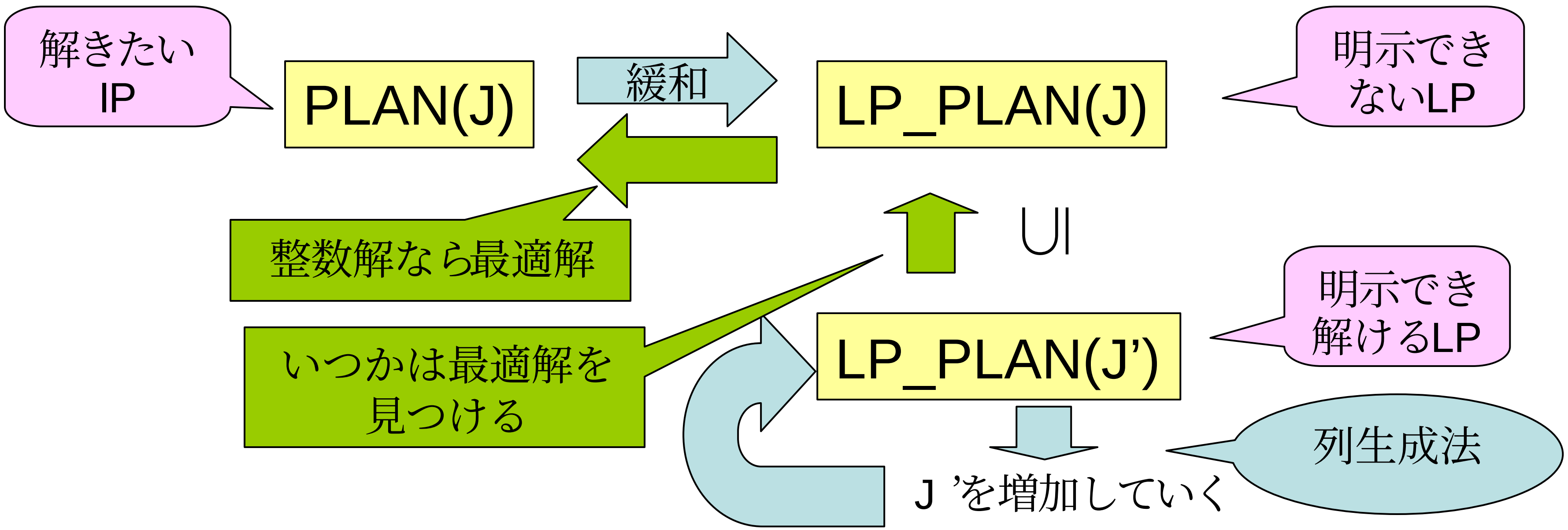
次に進む前に :今後の展開

- PLAN(J)を分枝限定法で解く

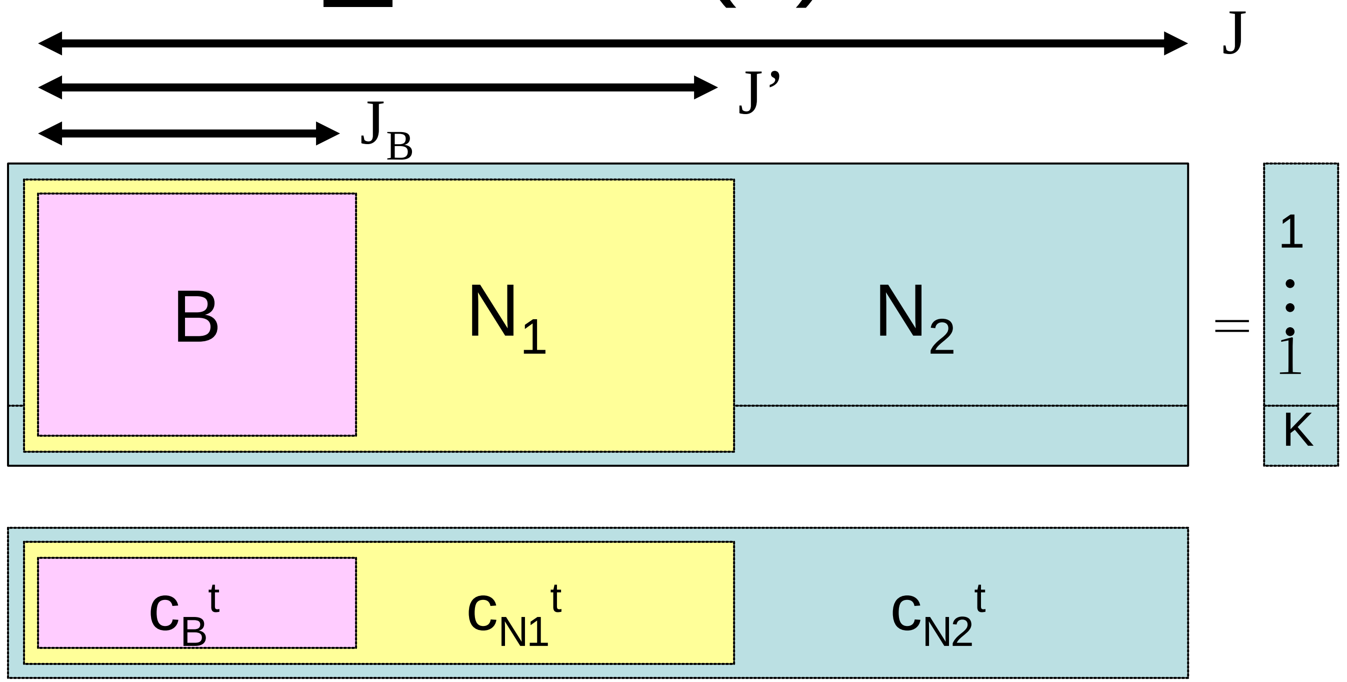


2.2 District generation

- ピースの数は指数オーダー
  →PLAN(J)は指数個の変数
  →列生成法を利用
- 現在得られているピースの集合 J'



LP_PLAN(J)の解き方



BがLP_PLAN(J’)の最適基底

⇕

$c_{N1}{}^t - c_B{}^t B^{-1} N_1 \geqq 0^t$

さらに, $c_{N2}{}^t - c_B{}^t B^{-1} N_2 \geqq 0^t$ も成立した
⇒BはLP_PLAN(J)の最適基底

$c(\mathbf{y}) - c_B{}^t B^{-1} \begin{pmatrix} \mathbf{y} \\ 1 \end{pmatrix} < 0$ なるyに対応するピースの存在判定

yをJ’に追加

最適性の判定

yがピースになる条件(＊)

- 選挙区人口の下限≦ $\sum_{i \in V} p_i y_i$ ≦上限
- V'={i | yi=1, i∈V}に誘導される部分グラフが連結
- y∈$\{0,1\}^n$

$$c(\mathbf{y}) - c_B{}^t B^{-1} \begin{pmatrix} \mathbf{y} \\ 1 \end{pmatrix} < 0$$

なるピースに対応する
yが存在するか?

⇔

以下のIPの最適値が負

$$\min \; c(\mathbf{y}) - c_B{}^t B^{-1} \begin{pmatrix} \mathbf{y} \\ 1 \end{pmatrix}$$

s.t. y**は(＊)を満たす.**

IPの解き方

$$\min_{y\in(*)}\left\{c(\mathbf{y})-c_B{}^t B^{-1}\begin{pmatrix}\mathbf{y}\\1\end{pmatrix}\right\}$$

$$=\min_{y\in(*)}\left\{\min_{u\in V}\left\{\sum_{i\in V\backslash u,\, y_u=1} s_{ui}y_i + s_{uu}y_u\right\}-\left\{\sum_{i\in V} p_i y_i + p_{n+1}\right\}\right\}$$

$$=\min_{y\in(*)}\left\{\min_{u\in V}\left\{\sum_{i\in V\backslash u} s_{ui}y_i\right\}-\left\{\sum_{i\in V} p_i y_i + p_{n+1}\right\}\right\}$$

所与

$$=\min_{y\in(*)}\left\{\min_{u\in V}\left\{\underbrace{\sum_{i\in V\backslash u}(s_{ui}-p_i)y_i - p_u - p_{n+1}}_{S(u)}\right\}\right\}$$

$$=\min_{y\in(*),u\in V} S(u)$$

$$=\min\left\{\min_{y\in(*)}S(1),\min_{y\in(*)}S(2),\cdots,\min_{y\in(*)}S(n)\right\}$$

上下限付きナップサック問題
＋
連結性

制約式を導入し解決したい

連結性の制約への導入

指数本必要
⇒×

- アイディア1 :線形不等式の導入
- アイディア2 :点uを根とする最短路木の部分木のみに限定する.

厳密性がない
↑
最短路木上にないピースはコンパクトでないことが多い

具体的には

$$S_{uj} = \{ i \in V \mid s_{ui} = s_{uj} - 1, (i, j) \in E \}$$

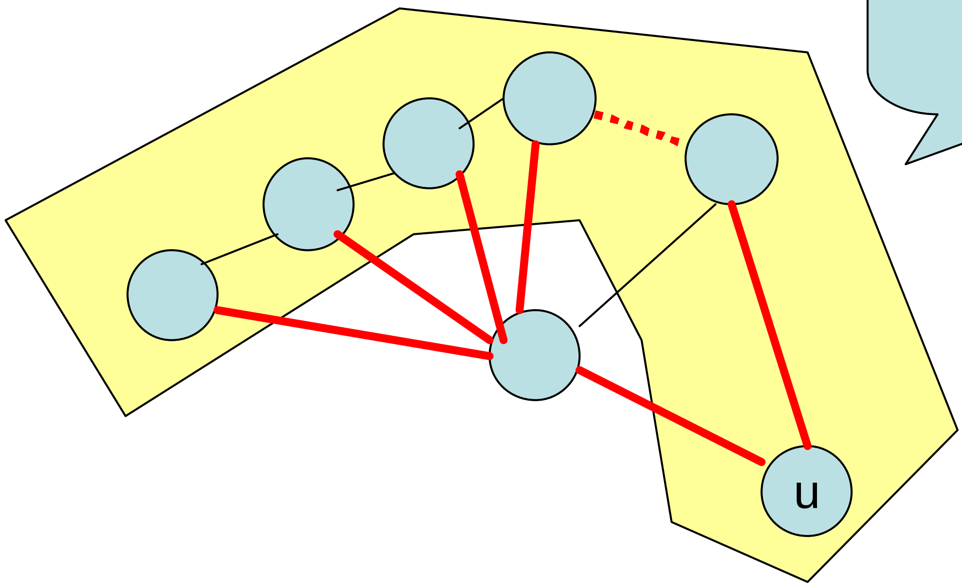
を設定し,

$$y_j \le \sum_{i \in S_{uj}} y_i$$

を制約 (*)に追加

理論はおしまい

例 :最短路木上に限定



このようなピース
は扱わなくなる.

3. Implementation Issues

実際に解く手順

1. データの準備
2. LP_PLAN(J)の初期解J’の導出
3. 最適化作業
4. 最適解から提案する区割案の作成

3.1 Basic Population Units

- すべての郡を1単位で扱うのは不適切
  - μ =（理想人口）以上の郡は分割不可避
    ⇒μ分減じる＋議席も減らす
  - 0.25μ~μの郡⇒0.25μ以下になるように分割
  - 地理的に帰属関係がある場合⇒合併
  - 0.02μ未満の郡は, 隣接最小人口郡と合併
- 00.2μ~0.25μの郡に便宜上再編

3.2 District Population Range

許容人口格差を狭くすると，コンパクトな区割案が出てこない可能性が高い

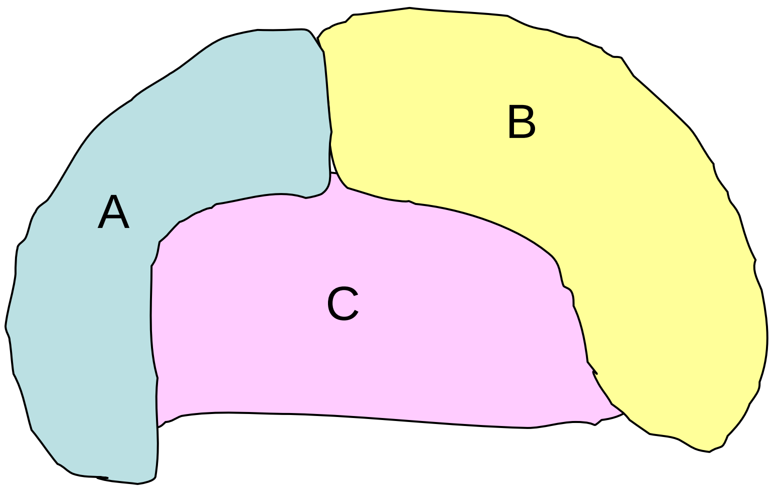
- 解決アイディア1 :郡を細かくする ×
- 解決アイディア2 :許容を広くする ◎

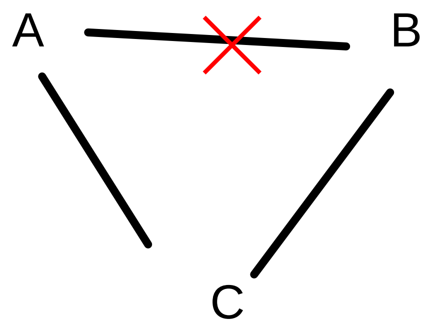
•点数が増えない
•後処理で，格差は修正できる

μの2%以内, 5%以内の2パターンで実験

3.3 Preprocessing

- 地理的隣接のままだと奇妙な形を許す
- 提案 :地理的に隣接する2郡の凸包内に, ほかの郡が含まれたら, 隣接とはしない.





3.4 Initial Plan: A Clustering Heuristics

- よい初期解⇒良いパフォーマンス
  - 区数に変化なし:現区割を採用
  - 区数に変化あり:ヒューリスティクスで見つけよう
- 素朴な方法
  - 適当な点を定める⇒広げていく

最後のほうでは
コンパクトでない
or連結でない

- 改良案
  - 参照点を1点定める(固定)
  - 参照案から遠い点を選び，拡大していく

最後に残るのは参照
点付近.

- 実行可能解が得られない時
  - 条件を満たさないピースに大きなコストを課し，列生成を始めてしまえ!

3.5 Column Generation

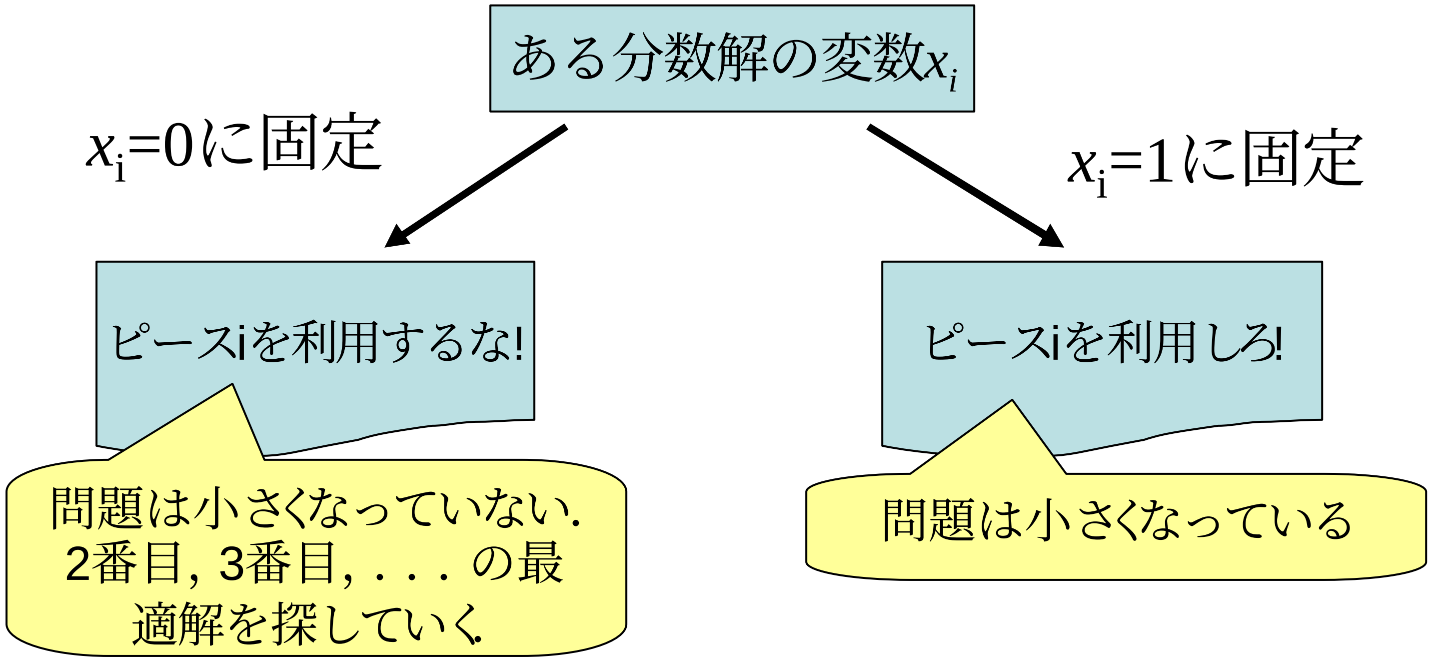
$\min_{y\in(*),u\in V} S(u)$ を解くのは，実際大変⇒工夫が必要!

- アイディア1 :$s_{ui}>3$なる点iを含むようなピースjは禁止（y_j=0と固定）
- アイディア2 :列生成をしてもコスト減少が-0.01程度になら停止
- アイディア3 :$\min_{y\in(*),u\in V} S(u)$ の最適解ではなく，負の値のピースが見つかれば利用してしまう.
- アイディア4 :S(u)をπ_uの大きな順に解く

3.6. Branching

標準的な分枝限定法は陰列挙型IPには不向き!

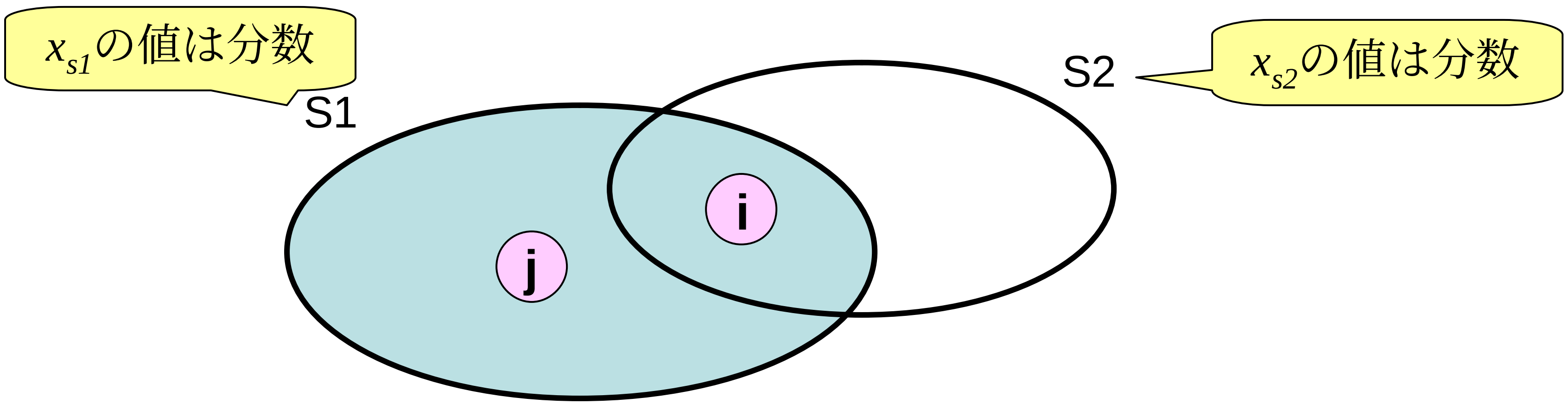
不向きな例　標準的な分枝規則



⇒解決アイディア :Ryan-Foster分枝規則 (R-F.81,Barnhart98,..)

Ryan-Foster分枝規則

- 2つの制約
  - SAME(i,j):郡iと郡jは同じ区に属す制約
  - DIFFER(i,j):郡iと郡jは同じ区に属さない制約
- 命題：LP_PLAN(J')が分数解を持つ．次のような分数解の区S1,区S2,郡i,郡jが存在する.



Ryan-Foster分枝規則（2）

ある分数解の変数x_{s1}

↓

別な分数変数x_{S2},郡i,郡jの発見

SAME(i,j)

DIFFER(i,j)

i とjは一緒にしろ!

全実行可能解は
どちらかに属す

背反

i とjは離せ!

現在の分数解は
2つの子問題両方で実行
可能になることはない

⇒標準的な方法で得られる効果は含んでいる

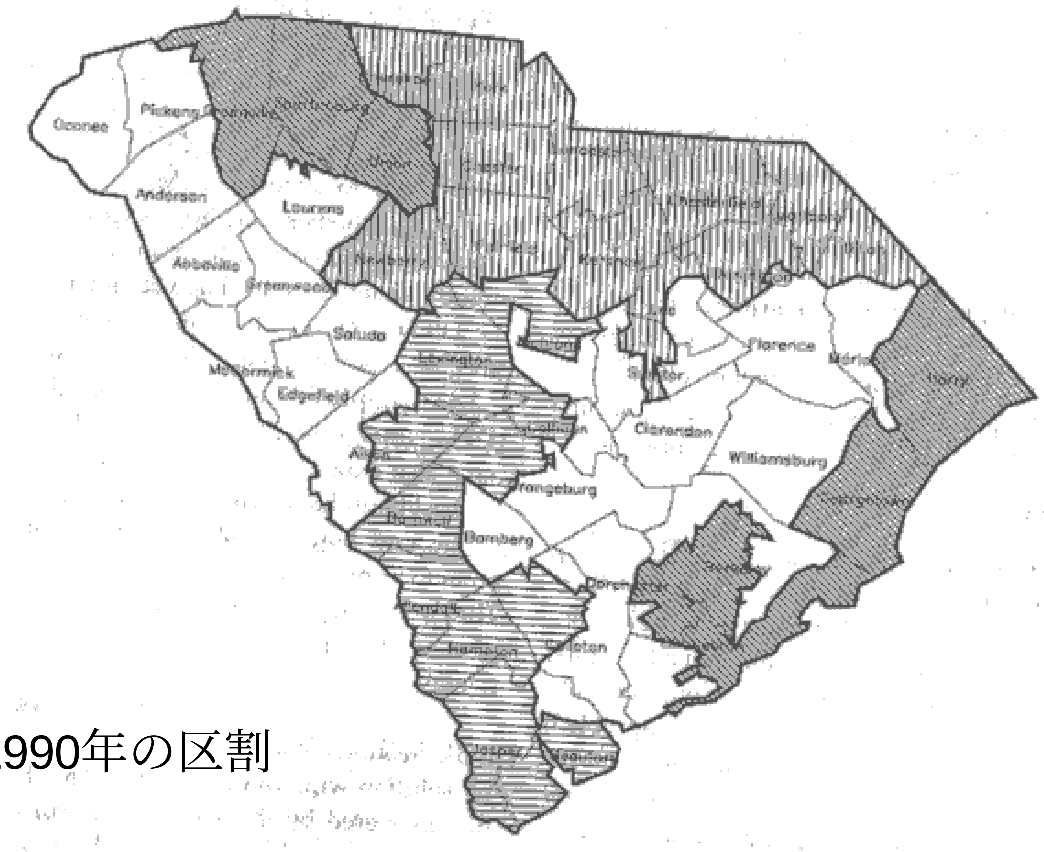
3.6.2 Implementation

- 活性ノード選択方法 :DFS⇔最良解優先 ◎
- 分数解の変数選択 案1 ◎
  - 0.5に近い値を持つピースをS1.
  - ピースS1で人口最大郡をi, iを含む他ピースで最も0.5に近いピースをS2.
  - S1,S2どちらかのみに属す郡をj
- 案2 :最大分数値のピースをS1.以下同様.
- 分枝 :基本的にSAME(i,j)優先のDFS

3.7. Postprocessing

- 最適解は人口格差有⇒境界線変更で調整
- 調整方法：
  - 区割案の隣接グラフHを作成. 点の重み :人口
  - 輸送問題に定式化
※ 供給点間, 需要点間の移動は好ましくない
⇒各枝にペナルティコストの設定
  - 最小費用輸送問題
  - 得られた輸送案を元に, 隣接部分を微調整

4. South Carolina Case Study



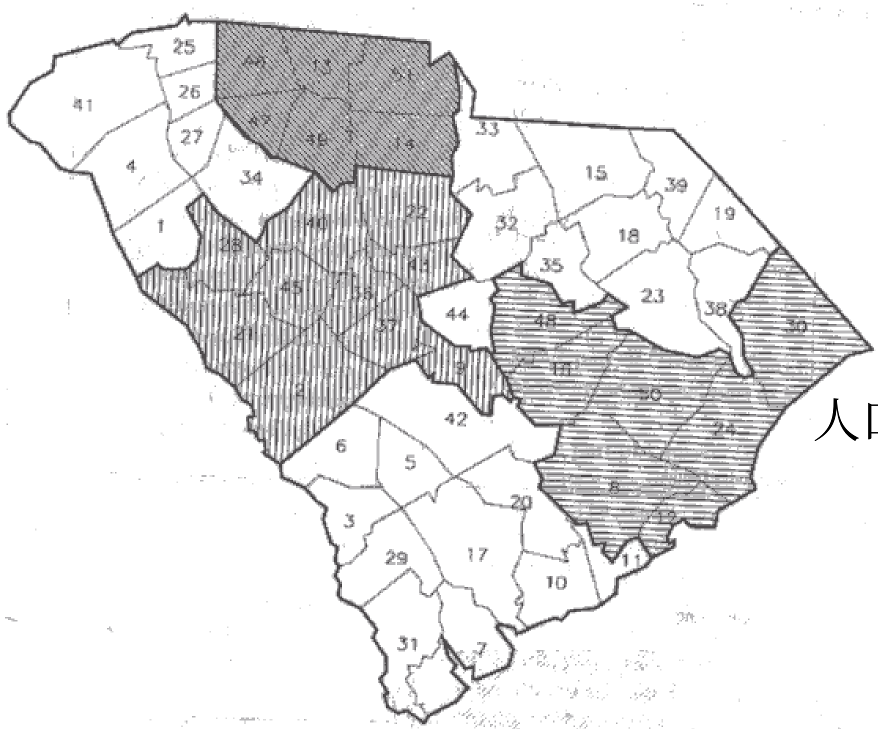
1990年の区割

46郡, 6議席

前処理で
•47郡
•隣接関係 約100

特徴
•13郡が2区に分割

4.2 Starting Solution and the Optimization Phase



人口幅5%許容の場合の最適解

人口格差2%許容の場合

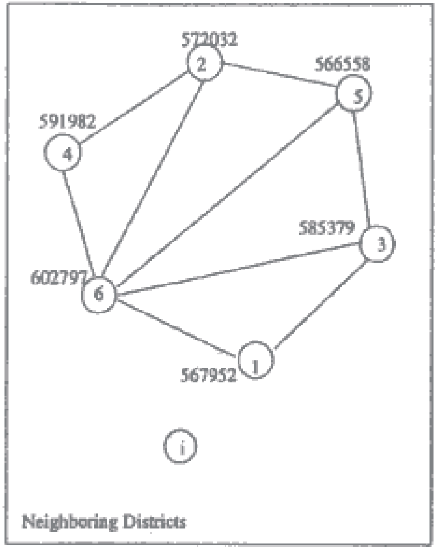
ヒューリスティクスによる初期解
•実行可能解は得られなかった
↓
最適解
•最適値 71, 平均からの差 1.86%
•生成カラム 1581本, 分枝木ノード数 61個

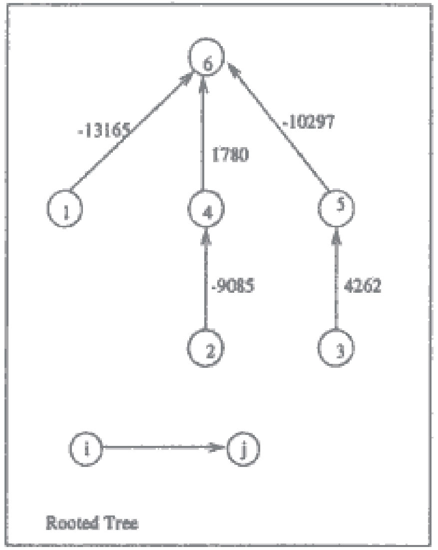
人口格差5%許容の場合

ヒューリスティクスによる初期解
•最適値 69, 平均からの差 4.37%
↓
最適解
•最適値 68, 平均からの差 3.73%
•生成カラム 802本, 分枝木ノード数 25個

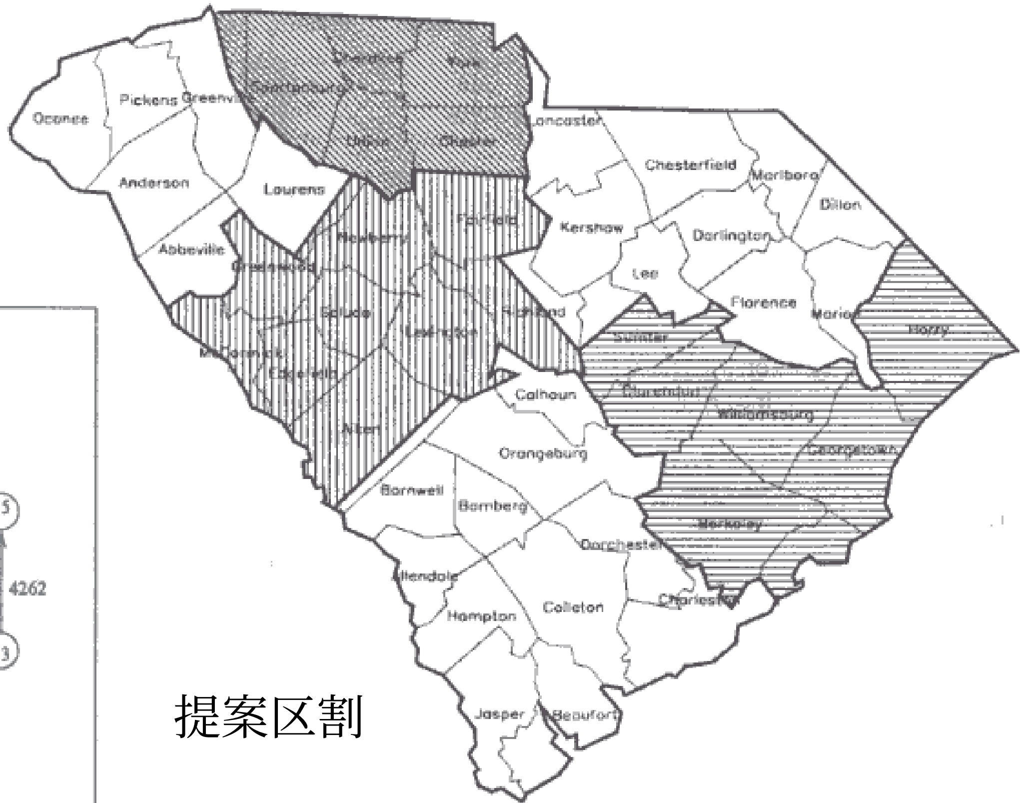
4.3 Postprocessing

輸送問題としてのモデル化





輸送問題の答え



提案区割

- 分割された郡は6郡のみ
- 人口格差0

5. Concluding Remarks

- ここでの提案アプローチはなかなかいいだろ.
- ほかの州でもやってみたら使えたよ.
- 変種のアプローチも考えられる
- 区割に関する他の基準に使ったらどうだろう.
  → いろいろ可能じゃないかな