

「組合せ最適化問題の自動化  
と現場適用の可能性」  
に関する研究

平成 29 年 3 月

東京工業大学工学院経営工学系  
教授 松井 知己

特定非営利活動法人ビュー・コミュニケーションズ  
副理事長 小松 秀樹

## はじめに

最適化問題の研究は古くから行われてきており、研究成果も相当蓄積されてきた。

一方、企業実務の現場では、問題の把握と設定そのものの困難性が高いため、いまだ研究成果が十分活用されているとは言い難い状況にある。

日本の生産性向上がとみに叫ばれている中で、生産性向上に直結する最適化技法の活用につき改めて研究する意味があると思われる。

今後求められる最適化技法は有効で効果的な技法となっていることを前提に、企業実務上の実践性が確保されかつ取り扱いが容易であることが条件となる。

このような観点から本共同研究では、まず最近の研究成果を概観し、確率的丸め法、最小下限設定、緩和問題化、パーフェクトグラフ活用等による求解の有効性、効果性に関する整理を行った。合せて、企業への実際の用途例を示しながら、現場での実践性、取扱い容易性への道すじを示すためのフレームワークを呈示することとした。

また、将来的には、自動化を志向することで取扱い容易性の解消につながるものと考ええる。

## 目次

はじめに

I 最近の研究成果概観・とりまとめ

II 最適化技法の用途例と研究フレームワーク

資料

1. 多次元割当問題
2. 船舶の航行速度最適化問題
3. 航空路ネットワーク設計問題
4. 多重彩色当問題
5. スポーツスケジューリング
6. 和音に対するピアノ運指決定法
7. 安定マッチングの戦略的操作可能性
8. パーフェクトサンプリング法

## 執筆分担

東京工業大学工学院経営工学系

I 最近の研究成果概観・とりまとめ

資料

特定非営利活動法人ビュー・コミュニケーションズ

はじめに

II 最適化技法の用途例と研究フレームワーク

## I 最近の研究成果概観・とりまとめ

## 1. 多次元割当問題

**Key words:** データフュージョン, 多ターゲット・多センサーシステム

多次元割当問題は, データフュージョンにおける基本的な問題である. 同一(と見做すことのできる)母集団に対する複数のデータが存在するとき, このデータの連結を行う作業はデータフュージョンと呼ばれる. 例としては, 特定地域のユーザに対する異なるアンケート調査結果が複数あるとき, アンケート調査においては匿名であるユーザに対し, 年齢・性別等の基本データを元にマッチングを行う作業が多次元割当問題であり, これを元にアンケート調査を結合が行われる. このような手法は統計的マッチング(**statistical matching**)と呼ばれることもある. 医療工学の分野では, コンピューター断層撮影法(**CT: Computed Tomography**)や陽電子放射断層撮影(**PET: Positron Emission Tomography**)および磁気共鳴画像(**MRI: Magnetic Resonance Imaging**)等の異なる装置で撮影された画像を統合する際に用いられ, 統合されたデータを用いて1つの3次元画像として表示するシステムに組み込まれている事が多い. この場合は, 各画像において捉えられた臓器や患部等の部位のマッチングを行う作業が, 多次元割当問題となる. 軍事利用の場面では, 多数のターゲットを複数のセンサで確認した際に, 各センサで補足されたターゲットのマッチングを行う作業が, 多次元割当問題となる. 本研究では, 対象となるターゲットに対し, 観測データは多次元正規分布に従うと仮定した際の最尤推定を行う問題を扱っている. この問題自体は **NP-困難**と呼ばれる, 理論的に効率的な解法が存在が絶望視されている問題クラスに含まれることが知られている. 本研究では, この問題に対し対数尤度に関する近似比率の保証を持ち, かつ求解時間が短い解法(多項式時間解法)を提案している. 解法は, 近年研究が長足の進歩を示している2次錐計画を用いた, 確率的丸め法を基礎としている.

## 2. 船舶の航行速度最適化問題

**Key word:** 燃料消費最小化, 資源配分問題,

原油価格高騰と環境問題への関心高まりから, 船舶の航行スケジュール効率化が注目を浴びている. 従来の船舶航行スケジュールでは, 船舶の設計段階において設計された経済的速度を用いて運航することが通常であった. これに対し, スケジュール毎に異なる寄港日程を視野に入れて, 燃料消費量を最小化する航行速度を決定する問題は, 航行速度最適化問題と呼ばれ, 近年研究が始まっている. 本研究では, 従来の寄港日程の設定に加え, 港間の積載量と, 港に置かれた積載予定荷物の在庫コストを含めた費用最小航行速度を求める問題を扱っている. 本研究で提案する解法は, 寄港日程を無視した部分スケジュール多数生成し, これらを組み合わせる事で費用最小の航行速度を求めるというものである. 最小費用の下界を効果的に用いることで, 生成する部分スケジュール数を大幅に減少し, 求解時間の短縮に成功している.

### 3. 航空路ネットワーク設計問題

**Key word:** ハブアンドスポークネットワーク, 2次整数計画, 線形化手法

航空ネットワークの設計において, 広範囲の航空需要を満たす必要のある場合, すべての空港に直行便を設けることは, 運行数を増やすばかりであり著しく不経済である. そこで, 運行の中継点として「ハブ空港」と呼ばれる乗り換え空港となる大型の空港を設置し, ハブ空港を中心とした交通ネットワークを構築する必要性が生じてくる. 「ハブ・アンド・スポーク・ネットワーク」の構築により広範囲に点在する貨物を効率良く運ぶことができるため, 航空会社にとってメリットは大きい. この「ハブ・アンド・スポーク・ネットワーク」を適切に構築することは航空会社にとって重要な課題である. 日本では 2009 年に ANA が, 那覇空港をベースに, 日本とアジアを結ぶ巨大な物流ハブを構築した. 同様の構造は, 宅配便ネットワークにも存在し, 「ハブ・アンド・スポーク・ネットワーク」の設計問題は物流における重要な問題となっている. 本研究では, 容量制約のない単一割当ハブ空港配置問題を解く効率的な近似解法を提案している. 解法は, 本来は整数 2 次計画となる問題を線形化手法を用いて整数線形計画問題に変換し, その緩和問題を解いて得られる解を確率的に丸める方法となっている.

### 4. 多重彩色当問題

**Key word:** 周波数割当問題, パーフェクトグラフ, 近似解法

グラフ彩色問題は, 有名な 4 色問題を含む, グラフ理論における基本的な問題であり, 古くより研究されている. 更に近年では, 携帯電話ネットワーク等の周波数割当問題が多重彩色問題となることが判明し, 1970 年代より数多くの研究がされている. 周波数割当問題とは, 携帯電話に代表される情報機器間をつなぐ無線周波数を効率的に活用するために, 各アンテナへの周波数の割当を決定する問題である. 実務において周波数割当問題を解く場面は, 地球規模のネットワークの設計や, 災害時の緊急無線網の設営等, 様々な状況が存在し, 状況毎に異なる要求に答える, 様々な性能を持つアルゴリズムを導く枠組みを構築する必要がある. 本研究ではグラフ彩色問題の拡張である多重彩色問題と周波数割当問題に対する近似解法の枠組みを提案し, 種々のグラフクラスに適用した際の理論的な近似精度及び必要計算量等を与えている. 本研究で提案する解法に共通する特徴として, パーフェクトグラフあるいは理想グラフと呼ばれる数学的に美しい構造を持つグラフクラスの特徴を積極的に用いている点が挙げられる.

### 5. スポーツスケジューリング

**Key word:** ホームアウェイ割当問題, 半正定値計画

スポーツの試合スケジュール作成の研究は, 2000 年頃よりアメリカを中心に大きく

発展し、ソフトウェア会社の新規参入分野として注目を浴びている。日本においては、Jリーグにおいて試合日程を作成するソフトウェアが2004年より導入されたが、他のプロスポーツ界においては試合スケジュール作成にソフトウェア会社等が積極的に介入した例はまだなく、試合スケジュールの作成は数名の担当者の勘と経験に頼っている事が殆どである。

日本のプロスポーツ界の多くはマネージメント面において大きな転換期を迎えようとしており、今後数年間の内に、日本プロスポーツにおいて収益管理やその透明性について詳細に議論されるようになることは間違いないだろう。プロスポーツの収入の中心は、試合観戦によるものであり、魅力的な「試合スケジュール」によって収入増加を図る事は、少ない投資で堅実な収入の増加を見込める最も重要な手段の一つである。更に、試合場と試合日程を巧妙に選択してチームの移動や宿泊の費用を抑えることは、第2リーグや学生リーグにとっては重要な課題である。本研究では、スポーツの試合日程を作成する問題に対し、その効率的な解法を提案している。対象とする問題は、試合会場を定めるもの、試合日程を定めるもの、会場と日程の両方を同時に定めるもの等、多岐に渡っている。それぞれの問題に対し、線形化手法や半正定値計画緩和等をもちいた近似解法や発見的解法の提案を行っている。

## 6. 安定結婚問題の戦略的操作可能性

**Key word:** ゲーム理論, マッチング問題, 安定結婚問題

安定結婚問題は、1962年に Gale と Shapley によって提起された問題である。男性と女性のグループが、互いに対する選好リストを持つとき、彼らの論文において、安定なマッチング(1対1対応)が必ず存在することが証明されており、これを求める解法も提案されている。彼らの研究は2012年のノーベル経済学賞の原点ともなっている。彼らの解法が現実において使用されている例も多く、日本では平成15年から実施されている医師臨床研修マッチング制度において使用されている。他には学校選択制度においても使用されている事がある。彼らの解法を用いるとき、参加者の一部が自らの選好を偽る(嘘をつく)動機(誘因)を持つ事が古くから知られているが、その構造については未解決のままであった。本研究では、参加者が嘘をつくことによって可能となる結果(得られるマッチング)に対する特徴付けを初めて与えている。更に、参加者の一部のみが選好を偽る事が許される場合については、その特徴付けを確認する事が計算量の観点から困難であることを示した。

## 7. パーフェクトサンプリング法

**Key word:** マルコフ連鎖モンテカルロ法, サンプリング法

マルコフ連鎖モンテカルロ法(MCMC法)は、待ち行列ネットワーク、ゲノム解析、金融等様々な分野において活用されており、既に古典的な方法と言ってよい。しかしな

がら、MCMC 法は乱数を用いた計算法であるため再現性が無く、得られた解の質の理論的保証が(多くの場合)存在しない。この欠点は、社会的に重要なコンピュータネットワークの解析や、食品の安全性に関わる遺伝子解析、公的な監査がある保険に関わる業務等では大きな問題点となる。MCMC 法の適用範囲が広がるにつれて、精度保証の存在する解法への要求は今後増えてゆくと予想される。これに対し近年、MCMC 法において理論的な近似精度の保証を与える全多項式時間近似乱拓スキーム (FPRAS) に関する研究が、アルゴリズム分野において急速に進んでいる。精度保証を可能とする FPRAS を構築するには、サンプリング法の理論的な精度保証が必要不可欠である。本研究で対象とするパーフェクトサンプリング法を構築することで、MCMC 法の精度保証を行う事が可能となる。1996 年に Propp と Wilson によって提案された CFTP(coupling from the past)法は、マルコフ連鎖の極限分布に厳密に従うサンプリング (パーフェクトサンプリング) を可能とする驚くべき手法である。日本においては CFTP 法の認知度は未だ低く、その研究を継続的に行っている研究者は非常に少ない。本研究では (医療統計等において重要な『正確検定』を行う際に必要となる) 分割表を生成する CFTP 法、ゲノム解析等の確率モデルで多用される(離散)ディリクレ分布に従うサンプリングを行う CFTP 法、待ち行列ネットワークの基本モデルである Jackson 型ネットワークの定常分布に従うサンプリングを行う CFTP 法の構築に成功している。



## Ⅱ 最適化技法の用途例と研究フレームワーク

## 1. 最適化技法の用途例 ALP

最近の小売、サービス市場で労働力不足がよく言われるようになってきた。また、IT技術の進展により、自動発注やロボット化の検討・導入も進み出した。

このような市場環境の中で、ALP (Auto Labor Planning,シフトスケジューリング) へのニーズは十分あると考えられる。

スケジューリングには既に各種最適化技法 (解法) が存在しているが、問題の把握・設定、解法の選択もしくは解法そのものを構築することの困難性が横たわっている。

また、既存のスケジューリングソフト (ex. 看護師の Excel シフト表作成) も現実問題の一部 (ex. A 看護師と B 看護師の組合せは避けたい) にしか対応しておらず、より複雑で不確定要素の多い実務では、その利用は必ずしも十分とは言えない状況にある (実際には人の手でスケジューリングされている)。このような現状を考えると、現実的・効果的な ALP を研究する意義は高いと考えられる。

## 2. 研究フレームの考察

### (1) 現実の問題について

従来のシフトスケジュールのフレームは、与えられた時間枠（ex. 月曜日から日曜日の勤務時間帯）に対して、各人別勤務可能状況（ex. 日勤、早出、遅出区分）や各人の組合せ適否条件等を考慮したシフトスケジュールを計算するものが多い。

作業区分毎に必要なスキルや必要工数が異なることや、人の代替性の程度やボトルネック作業の認識具合により、管理者は勘と経験によってコントローラブルと考えるシフト表を相当の時間を割いて、手作業で作成することとなる。

つまり少なくとも作業区分を加味してシフトスケジュールを考察する必要があることとなる。

そこで、スーパーマーケットを例に、作業区分を加味したスケジュールはどういうフレームで考えるべきか、その試案を整理してみることにする。

### (2) スーパーマーケットの作業区分

食品スーパーを例に、一般従業員（パートを含む）の主要作業区分の大枠を示すと下表のようになっている。

部門 作業区分	1 ドライ グロサリー 〔一般食品、 酒、雑貨菓 子等〕	2 チルドグ ロサリー 〔精肉 日配品 低温工業品〕	3 青果 〔野菜 果物 農産品等〕	4 鮮魚 〔生魚 塩干 水産品〕	5 デリカ 〔惣菜 米飯 バーカリー〕	6 フロント 〔レジ サービス カウンター等〕
a 入荷	○	○	○	○	○	—
b 陳列	○	○	○	○	○	—
c 包装	—	○	○	○	○	—
d カット、調理	—	—	—	—	○	—
e 値引、廃棄	○	○	○	○	○	—
f 棚メンテ	○	○	○	○	○	—
g 清掃、事務	○	○	○	○	○	—
h その他	○	○	○	○	○	○

### (3) 標準作業工数

シフトスケジュールを決める上では、作業区分ごとに必要工数（人時間）が与えられていなければならない。

しかし、現場の売り場において、表にある部門別作業区分別の必要工数が標準設定されていることは、ほとんどないと考えられる。

来店客数の変化、地域イベントの状況、天候要因、メディアの影響、ライフスタイルの変化など、標準工数を設定したとしても、市場環境変化によりその値の妥当性が危うくなってしまふ。また、動作研究、時間分析などの詳細分析作業を積み上げ合理的に標準工数を設定するノウハウも当業界に十分あるとは言えない状況にもある。

つまり、需給条件を十分加味した標準工数の設定は極めて困難と言わざるを得ないのが実情である。

そこで、発想を大きく転換し、事業側の管理目標から標準工数の設定方法を考えてみることにする。

### (4) 管理目標型標準工数の設定について

以下の①～④の手順で部門別必要作業工数（標準工数）を設定する。

以下、部門  $j$  の所定期間（ex. 1日）における必要工数を考える。

①労働生産性を定義する　ここでは、粗利労働生産性を用いることとする。

$$\text{粗利労働生産性} = \frac{\alpha \cdot V \text{ (粗利益、所定期間)}}{L^S \text{ (労働賃金総額、所定期間)}}$$

$$\alpha : \text{粗利率、} V : \text{売上高、} L^S = \sum_{i=1}^n w_i l_i^S$$

( $w_i$ 、 $i$ の賃率、 $l_i^S$ 、 $i$ の労働時間、労働者数  $n$ )

②管理目標を設定する　粗利労働生産性係数を $\delta$ とおくと、 $\delta$ の値を目標値として事業者が設定する。

$$\delta = \frac{\alpha \cdot V}{L^S}$$

③部門別売上予測を行う　売上看見合う労働投入を所定目標（ $\delta$ ）のもとで行うために、売上予測が必要となる。

$$\text{売上予測 } \hat{V} = \sum_{j=1}^m \hat{V}_j$$

(部門  $j$ 、部門数  $m$ )

④日々の部門別必要作業工数表を作成する

$$\frac{\alpha \cdot \hat{V}_j}{\sum_{i=1}^n w_{ij} l_{ij}^s} = \delta_j = \delta$$

( $\delta_1 = \delta_2 = \dots = \delta_j$ 、生産性係数は部門間で共通と仮定)

ここで簡素化のために  $w_{ij} = w$  とすると

$$\sum_{i=1}^n l_{ij}^s = \frac{\alpha \cdot \hat{V}_j}{w \cdot \delta} = \frac{\alpha \cdot \hat{V}_j}{w \cdot \delta}$$

より、部門  $j$  の必要総労働工数  $l_j^s$  ( $= \sum_{i=1}^n l_{ij}^s$ ) は粗利予測額を賃率で換算した労働生産性係数で除した値で与えられる。

例として、グロサリー部門の1日当りの作業区分別必要工数表を示してみた。

	8:00~	11:00~	13:00~	16:00~	19:00~	21:00~	計
入荷	0.5hr×2人	—	—	—	—	—	1人 hr
陳列	0.5hr×2人	1hr×1人	1hr×1人	1hr×1人	1hr×1人	0.5hr×2人	6人 hr
包装・カット	—	—	—	—	—	—	—
調理	—	—	—	—	—	—	—
値引・廃棄	0.5hr×1人	—	—	1hr×1人	0.5hr×1人	0.5hr×1人	2.5人 hr
棚メンテ	—	1hr×1人	—	0.5hr×1人	—	1hr×2人	3.5人 hr
清掃等	—	—	—	—	0.5hr×1人	0.5hr×1人	1人 hr
事務等	—	0.5hr×1人	—	0.5hr×1人	—	0.5hr×1人	1.5人 hr
計	2.5人 hr	2.5人 hr	1人 hr	3人 hr	2人 hr	4.5人 hr	15.5人 hr

$$\frac{\alpha \cdot \hat{V}_j}{w \cdot \delta_j}$$

この表は、 $j$  部門の1日の作業区分別時間帯別必要工数表となっている。

8×6 行列形式で次のような条件が与えられている。

$$\begin{array}{l}
 \text{計} \quad \text{作業区分必要工数} \\
 \begin{pmatrix} h_{11} & h_{12} & \cdots & h_{16} \\ h_{21} & h_{22} & \cdots & h_{26} \\ \vdots & \vdots & \cdots & \vdots \\ h_{81} & h_{82} & \cdots & h_{86} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_8 \end{pmatrix} \quad h_{kt} : t \text{ 時間帯の作業 } k \text{ の必要工数} \\
 \\
 \text{計} \quad (h_1 \ h_2 \ \cdots \ h_6) \quad \frac{\alpha \cdot \bar{V}}{w \cdot \delta} \quad h_{kt} \leq 2 \sim 3 \\
 \\
 \text{時間帯必要工数} \quad \sum_{k=1}^8 h_{k1} + \cdots + \sum_{k=1}^8 h_{k6} = \sum_{t=1}^6 h_{1t} + \cdots + \sum_{t=1}^6 h_{8t} = \frac{\alpha \cdot \bar{V}}{w \cdot \delta}
 \end{array}$$

管理者は計算された j 部門の総必要工数（人時間）  $\frac{\alpha \cdot \bar{V}}{w \cdot \delta}$  を 8 作業区分に割付ける ( $u_1 \sim u_8$ )。

その後に、時間帯別に各作業工数を再割付けする。（尚、時給賃金の条件より、 $h_{kt}$  は整数値を取るものと仮定する。）

このような割付け作業は売上とほぼ比例すると想定される店舗規模タイプ別に行うと、店舗運営ノウハウの蓄積から考えてさほど困難性はないと思われる。

尚、実際の検討に際しては、

- ✓ 曜日別に必要工数は異なるのか
- ✓ 同一作業区分の工数は時間帯別に代替性があるのか
- ✓ 作業項目の優先順位はつけられるのか
- ✓ 時間帯別ボトルネック作業はあるのか
- ✓ 作業難易度、専門性の判定区分は可能か

など、より詳細な分析も必要となる。

(5) 労働力調達表

労働市場のやりとりを通して、管理者は以下のような出勤可能表を作成する。

曜日	月				火				日			
時間帯 人				...				...				
1	○	○			○	○						
2		○	○						.....		○	○
3		○	○			○	○					
.												
.												
.												
n		○				○				○		

○ : 出勤可能時間帯

$L^D = \sum_{i=1}^n l_i^D$   $l_i^D$  : 個人の出勤可能時間、 $l_i^D = \sum_{t=1}^M l_{it}^D$ 、 $M=7 \times$ 時間帯区分数(t)  
尚、 $L^D$ と $L^S$ は必ずしも一致しない。

(6) 部門別作業区分別必要工数表と部門別各人別出勤可能時間帯表

管理目標から導かれる「部門別作業区分別必要工数表」と労働需給関係から導かれる「部門別各人別出勤可能時間帯表」とを突き合せ、労働生産性が最大となる作業シフトスケジュールを求めることがテーマである。

パラメータ化された  $\delta$ を最大とする、各人別労働時間  $l_{it}^D$  を各作業区分別時間帯別必要工数 (人時間)  $h_{kt}^S$ にどう割付けるかを求める問題と設定できる。

ただし、各時間帯において  $\sum_{i=1}^n l_{it}^D < \sum_{k=1}^g h_{kt}^S$  (供給可能労働時間 < 必要工数) のときは、作業時間帯の拡張を可能とする条件が追加されなければならない。

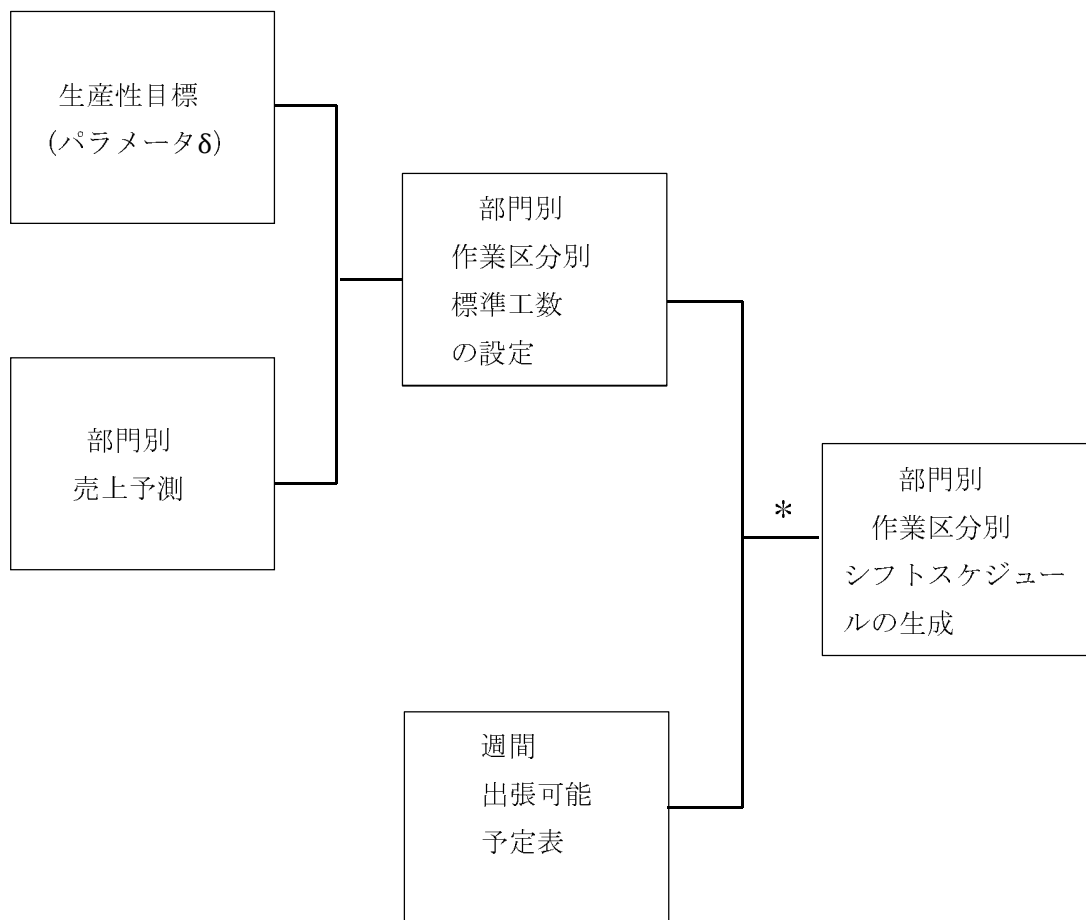
その他作業優先度、ボトルネック作業の状況等勘案すべき条件が存在している。

また、ここまでは同一賃金同一労働といった仮定で考えてきたが、実際には作業区分毎に作業難易度、専門性の程度が異なったり、労働の質も必ずしも均質ではなく差があるのが通常である。

例えば、労働者への一定の人事評価 (ex. せいぜい A,B,C 程度) が与えられ、作業区分ごとに求められる要求能力が定義できると仮定すると、各人の各作業への割付け方

法も異なったものとなり、より複雑（例えば賃率  $w$  が異なる）な問題となる。  
いずれにせよ、最適化問題としてどのように定式化し、どのように解を求めて行くか  
今後詳細な研究が必要となる。

(7) 研究フレームフロー図



\* 条件付最適化技法の研究



## 資料

1. 多次元割当問題  
データフュージョン、多ターゲット・多センサーシステム
2. 船舶の航行速度最適化問題  
燃料消費最小化、資源配分問題、最短路問題
3. 航空路ネットワーク設計問題  
ハブアンドスポークネットワーク、2次整数計画、線形化手法
4. 多重彩色当問題  
周波数割当問題、パーフェクトグラフ、近似解法
5. スポーツスケジューリング  
ホームアウェイ割当問題、半正定値計画
6. 和音に対するピアノ運指決定法  
運指決定問題、逆最短路問題
7. 安定マッチングの戦略的操作可能性  
ゲーム理論、マッチング問題、安定結婚問題
8. パーフェクトサンプリング法  
マルコフ連鎖モンテカルロ法、サンプリング

### 多次元割当問題

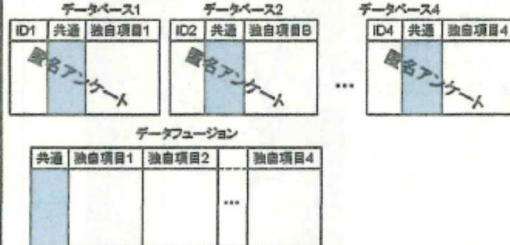
Y. Kuroki and T. Matsui, "An Approximation Algorithm for Multidimensional Assignment Problems Minimizing the Sum of Squared Errors," *Discrete Applied Mathematics*, 157 (2009).

黒木 裕介  
松井 知己

### データフュージョンモデル

2006年頃：マーケティングの勉強をしていた。  
木戸 茂, 『広告マネジメント』, 朝倉書店, 2004.

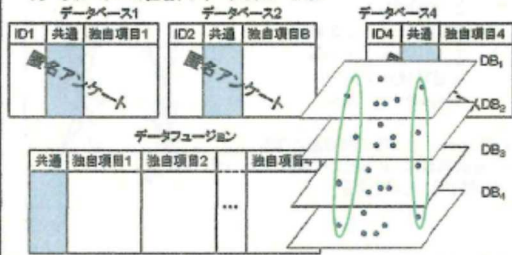
同一グループへの匿名アンケートのフュージョン



### データフュージョンモデル

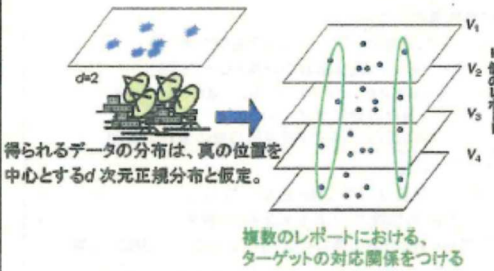
2006年頃：マーケティングの勉強をしていた。  
木戸 茂, 『広告マネジメント』, 朝倉書店, 2004.

同一グループへの匿名アンケートのフュージョン



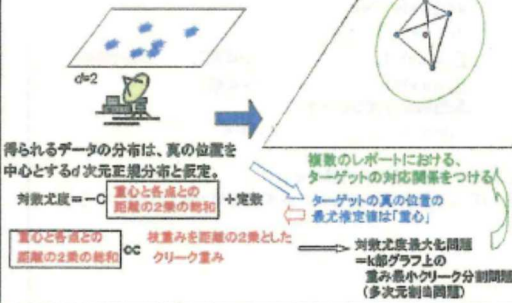
### 多ターゲット・多センサーシステム

多ターゲット・多センサーシステムにおける位置の同定



### 多ターゲット・多センサーシステム

多ターゲット・多センサーシステムにおける位置の同定



### k次元割当問題(枝重みモデル)

**Input:** 完全k部グラフ  $G=(V_1, V_2, \dots, V_k, E)$   
ただし  $|V_1|=|V_2|=\dots=|V_k|=n$ ,  $E$ は完全k部グラフとなる枝集合  
非負枝重み  $w: E \rightarrow \mathbb{R}$ .

**Output:**  $G$ 中の  $n$ 個の  $k$ クリークで以下を満たすもの

- (1)互いに素 ( $G$ の頂点集合の分割となっている)
- 各  $V_j$ から頂点を1つずつ選んだ, 互いに素なクリーク  $n$ 個
- (2)クリーク重みの総和が最小のもの

注: クリーク  $Q$ の重み =  $Q$ 中の2頂点間の枝の重みの総和

最小重みの決定

この中の問題は, 中に解答が送れている  
「組合せ最適化」の第10章「グラフ上の最適化」で扱った  
「完全k部グラフの枝重みを最小にするクリーク」の問題

**k次元割当問題(枝重みモデル)**

**Input:** 完全k部グラフ $G=(V_1, V_2, \dots, V_k; E)$   
 ただし $|V_1|=|V_2|=\dots=|V_k|=n$ ,  $E$ は完全k部グラフとなる枝集合  
 非負枝重み $w: E \rightarrow R_+$

**Output:**  $G$ 中の $n$ 個の $k$ クリーク  
 (1)互いに素( $G$ の頂点を各 $V_j$ から頂点を1つずつ選んだ、互いに素なクリーク $n$ 個)  
 (2)クリーク重みの総和が最小のもの  
 注: クリーク $Q$ の重み= $Q$ 中の2頂点間の枝の重みの総和

本研究での仮定:  
 (1) $G$ 中の頂点は、 $R^d$ に埋め込まれている  
 (2)頂点 $u, v$ 間の枝の重み $w(u, v) = \|u - v\|^2$

**k次元割当問題(枝重みモデル)**

**Input:** 完全k部グラフ $G=(V_1, V_2, \dots, V_k; E)$   
 ただし $|V_1|=|V_2|=\dots=|V_k|=n$ ,  $E$ は完全k部グラフとなる枝集合  
 非負枝重み $w: E \rightarrow R_+$

**Output:**  $G$ 中の $n$ 個の $k$ クリーク  
 (1)互いに素( $G$ の頂点を各 $V_j$ から頂点を1つずつ選んだ、互いに素なクリーク $n$ 個)  
 (2)クリーク重みの総和が最小のもの  
 注: クリーク $Q$ の重み= $Q$ 中の2頂点間の枝の重みの総和

本研究での仮定:  
 (1) $G$ 中の頂点は、 $R^d$ に埋め込まれている  
 (2)頂点 $u, v$ 間の枝の重み $w(u, v) = \|u - v\|^2$

**k次元割当問題(枝重みモデル)**

**Input:** 完全k部グラフ $G=(V_1, V_2, \dots, V_k; E)$   
 ただし $|V_1|=|V_2|=\dots=|V_k|=n$ ,  $E$ は完全k部グラフとなる枝集合  
 非負枝重み $w: E \rightarrow R_+$

**Output:**  $G$ 中の $n$ 個の $k$ クリークで以下を満たすもの  
 (1)互いに素( $G$ の頂点集合の分割となっている)  
 各 $V_j$ から頂点を1つずつ選んだ、互いに素なクリーク $n$ 個  
 (2)クリーク重みの総和が最小のもの  
 注: クリーク $Q$ の重み= $Q$ 中の2頂点間の枝の重みの総和

本研究での仮定:  
 (1) $G$ 中の頂点は、 $R^d$ に埋め込まれている。  
 頂点 $v$ の位置ベクトルを $v$ (太字)で表す  
 (2)頂点 $u, v$ 間の枝の重み $w(u, v) = \|u - v\|^2$  (ユークリッド距離の2乗)

**線形目的関数を持つ整数計画としての定式化**

グラフの枝に変数を定義する

$$x(e) = \begin{cases} 1 & (\text{枝}e\text{をクリーク枝として選ぶ}) \\ 0 & (\text{選ばない}) \end{cases}$$

min.  $\sum_e w(e)x(e)$   
 s. t.  $\sum_u x(u, v) = 1$  ( $u \in \text{略}$ ),  
 $\sum_v x(u, v) = 1$  ( $v \in \text{略}$ ),  
 各連結成分はクリーク,  
 $x(e) \in \{0, 1\}$  ( $\forall e$ ).

枝重み $w(e)$ は、頂点間距離の2乗  
 → 目的関数は、ある凸2次関数 $f_d(x)$ でも表現することができる。

**凸2次目的関数の導入**

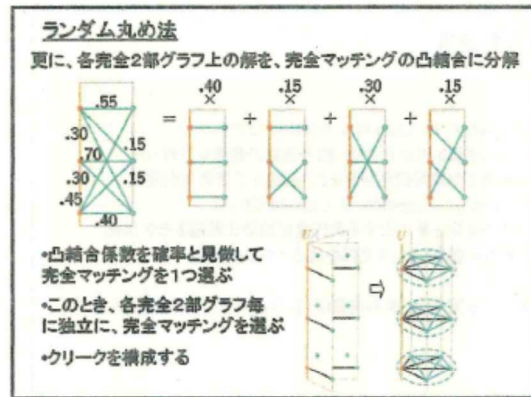
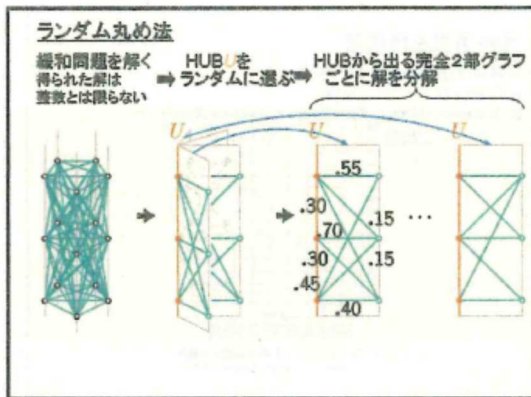
min.  $z$  新しい変数の導入  
 s. t.  $z \geq \sum_e w(e)x(e)$  最適値の上界  
 $z \geq f_d(x)$  整数計画のときは、同じ値を取るので冗長

$\sum_u x(u, v) = 1$  ( $u \in \text{略}$ ), 凸2次関数  
 $\sum_v x(u, v) = 1$  ( $v \in \text{略}$ ),  
 各連結成分はクリーク,  
 $x(e) \in \{0, 1\}$  ( $\forall e$ ).

**連続緩和問題**

min.  $z$   
 s. t.  $z \geq \sum_e w(e)x(e)$ ,  
 $z \geq f_d(x)$ , 凸2次制約  
 $\sum_u x(u, v) = 1$  ( $u \in \text{略}$ ),  
 $\sum_v x(u, v) = 1$  ( $v \in \text{略}$ ),  
~~各連結成分はクリーク~~,  
 $x(e) \geq 0$  ( $\forall e$ ).

凸計画なので、効率的に解ける。  
 例えば、2次計画に変形して多項式時間で解くことができる



### 既存の研究と、本研究の結果

k	距離	近似精度	
2	一般	Class P	古典的な最適問題
3	三角不等式	4/3-近似	Crama&Spieksma(1992)
k:一般	三角不等式	(2-2/k)-近似	Bandelt, Crama & Spieksma (1994)
k:一般	R <sup>k</sup> に埋め込むユークリッド距離の2乗	(4-6/k)-近似	Bandelt, Crama & Spieksma (1994)
k:一般	R <sup>k</sup> に埋め込むユークリッド距離の2乗	(5/2-3/k)-近似	本研究

## 船舶の航行速度最適化問題

梶野修平 河瀬康志 松井知己

### 研究背景

**社会的背景**  
 原油価格の高騰・海運市場の停滞  
 ⇒ 貨物船等の船舶運用コスト最適化に対する関心の高まり  
 燃料消費の観点から、温室効果ガスの排出量削減も期待

従来、船舶のスケジューリング・ルーティングは固定された航行速度のもと計画されてきた

数値計画の手法を用いて  
 航行速度も変数としたスケジューリングを行う

### 取り扱うスケジュールの例

Port	Distance (nautical miles)	Time window (days)
Antwerp	0	[0, 0]
Millers Haven	510	[1, 3]
Boston	2699	[9, 13]
Charleston	376	[11, 15]
Algeciras	2025	[20, 24]
Point Lisas	3037	[32, 36]
Houston	2263	[35, 39]

- ルートは固定(与えられているものとする)
- 各港の到着時間に関する制約として、“時間枠制約”を設定

### 解法のアイデア

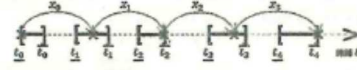
時間制約を取り除いた緩和スケジューリング問題の最適解を考える

**緩和問題の最適スケジュールが、全ての時間制約を満たしている**  
 ⇒ 緩和問題の最適解は、元の問題でも最適解

**時間制約を満たしていない**  
 ⇒ 時間制約を満たさない港で到着時間を固定して、港の前後で問題を分割

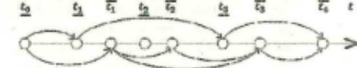
### グラフの生成

スケジュールのイメージ:



次のルールでグラフGを生成する

1. 頂点集合  $V = U \cup \{t_0, \bar{t}_n\} \cup \{t_k, \bar{t}_k\}$
2. 各頂点対  $(u, v) \in V^2$  ( $u < v$ ) に対して、 $u$  を始点、 $v$  を終点とする時間制約を取り除いたスケジューリング問題を解く  
 ⇒ 得られた最適スケジュールが時間制約を満たすならば、 $u$  から  $v$  への有向辺を生成 (枝長は得られた最小費用に設定)

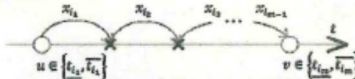


グラフG上の  $t_0$  から  $t_n$  までの最短路は、総費用最小スケジュール

### 枝生成のための部分問題

グラフGの枝を生成するため、

$u$  を始点、 $v$  を終点とする時間制約を取り除いたスケジューリング問題を解く必要がある



$$\begin{aligned}
 \text{P1: Min. } & \sum_{k=1}^{m-1} f_k(x_k) \\
 \text{s.t. } & \sum_{k=1}^{m-1} x_k = v - u, \\
 & x_k \geq d_k / v_{\max} \quad (k \in \{1, \dots, m-1\}).
 \end{aligned}$$

- 上の問題は、資源配分問題と捉えられる
- T. Ibaraki and N. Katoh (1988) の解法を改訂

### 最短経路問題の解法

最適スケジュールを得るためには、

- ▷ 時間制約を取り除いた資源配分問題を解く
- ▷ グラフGを生成
- ▷ グラフG上の最短経路を求める。

計算回数削減のため、最短経路問題の3つの解法を提案

### 最短経路問題の解法

探索方法	枝の生成方法 (SCRの呼び出し回数に関係)
Dijkstra 1	ダイクストラ法 すべて生成
Dijkstra 2	ダイクストラ法 呼び出し時、逐次生成
A*	A*アルゴリズム 呼び出し時、逐次生成

- A\*アルゴリズム ... 各頂点  $v$  からゴールまでの距離の下界値  $h(v)$  を用いた、ダイクストラ法の一般化アルゴリズム
- $h(v)$  は、 $v$  から  $\bar{t}_n$  までの緩和問題 (P1) の最適スケジュールに設定

### 実験インスタンス

K. Fagerholt, G. Laporte, I. Norstad (2010) の方法に基づいた。

- $v_{\max} = 20$  (knot), 標準航行速度: 14 knot
- $w_i \in [0, w_{\max}]$ ,  $w_{\max} = 60000$  (t)
- $d_i \in [650, 6500]$  (nautical mile) (2-20 日に相当)
- $c_1 \propto w_i$ ,  $c_1 = 10000$  (\$/day) when  $c_1 = 1000$  (t)

### パラメータ

1. 航路上の港総数: {8, 16, 32}
2. 時間許容  $b = \{5, 10\}$  (days)
  - ▷  $\{t_i, \bar{t}_i\} \in \{[u_i - 0.75b, u_i + 0.25b], [u_i - 0.25b, u_i + 0.75b]\}$
  - <  $u_i$  は標準航行速度のときの各港の到着時間 >
3. 燃料単価  $P_{\text{FUEL}} = \{400, 800\}$  (\$)

### 計算結果(1/2)

3解法での「資源割当問題」を解いた回数

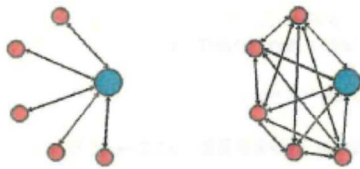
港数 $n$	時間単位	燃料消費	Dijkstra 1	Dijkstra 2	A*
5	400	120	113.5 (0.946)	74 (0.617)	
	800	120	111.5 (0.929)	72.2 (0.635)	
	10	120	112.6 (0.928)	57.9 (0.483)	
15	400	496	487.5 (0.983)	379.1 (0.764)	
	800	496	481.0 (0.976)	273.6 (0.552)	
	10	496	487.0 (0.982)	305.8 (0.617)	
32	400	2016	2003.7 (0.995)	1848.1 (0.917)	
	800	2016	1997.9 (0.991)	1846.2 (0.917)	
	10	2016	2008.5 (0.996)	1733.9 (0.855)	

・サイズにもよるが、A\* で全体的に計算回数が抑えられている

### 燃料消費削減のための 航空路ネットワーク設計問題

岩佐大, 齊藤廣大, 松井知己

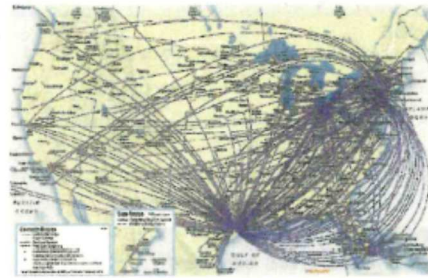
### 航空路線形態



**ハブ・アンド・スポーク型**  
ハブ空港を中心に、放射状に  
周辺への路線を設計

**ポイント・トゥ・ポイント型**  
ハブ空港を経由せず空港間を  
直接結ぶ路線を設計

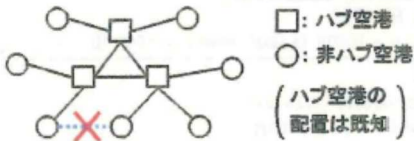
### Hub Network



コンチネンタル航空 ([http://www.airlineroutemaps.com/USAC/Continental\\_Airlines\\_usa.shtml](http://www.airlineroutemaps.com/USAC/Continental_Airlines_usa.shtml))

### 問題設定

- 目的: 空港間の輸送量と単位輸送コストが与えられたとき、総輸送コストが最小となるような接続を求める



- 各ハブ空港は互いに直行便を持つ
- 各非ハブ空港は、唯一のハブ空港に接続
- 非ハブ空港間の直行便は存在しない

### 記号の定義

- $H$  : ハブ空港の集合
- $N$  : 非ハブ空港の集合
- $c_{ij}$  : 空港  $i, j$  間の単位輸送コスト  
(非負,  $c_{ij} = c_{ji}$  を仮定,  
ハブ間で三角不等式を満たす)
- $w_{ij}$  : 空港  $i$  から  $j$  への輸送量  
(非負を仮定)

### 定式化(四2次0-1整数計画問題)

$$\begin{aligned} \min. & \sum_{(p,q) \in N \times N} w_{pq} (\sum_{i \in H} c_{pi} x_{pi} \\ & + \sum_{(i,j) \in H \times H} c_{ij} x_{pi} x_{qj} + \sum_{j \in H} c_{jq} x_{qj}) \\ \text{s.t.} & \sum_{i \in H} x_{pi} = 1, \quad \forall p \in N, \\ & x_{pi} \in \{0, 1\}, \quad \forall (p, i) \in N \times H. \end{aligned}$$

$$x_{pi} = \begin{cases} 1 & \text{非ハブ } p \text{ はハブ } i \text{ に接続する} \\ 0 & \text{非ハブ } p \text{ はハブ } i \text{ に接続しない} \end{cases}$$

▶ 目的関数の詳細は次のスライドで解説する

### 目的関数

$$\begin{aligned} & \sum_{(p,q) \in N \times N} w_{pq} \\ & (\sum_{i \in H} c_{pi} x_{pi} \\ & + \sum_{(i,j) \in H \times H} c_{ij} x_{pi} x_{qj} \\ & + \sum_{j \in H} c_{jq} x_{qj}) \end{aligned}$$

すべての非ハブペア $(p,q)$  ( $p$  から  $q$  への輸送量)  
 ( $p$  から接続するハブ  $i$  へ  
 +ハブ  $i$  からハブ  $j$  へ  
 +  $q$  に接続するハブ  $j$  から  $q$  へ)

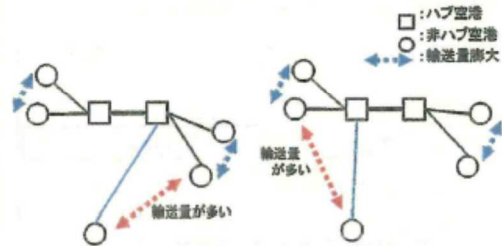
$i = j$  のとき  
 $c_{ij} = 0$  となり、  
 この項はゼロとなる

### 定式化(四2次0-1整数計画問題)

$$\begin{aligned} \min. & \sum_{(p,q) \in N \times N} w_{pq} (\sum_{i \in H} c_{pi} x_{pi} \\ & + \sum_{(i,j) \in H \times H} c_{ij} x_{pi} x_{qj} + \sum_{j \in H} c_{jq} x_{qj}) \\ \text{s.t.} & \sum_{i \in H} x_{pi} = 1, \quad \forall p \in N, \\ & x_{pi} \in \{0, 1\}, \quad \forall (p, i) \in N \times H. \end{aligned}$$

### 問題の難しさ

各非ハブを一番近いハブに接続すれば良い、訳ではない



### ハブ・アンド・スポーク型の先行研究

- (O'Kelly, 1987) ハブ空港の配置と、接続を決定  
 →2次計画問題として定式化、発見的解法
- (Sohn & Park, 2000) ハブ空港3つ、接続のみ決定  
 →NP困難の証明、厳密解法(分枝限定法)

### 定式化(線形化手法)

四2次0-1整数計画問題  
 ↓(線形化手法: linearization)  
 0-1整数線形計画  
 線形緩和問題 → 計算機実験

**定式化(0-1整数計画問題)**

$$\min. \sum_{(p,q) \in N \times N} w_{pq} (\sum_{i \in H} c_{pi} x_{pi} + \sum_{(i,j) \in H \times H} c_{ij} x_{pi} x_{qj} + \sum_{j \in H} c_{jq} x_{qj})$$

s.t.  $\sum_{i \in H} x_{pi} = 1, \quad \forall p \in N,$   $y_{piqj}$  とおく.  
 $x_{pi} \in \{0, 1\}, \quad \forall (p, i) \in N \times H.$

↓

- 非線形制約 ( $y_{piqj} = x_{pi} x_{qj}$ ) を変形
- 整数制約を緩和 (線形化手法)

線形緩和問題

**定式化(0-1整数線形計画)**

$$\min. \sum_{(p,q) \in N \times N} w_{pq} (\sum_{i \in H} c_{pi} x_{pi} + \sum_{(i,j) \in H \times H} c_{ij} y_{piqj} + \sum_{j \in H} c_{jq} x_{qj})$$

s.t.  $\sum_{i \in H} x_{pi} = 1, \quad \forall p \in N,$   
 $x_{pi} \in \{0, 1\}$ ,  $\forall (p, i) \in N \times H,$   
 $\sum_{j \in H} y_{piqj} = x_{pi}, \quad \forall (p, q) \in N \times N, \forall i \in H,$   
 $\sum_{i \in H} y_{piqj} = x_{qj}, \quad \forall (p, q) \in N \times N, \forall j \in H,$   
 $y_{piqj} \geq 0, \quad \forall (p, q, i, j) \in N \times N \times H \times H.$

目的関数は線形(0-1変数有り)

**定式化(0-1整数線形計画)**

$$\min. \sum_{(p,q) \in N \times N} w_{pq} (\sum_{i \in H} c_{pi} x_{pi} + \sum_{(i,j) \in H \times H} c_{ij} y_{piqj} + \sum_{j \in H} c_{jq} x_{qj})$$

s.t.  $\sum_{i \in H} x_{pi} = 1, \quad \forall p \in N,$   
 $x_{pi} \in \{0, 1\}$ ,  $\forall (p, i) \in N \times H,$   
 $\sum_{j \in H} y_{piqj} = x_{pi}, \quad \forall (p, q) \in N \times N, \forall i \in H,$   
 $\sum_{i \in H} y_{piqj} = x_{qj}, \quad \forall (p, q) \in N \times N, \forall j \in H,$   
 $y_{piqj} \geq 0, \quad \forall (p, q, i, j) \in N \times N \times H \times H.$

目的関数は線形(0-1変数有り)

**線形緩和問題**  
(線形計画問題なので単体法等で簡単に解ける)

$$\min. \sum_{(p,q) \in N \times N} w_{pq} (\sum_{i \in H} c_{pi} x_{pi} + \sum_{(i,j) \in H \times H} c_{ij} y_{piqj} + \sum_{j \in H} c_{jq} x_{qj})$$

s.t.  $\sum_{i \in H} x_{pi} = 1, \quad \forall p \in N,$   
 $0 \leq x_{pi} \leq 1$ ,  $\forall (p, i) \in N \times H,$   
 $\sum_{j \in H} y_{piqj} = x_{pi}, \quad \forall (p, q) \in N \times N, \forall i \in H,$   
 $\sum_{i \in H} y_{piqj} = x_{qj}, \quad \forall (p, q) \in N \times N, \forall j \in H,$   
 $y_{piqj} \geq 0, \quad \forall (p, q, i, j) \in N \times N \times H \times H.$

目的関数も制約もすべて線形

**計算機実験**

CABデータ: 米25都市の航空データ  
 APデータ: オーストラリアの郵便

**ハブ数=3**

全ての問題例で、線形緩和問題が整数解を与えた  
(最適整数解が線形計画で得られる)

**ハブ数=4**

ハブ数+非ハブ数	10	15	20	25
解いた問題数	840	5460	19380	50600
非整数解となった問題	0	6	4	16

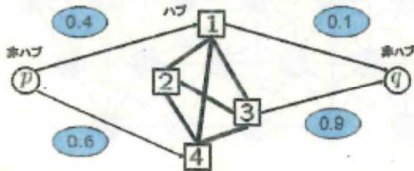
**確率的2-近似解法**

線形緩和問題を解く  
 独立丸め法



## 2-近似解法

- Step 1:** 線形緩和問題を解き、最適解  $(x^*, y^*)$  を得る。  
**Step 2:** 各非ハブ空港  $p \in N$  に対し、 $x_{pi}^*$  の確率で独立に  $x_{pi}^*$  を1に丸める。(近似解  $X_{pi}$  を得る。)  
**Step 3:**  $Y_{piqj} = X_{pi} \cdot X_{qj}$  の式から、近似解  $Y_{piqj}$  を計算。



50

## 2-近似解法

- Step 1:** 線形緩和問題を解き、最適解  $(x^*, y^*)$  を得る。  
**Step 2:** 各非ハブ空港  $p \in N$  に対し、 $x_{pi}^*$  の確率で独立に  $x_{pi}^*$  を1に丸める。(近似解  $X_{pi}$  を得る。)  
**Step 3:**  $Y_{piqj} = X_{pi} \cdot X_{qj}$  の式から、近似解  $Y_{piqj}$  を計算。

**定理:** 上記の解法で得られる解の目的関数値は、最適値(最小値)の2倍以下である

51

## 多重彩色当問題

Y. Miyamoto and T. Matsui,  
 "Perfectness and imperfections of unit disk graphs on triangular  
 lattice points," Discrete Mathematics, 309 (2009).

宮本 裕一郎  
 松井 知己

## 携帯電話のアンテナへの周波数割当問題

### 携帯電話のしくみ

- Q. 近くで使っても、なぜ混線しないのか?  
 A. 違う周波数チャンネルを使っているから。  
 Q. 周波数チャンネルは、足りるのか?  
 A. アンテナが周波数を持っているから。  
 (携帯電話によって、使用する周波数チャンネルが固定されているわけではない)  
 遠いアンテナは、同じ周波数を使う。

## 携帯電話を使う

利用可能エリア全体を、複数のサービスエリアで分割。

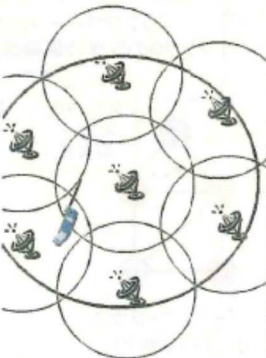
各サービスエリアを、複数のアンテナでカバー。

アンテナの有効範囲は、数km。

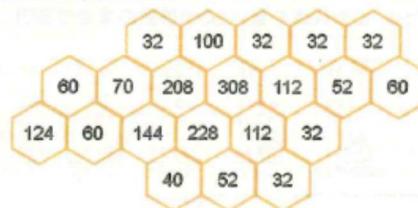
サービスエリアのコンピュータに、(常に)自分の番号を発信。

かけるときは、最も近くのアンテナが使用している周波数チャンネルを教えてください。

かかって来た際は、番号が登録されたサービスエリアを探す。

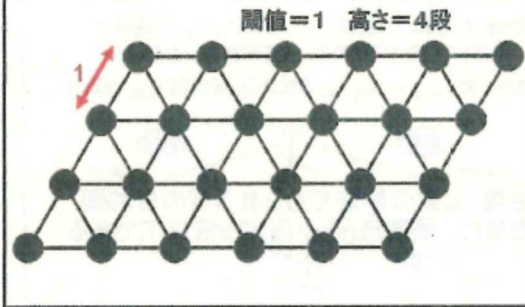


## 周波数割当問題の例

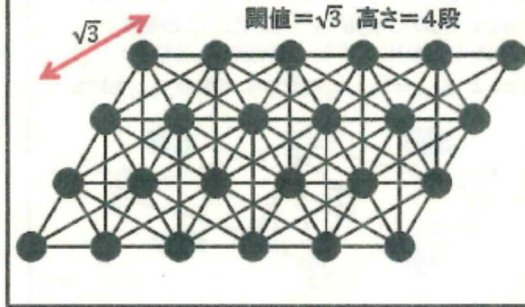


- 各セル(中心に基地局)に所与の数のチャンネルを割り当て
- 隣り合うセル内に割り当てたチャンネルは相異なるもの
- 使うチャンネル数最小化

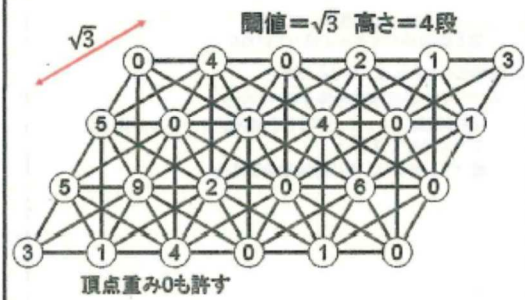
三角格子点上の単位円グラフ



三角格子点上の単位円グラフ



三角格子点上の頂点重み付き単位円グラフ



多重彩色問題(multicoloring problem)

入力: 単純無向グラフ  $G=(V,E)$   
 頂点重み関数  $w: V \rightarrow \mathbb{Z}^+$   
 目的関数値 = 6  
 [4,5,6] 3  
 [1] 1 2  
 [2,3] 2 0  
 [2,3] [ ]

出力: 多重彩色関数  $c: V \rightarrow \mathbb{Z}^n$   
 目的: 必要色数(最大値)最小化  
 制約:  $|c(v)|=w(v), \forall v \in V$   
 (頂点  $v$  は  $w(v)$  色必要)  
 $c(u) \cap c(v) = \emptyset, \forall \{u,v\} \in E$   
 (隣接頂点対は同じ色を共有しない)

閾値=1のときの多重彩色の近似解法

簡単のため、どの頂点の重みも3の倍数の場合で説明

<p>6頂のグラフ</p>	<table border="1"> <tr><td>3</td><td>6</td><td>3</td><td>9</td></tr> <tr><td>3</td><td>9</td><td>3</td><td>6</td></tr> <tr><td>6</td><td>3</td><td>9</td><td>3</td></tr> <tr><td>9</td><td>6</td><td>3</td><td>9</td></tr> </table>	3	6	3	9	3	9	3	6	6	3	9	3	9	6	3	9	<table border="1"> <tr><td>1</td><td>2</td><td>1</td><td>3</td></tr> <tr><td>1</td><td>3</td><td>1</td><td>2</td></tr> <tr><td>2</td><td>1</td><td>3</td><td>1</td></tr> <tr><td>3</td><td>2</td><td>1</td><td>3</td></tr> </table>	1	2	1	3	1	3	1	2	2	1	3	1	3	2	1	3	<table border="1"> <tr><td>1</td><td>2</td><td>1</td><td>3</td></tr> <tr><td>1</td><td>3</td><td>1</td><td>2</td></tr> <tr><td>2</td><td>1</td><td>3</td><td>1</td></tr> <tr><td>3</td><td>2</td><td>1</td><td>3</td></tr> </table>	1	2	1	3	1	3	1	2	2	1	3	1	3	2	1	3	<table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	6	3	9																																																																	
3	9	3	6																																																																	
6	3	9	3																																																																	
9	6	3	9																																																																	
1	2	1	3																																																																	
1	3	1	2																																																																	
2	1	3	1																																																																	
3	2	1	3																																																																	
1	2	1	3																																																																	
1	3	1	2																																																																	
2	1	3	1																																																																	
3	2	1	3																																																																	
0	0	0	0																																																																	
0	0	0	0																																																																	
0	0	0	0																																																																	
0	0	0	0																																																																	

レイヤ1    レイヤ2    レイヤ3    レイヤ4

頂定理を4つのレイヤに分解

必要色数  $\leq 4/3 \times \chi(G,w)$

0でない重みは元の重みのちょうど1/3

$\leq 1/3 \times w(G,w) \leq 1/3 \times \chi(G,w)$

[Hochbaum (1987)]  
 のshifting strategy  
 に類似

定義 パーフェクトグラフ

グラフ  $G$  がパーフェクトであるとは、  
 グラフ  $G$  の任意の誘導部分グラフ  $H$  において

クリーク数  $\omega(H) = \chi(H)$  彩色数

クリークの  
 大きさの  
 最大値

通常彩色で  
 必要な色数  
 の最小値

パーフェクトグラフ  
 (perfect graph)

パーフェクトでないグラフ  
 (imperfect graph)

### 多重彩色問題とパーフェクトグラフ

頂点重み付きグラフ  $(G, w)$

記法

$w(G, w)$ :  $(G, w)$  のクリークの頂点重み和の最大値  
 $\chi(G, w)$ :  $(G, w)$  の多重彩色に必要な最小色数

頂点重み付きグラフに関して以下の定理が知られている

定理 [Grötschel, Lovász & Schrijver (1981)]

グラフ  $G$  がパーフェクトならば、任意の重み  $w$  に関して

$$w(G, w) = \chi(G, w)$$

重み付きクリーク数 = 多重彩色数

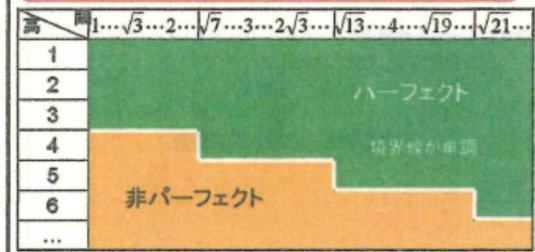
頂点重み付きグラフ  $(G, w)$  の最適多重彩色は (強) 多項式時間で求まる。

### 三角格子点上の単位円グラフの分類

【定理】

高さ  $\leq 3$  段: パーフェクト

高さ  $\geq 4$  段: パーフェクト  $\Leftrightarrow$  閾値  $\geq \sqrt{\text{高}^2 - 3\text{高} + 3}$



### 近似解法の性能

近似率



閾値=2: 提案する (5/3)-近似解法 < (7/3)-近似解法  
 [Feder & Shende (2000)]

### スポーツスケジューリング

松井知己

### スポーツ・スケジューリング

スポーツ・スケジューリング (対戦日程計画)

Timetabling の一分野

スポーツ競技などの最適な対戦日程を決める  
 スケジュール作成アルゴリズムの構築

初期の研究: 1970 年代後半

ここ数年で研究が活発化

Timetabling の中でも大きな分野に

### スケジューリングの実務における現状 (日本)

**プロ野球 セリーグ:** 6球団, 年間420試合

球団の営業担当者と運営職員で構成される「日程編成会議」で作成。リーグ理事会の承認を経て前年秋ごろに発表。詳細は非公開。

**サッカー J1リーグ:**

10チーム(1993年)から18チーム(2005年)の総当たり戦。  
 ソフトウェア「日程君」導入 (新日鉄住金ソリューションズ)

**アイスホッケー アジアリーグ:** 9チーム 全171試合

非公開

**アメリカンフットボール 社会人リーグ:**

18チーム(6チーム×3ディビジョン)による総当たり戦。  
 年間45試合  
 事務局長が作成。

**スポーツ・スケジュールリングの実例**

スケジュールリングの目的  
 移動距離最小化, 観客数最大化,  
 公平性最大化, ... → 収益最大化

スケジュールへの要求  
 試合開催場所, 固定された試合,  
 TV中継, 各チームの強さ, 予備日, ...

現実問題から発生した研究分野

**場所付き総当りリーグ戦**

各チームに本拠地があり, どちらかの本拠地で試合  
 本拠地での試合: ホームゲーム  
 遠征先での試合: アウェイゲーム

	1	2	3	
A:	C	B	D	← A対D (Aの本拠地)
B:	D	A	C	
C:	A	D	B	← 対応
D:	B	C	A	← A対D (Aの本拠地)

スケジュール(4チーム)

**試合場所の割当(HAT)**

①	②	③	④	⑤	①	②	③	④	⑤		
巨人:	中	阪	横	広	ヤ	巨:	A	H	A	H	
阪神:	ヤ	巨	中	横	広	阪:	A	H	A	A	H
広島:	横	中	ヤ	巨	阪	→ 広:	H	H	A	H	A
中日:	巨	広	阪	ヤ	横	中:	H	A	H	A	H
横浜:	広	ヤ	巨	阪	中	横:	A	H	A	H	A
ヤフ:	阪	横	広	中	巨	ヤ:	H	A	H	H	A

Home-Away Table (HAT)

各試合をどちらの本拠地で行う? H:ホーム  
 A:アウェイ

**スケジュールの質の指標 ブレーク数**

HA割当の質の指標:「ブレーク」  
 チーム*t*がスロット*s*にブレークを持つ ⇔

$\dots s-1 \ s \ \dots$   
 $t: \dots A \ A \ \dots$  もしくは  $t: \dots H \ H \ \dots$

$t: A \ A \ A \ A \ H \ H \ A \rightarrow$  (チーム*t*のブレーク数) = 4

ブレーク数の意味で最も良いHATは?  
 (ブレーク数が少ない方が良い)

HHブレークとAAブレークは,  
 (スロット毎に)同じ数.

AA  
 AA  
 AH  
 AH  
 AH  
 HA  
 HA  
 HA  
 HH  
 HH

**ブレーク**

HA割当の質の指標:「ブレーク」

チーム*t*がスロット*s*にブレークを持つ ⇔

$\dots s-1 \ s \ \dots$   
 $t: \dots A \ A \ \dots$  もしくは  $t: \dots H \ H \ \dots$

$t: A \ A \ A \ A \ H \ H \ A \rightarrow$  (チーム*t*のブレーク数) = 4

HHブレークとAAブレークは, (スロット毎に)同じ数.

AA  
 AA  
 AH  
 HA  
 HH  
 HH

**実例**

実例では?  
 3Hと3Aの禁止(4Hと4Aの禁止?)  
 各チームのブレークは2回程度

### 2001年のJリーグ前半戦のスケジュール

磐田	: H A H H H H H A H H A H	1
市原	: H A H H H H H A H H A H	3
名古屋	: H A H H H H H A H H A H	1
浦和	: H A H H H H H A H H A H	4
C大阪	: H A H H H H H A H H A H	1
札幌	: H A H H H H H A H H A H	1
福岡	: H A H H H H H A H H A H	3
G大阪	: H A H H H H H A H H A H	1
横浜FM	: H A H H H H H A H H A H	3
神戸	: H A H H H H H A H H A H	1
F東京	: H A H H H H H A H H A H	3
東京V	: H A H H H H H A H H A H	3
柏	: H A H H H H H A H H A H	2
清水	: H A H H H H H A H H A H	1
鹿島	: H A H H H H H A H H A H	1
広島	: H A H H H H H A H H A H	3

16チーム 32

### 2001年のJリーグ後半戦のスケジュール

磐田	: H A H H H H A A H H H A	3
市原	: H A H H H H A A H H H A	2
名古屋	: H A H H H H A A H H H A	3
浦和	: H A H H H H A A H H H A	2
C大阪	: H A H H H H A A H H H A	3
札幌	: H A H H H H A A H H H A	3
福岡	: H A H H H H A A H H H A	2
G大阪	: H A H H H H A A H H H A	3
横浜FM	: H A H H H H A A H H H A	1
神戸	: H A H H H H A A H H H A	1
F東京	: H A H H H H A A H H H A	1
東京V	: H A H H H H A A H H H A	1
柏	: H A H H H H A A H H H A	2
清水	: H A H H H H A A H H H A	3
鹿島	: H A H H H H A A H H H A	1
広島	: H A H H H H A A H H H A	1

16チーム

### 2001年のJリーグのスケジュール

磐田	: HH	1	磐田	: AA AA	HH	3		
市原	: HH HH	HH	3	市原	: AA AA	HH	2	
名古屋	: HH	HH	HH	1	名古屋	: AA AA	HH	3
浦和	: HH	HH	AA	4	浦和	: AA AA	HH	2
C大阪	: AA	AA	HH	1	C大阪	: HH AA	HH	3
札幌	: AA	AA	HH	1	札幌	: HH HH	AA	3
福岡	: AA	AA	HH	3	福岡	: HH HH	AA	2
G大阪	: HH	AA	HH	1	G大阪	: AA HH	AA	3
横浜FM	: HH	AA	HH	3	横浜FM	: HH	HH	1
神戸	: HH	AA	HH	1	神戸	: AA	HH	1
F東京	: AA	AA	HH	3	F東京	: HH	HH	1
東京V	: HH	AA	AA	3	東京V	: AA	HH	1
柏	: AA	AA	HH	2	柏	: HH HH	HH	2
清水	: AA	AA	HH	1	清水	: HH HH	AA	3
鹿島	: AA	AA	HH	1	鹿島	: HH	HH	1
広島	: HH	HH	AA	3	広島	: AA	HH	1

### J league (2006)

Kashima	HH HH	3	Kashima	AA AA	AA	3
Urawa	HH HH	3	Urawa	AA	AAA	3
Omiya	HH HH	3	Omiya	AA	AAA	3
Chiba	HH HH	HH	Chiba	AA AA	AA	3
FTokyo	HH HH	HH	FTokyo	AA AA	AA	3
Kawasaki	HH HH	HH	Kawasaki	AA	AAA	3
YokohamaFM	HH HH	HH	YokohamaFM	AA AA	AA	3
Kofu	HH HH	HH	Kofu	AA AA	AA	3
Niigata	HH HH	HH	Niigata	AAA	AA	3
Shimizu	HH HH	HH	Shimizu	AA AA	AA	2
Iwata	HH HH	HH	Iwata	AA AA	AA	2
Nagoya	HH HH	HH	Nagoya	AA AA	AA	2
Kyoto	HH HH	HH	Kyoto	AA AA	AA	3
Osaka	HH HH	HH	Osaka	AA AA	AA	3
COsaka	HH HH	HH	COsaka	AA AA	AA	3
Hiroshima	HH HH	HH	Hiroshima	AA AA	AA	3
Fukuoka	HH HH	HH	Fukuoka	AA AA	AA	2
Oita	HH HH	HH	Oita	AA	AAA	3

### Serie A (Italy) (2000/2001)

Atalanta	: HH HH	2	Atalanta	: AA AA	2
Bari	: HH HH	2	Bari	: AA AA	2
Bologna	: HH HH	2	Bologna	: AA	1
Brescia	: HH HH	2	Brescia	: AA AA	2
Fiorentina	: HH HH	2	Fiorentina	: AA AA	2
Inter	: HH HH	2	Inter	: AA AA	2
Juventus	: HH HH	2	Juventus	: AA AA	2
Lazio	: HH HH	2	Lazio	: AA	1
Lecco	: HH HH	2	Lecco	: AA AA	2
Milan	: HH HH	2	Milan	: AA AA	2
Napoli	: HH HH	2	Napoli	: AA AA	2
Parma	: HH HH	2	Parma	: AA AA	2
Perugia	: HH HH	2	Perugia	: AA AA	2
Reggina	: HH	1	Reggina	: AA AA	2
Roma	: HH	1	Roma	: AA AA	2
Udinese	: HH	1	Udinese	: AA AA	2
Verona	: HH HH	2	Verona	: AA AA	2
Vioenza	: HH HH	2	Vioenza	: AA AA	2

### スケジュールの作成方法 [Regin等]

①②③          ①②③          ①②③

1 : 2 4 3    1 : A H A    1 : H H H  
2 : 1 3 4    2 : H H A    2 : A A A  
3 : 4 2 1    3 : A A H    3 : H H A  
4 : 3 1 2    4 : H A H    4 : A A H, ...

入力:スケジュール    出力:対応するHA割当

制約条件: 対戦ペアの片方はH, 片方がA (対戦制約)

→ 目的関数: "ブレーク数"

### ブレイク数最適化問題

#### ブレイク数最小化/最大化問題

入力: スケジュール (チーム数  $2n$ )

出力: 対戦制約を満たすHA割当のうち、  
ブレイク数が最小/最大のもの。

→ 最小化, 最大化の意味, 先行研究

### ブレイク数最小化/最大化の目的

1: A A A A A	1: A H H A A
2: A H A H A	2: H H A A H
3: H H A A H	3: A H A H A
4: H A H A A	4: H A A H H
5: A A H H H	5: A A H A H
6: H H H H H (14)	6: H A A H A (3)

#### ブレイク数最小化の目的

A と H を交互に (AAによる選手の消耗を防ぐ)

#### ブレイク数最大化の目的: 総移動回数最小化 AH HA AA

総移動回数 =  $2n(2n-1) - (1/2)(\text{ブレイク数})$

→ 経験的に, 移動距離最小化問題の良い解を与える  
(アメリカマイナーリーグ野球等での研究)

### ブレイク数 最小化/最大化の先行研究

#### ブレイク数最小化問題

NP-困難[Post&Woeginger2006]

先行研究 (分枝限定ベース)

Regin (1998), 制約論理, ~チーム数20

Trick (2000), 整数計画, ~チーム数22

Elf, Jünger, Rinadi (2003),

MAX CUT + ABACUS ~チーム数26

### 計算時間

チーム数	Regin (CP)	Trick (IP)	Elf 他 (MAX CUT)
16	5	34	1
18	80	43	6
20	5603	1092	9
22		7802	37
24			73
26			339

(単位: 秒)

CPU	200MHz	266MHz	300MHz
環境	Solver (ILOG)	CPLEX (ILOG)	ABACUS

### ブレイク数 最小化/最大化 を解くアルゴリズム

線形計画/半正定値計画緩和を用いた  
多項式時間確率的アルゴリズム

MAX(MIN) RES CUT として定式化

LP緩和: ランダム化丸め法を適用

SDP緩和: Goemans, Williamson (1995) の

0.878 確率的近似アルゴリズムを適用

極めて高速な, 質の良い解の生成を確認

### MAX RES CUT

#### MAX RES CUT

入力: 無向グラフ  $G=(V, E)$ , 各枝に対する非負重み  
+ 頂点ペア  $\{v_1, v_2\}, \dots, \{v_n, v_{n+1}\}$

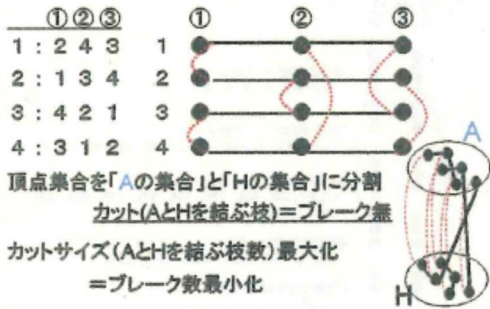
出力: 頂点ペア  $\{v_1, v_2\}, \dots, \{v_n, v_{n+1}\}$  を分け,  
重み最大のカット集合

#### NP-困難

Goemans, Williamson (1995) による

0.878 確率的近似アルゴリズム

### ブレイク数最小化問題と MAX RES CUT



### 計算時間の比較

チーム数	Reain	Trick	Elf et al.	本研究 [秒]
16	1	6.5	0.3	2.1
18	18	13	2	3.4
20	1245	323	3	4.9
22		678	12	6.0
24			24	12
26		(2ヶ月)	111	18
30				38
40				179

(CPU: 900 MHz, SDP Solver: SDPA 6.0, 超平面分離1000回)

### 計算実験の結果

チーム数	最適値 (IP)	ブレイク数最小化の		近似比率
		SDP緩和	差	
16	192	192	0	1.000
18	248	248	0	1.000
20	312	310	2	0.994
22	382	380	2	0.995
24	458	452	4	0.989
26	540 or 542	534	6 or 8	0.985<
30	—	720	?	?
40	—	1306	?	?

(超平面分離1000回)

### 和音に対するピアノ運指決定法

若松万紗子, 松井知己

### 研究の背景と目的

ピアノの上達には適切な指使いをすることが大切である。

しかしながら、楽譜には全ての音に対して運指が記載されているわけではない。

本研究の目的

- ピアノの楽譜データから、弾きやすい運指を出力するシステムを構築すること。
- 和音に対応した運指決定法を提案すること。

### 運指決定法

- 音の集合  $S = \{21, 22, \dots, 108\}$   
(88鍵盤のピアノで発音可能な領域)
- 指の集合  $\{1, 2, 3, 4, 5\}$   
(1: 親指, 2: 人差し指, 3: 中指, 4: 薬指, 5: 小指)
- 音  $s_j$  を指番号  $j_i$  で弾いた後、音  $s_{i+1}$  を指番号  $j_{i+1}$  で弾くコスト  $c(s_{i+1} - s_i; j_i, j_{i+1})$

大沼, "ピアノの運指決定法," 2010年.  
上記論文のコストを使用。

### 定式化 —単旋律—

隣り合う2つの音間のコストの和が最小となる選指を最適な選指として求める。

入力: 音の系列  $s_1, s_2, \dots, s_n$   
 出力: 指の系列  $f_1, f_2, \dots, f_n$  最小化:  $\sum_{p=1}^{n-1} c(s_{p+1} - s_p, f_p, f_{p+1})$

最短経路を求める。

赤線が最短経路であり、理想的な選指(教師選指)に対応するノースP.

例: 60 62 64 65 67 69 71 72  
 1 2 3 1 2 3 4 5

### 和音に対応した選指決定法 グラフ構造

### コストの設定

総和移動コスト  
 和音コスト  
 平均移動コスト **提案**  
 最大移動コスト **提案**

### 和音に対応した選指決定法

4つのコストを導入

①総和移動コスト  
 和音のそれぞれの音から、次の和音のそれぞれの音に移動するコスト(単旋律で使用したコスト)の和を総和移動コストとする。

$1+3+9+1+2+1=17$

### 和音コストの導入

②和音コスト  
 ある和音をある指使いで弾くときのコストを和音コストとする。

一番低い音の指番号 + 各音間のコスト + 一番低い音と一番高い音間のコスト

$1+2+2+2+3=10$

### 平均移動コスト

③平均移動コスト  
 総和移動コストを枝の本数で割った値を平均移動コストとする。和音数が多い場所のコストが重くなり、その場所だけが最適選指となってしまうことを避ける。

枝数 6

$\frac{1+3+9+1+2+1}{6}=2$



### 重み付け

トータルコスト =  $1 \times (\text{総和移動コスト}) + 1 \times (\text{平均移動コスト}) + 1 \times (\text{最大移動コスト}) + 1 \times (\text{黒鍵}) + 5 \times (\text{和音コスト}) + 15 \times (\text{基本選指}) + 10 \times (\text{困難選指})$

【和音コスト】  
指を広げても届かない選指が選ばれないようにする。

【基本選指】  
一般的に用いられる選指が選ばれるようにする。

【困難選指】  
多くの困難な選指が選ばれないようにする。

104

### 検証結果

楽譜に記載されている選指と最短経路問題を解いて得られた選指との適合率

曲名	①~④で最適なもの		トータルコスト
	適合率	パターン	
美しきアメリカ	82%	②	93%
イ長調の前奏曲	86%	②	80%
アロハオエ	45%	①~④	100%
しずかな歌	100%	③, ④	100%

①総和移動コスト, ②和音コスト, ③平均移動コスト, ④最大移動コスト  
トータルコスト:  $1 \times (\text{総和移動コスト}) + 1 \times (\text{平均移動コスト}) + 1 \times (\text{最大移動コスト}) + 1 \times (\text{黒鍵}) + 5 \times (\text{和音コスト}) + 15 \times (\text{基本選指}) + 10 \times (\text{困難選指})$

### 逆最適化問題

重みのチューニングをおこなう

総和移動コスト, 平均移動コスト, 最大移動コスト, 和音コスト, 黒鍵, 基本選指, 困難選指の重みを  $\alpha_1, \alpha_2, \dots, \alpha_7$  とおく。  
(先程は,  $(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7) = (1, 1, 1, 5, 1, 15, 10)$  としていた。)

選指  $u, v$  間の上記コストを  $f_1(u, v), f_2(u, v), \dots, f_7(u, v)$  と表す。  
これらを用いて選指  $u, v$  間の距離を  $w(u, v) = \alpha_1 f_1(u, v) + \alpha_2 f_2(u, v) + \dots + \alpha_7 f_7(u, v)$  と定める。

理想的な選指(教師選指)が得られているフレーズから生成されるグラフ  $G=(V, A)$  に対して, 教師選指に対応するパス  $P$  が最短経路となるように, 重み  $\alpha_1, \alpha_2, \dots, \alpha_7$  を定める。  
推奨値  $\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_7$  と  $\alpha_1, \alpha_2, \dots, \alpha_7$  の距離  $\left| \frac{\alpha_i - \tilde{\alpha}_i}{\tilde{\alpha}_i} \right| \leq \beta_i$  を最小化する。  
( $(\tilde{\alpha}_1, \tilde{\alpha}_2, \tilde{\alpha}_3, \tilde{\alpha}_4, \tilde{\alpha}_5, \tilde{\alpha}_6, \tilde{\alpha}_7) = (1, 1, 1, 5, 1, 15, 10)$  とする。)

108

### 定式化

$$\min. \beta_1 + \beta_2 + \beta_3 + \beta_4 + \beta_5 + \beta_6 + \beta_7$$

s.t.  $\pi(v) - \pi(u) \geq w(u, v) \quad (\forall (u, v) \in A)$   
 $\pi(v) - \pi(u) = w(u, v) \quad (\forall (u, v) \in P)$  (教師選指に対応するパス)  
 $w(u, v) = \sum_{i=1}^7 \alpha_i f_i(u, v) \quad (\forall (u, v) \in A)$   
 $1 - \beta_i \leq \frac{\alpha_i}{\tilde{\alpha}_i} \leq 1 + \beta_i \quad (\forall i \in \{1, 2, \dots, 7\})$

108

### 実験結果

推奨値

$\tilde{\alpha}_1$	$\tilde{\alpha}_2$	$\tilde{\alpha}_3$	$\tilde{\alpha}_4$	$\tilde{\alpha}_5$	$\tilde{\alpha}_6$	$\tilde{\alpha}_7$
1	1	1	5	1	15	10

得られた重み

	総和 $\alpha_1$	平均 $\alpha_2$	最大 $\alpha_3$	和音 $\alpha_4$	黒鍵 $\alpha_5$	基本 $\alpha_6$	困難 $\alpha_7$
美しきアメリカ	1	1	1	4	1	15	10
イ長調の前奏曲	1	1	1	1	1	15	10
アロハオエ	1	1	1	5	1	15	10
しずかな歌	1	1	1	5	1	15	10

108

### 検証

逆最適化問題を解いて得られた重みを用いて最短経路問題を解いた選指と楽譜に記載された適合率

	推奨値 ( $\tilde{\alpha}_1, \tilde{\alpha}_2, \tilde{\alpha}_3, \tilde{\alpha}_4, \tilde{\alpha}_5, \tilde{\alpha}_6, \tilde{\alpha}_7$ )	得られた重み ( $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7$ )
	美しきアメリカ	
重み	(1, 1, 1, 5, 1, 15, 10)	(1, 1, 1, 4, 1, 15, 10)
適合率	93%	84%
イ長調の前奏曲		
重み	(1, 1, 1, 5, 1, 15, 10)	(1, 1, 1, 1, 1, 15, 10)
適合率	80%	88%

108

### 検証

得られた重みを他の曲に用いた場合の適合率

	美しいアメリカ	イ長調の前奏曲	アロハオエ	しずかな歌
美しいアメリカ (1, 1, 1, 4, 1, 15, 10)	94%	78%	100%	100%
イ長調の前奏曲 (1, 1, 1, 1, 1, 15, 10)	92%	85%	100%	100%
アロハオエ しずかな歌 (1, 1, 1, 5, 1, 15, 10)	93%	80%	100%	100%

110

### まとめと課題

#### まとめ

和音特有の性質を重視することで適合率を上げることができた。

#### 今後の課題

複数の曲に対して、教師選指を出力することができる重みを得る。

コスト設定や規則を増加させる。

音のつながりなど演奏の表現方法も考慮した選指決定法を提案する。

111

### 安定マッチングの戦略的操作可能性

H. Kobayashi and T. Matsui, "Cheating Strategies for the Gale-Shapley Algorithm with Complete Preference Lists," *Algorithmica*, 58 (2010).

H. Kobayashi and T. Matsui, "Successful Manipulation in Stable Marriage Model with Complete Preference Lists," *IEICE TRANSACTIONS on Information and Systems*, E92-D (2009).

小林 弘暹  
松井 知己

### 安定結婚問題と GS-algorithm

男性の選好リスト 好き ⇄ 嫌い	女性の選好リスト 好き ⇄ 嫌い	プロポージングペア
男性1 D A B C	女性A 3 1 4 2	1 ⇄ A
男性2 B C A D	女性B 1 4 3 2	2 ⇄ B
男性3 B D C A	女性C 1 3 2 4	3 ⇄ C
男性4 C A D B	女性D 2 3 4 1	4 ⇄ D

プロポージングペアの無いマッチングを安定マッチングと呼ぶ

Gale and Shapley (1962)

どのような選好リストに対しても、少なくとも一つの安定マッチングが存在する。GSアルゴリズムの提案。

D. Gale, (~2008)

L. Shapley,

A. Roth



2012年  
ノーベル経済学賞

112

### 女性が嘘をつく可能性が示唆される例

男性プロポーズのGSアルゴリズムでは、女性は嘘をつく誘因を持つ場合がある。

男子の選好リスト

女子の選好リスト

	好き ⇄ 嫌い
男性1	B A C
男性2	A B C
男性3	C B

	好き ⇄ 嫌い
女性A	1 2 3
女性B	2 1 3
女性C	3 1 2

### 女性が嘘をつく可能性が示唆される例

男性プロポーズのGSアルゴリズムでは、女性は嘘をつく誘因を持つ場合がある。

女の子の選好リスト

男子の選好リスト

	好き ⇄ 嫌い
男性1	B A C
男性2	A B C
男性3	A C B

	好き ⇄ 嫌い
女性A	1 2 3
女性B	2 1 3
女性C	3 1 2

113

### 女性が嘘をつく可能性が示唆される例

男性プロポーズのGSアルゴリズムでは、女性は嘘をつく誘因を持つ場合がある。

女の子の選好リスト

女性A	1	2	3
女性B	2	1	3
女性C	3	1	2

男子の選好リスト

男性1	B	A	C
男性2	A	B	C
男性3	A	C	B

女性Aは、本来選んでいた、男性3を、一時的に受け入れることによって、より良いパートナーを得ようとしている

### 未解決問題: 可能な嘘のつき方

2007年 小林弘道氏 松井研(中大)所属  
卒論テーマに選ぶ

- 人数の少ない例
- 女性は皆、同じか良くなる
- 嘘をつかなくても良くなる

→作業の解説

女性Aは、男性3を『キープ』にする。

### 主結果

Gale-Shapleyの結果

男性の完全選好リスト      女性の完全選好リスト

Gale-Shapley Algorithm

安定結婚 (安定完全マッチング)

男性の完全選好リスト      女性の完全選好リスト

Gale-Shapley Algorithm

完全マッチング  $O(n^2)$  algorithm

### 定義 (rooted suitor graph)

与えられた完全マッチング  $\mu$

男性のリスト

男性	好き ⇄ 嫌い
A	3 2 1 4
B	2 3 4 1
C	3 1 2 4
D	2 3 4 1

rooted suitor graph

男性 A, B, C, D      女性 1, 2, 3, 4      r (根)

### 定理

与えられた完全マッチング  $\mu$

男性のリスト

男性	好き ⇄ 嫌い
A	3 2 1 4
B	2 3 4 1
C	3 1 2 4
D	2 3 4 1

rooted suitor graph

男性 A, B, C, D      女性 1, 2, 3, 4      r (根)

定理1: 与えられた完全マッチングがGS-alg. で得られるような女性の完全選好リストが存在 ⇔ rooted suitor graph が outgoing spanning tree を持つ

### 定理

与えられた完全マッチング  $\mu$

男性のリスト

男性	好き ⇄ 嫌い
A	3 2 1 4
B	2 3 4 1
C	3 1 2 4
D	2 3 4 1

outgoing spanning tree 存在

女性のリスト

女性	好き ⇄ 嫌い
1	C ? ? ?
2	A B ? ?
3	D C ? ?
4	B ? ? ?

リストの更新法:  
 $\mu$ の相手を1番, outgoing spanning tree の親頂点の男性を2番にする。  
 3番以降は何でも良い。親頂点が根の場合は、2番以降は何でも良い。

### 部分マッチングへの拡張

与えられた部分マッチングを含む完全マッチングが GS algorithm で得られるような女性の完全選好リスト

部分マッチング 部分マッチング

**NP-完全**  **$O(n^2)$**

Algorithms of Matching Under Preferences  
David F. Manlove,  
World Scientific Pub Co Inc., 2013.

### パーフェクトサンプリング法

S. Kijima and T. Matsui,  
Polynomial Time Perfect Sampling Algorithm  
for Two-Rowed Contingency Tables,  
Random Structures and Algorithms, 29 (2006).

来嶋 秀治  
松井 知己

### マルコフ連鎖を用いたサンプリング

1. 目的の分布を定常分布にもつマルコフ連鎖を設計する。  
2. 十分な回数推移させて、定常分布からサンプリングする。

マルコフ連鎖  $M$ :  $\begin{cases} \text{有限離散の状態空間 } \Omega \\ \text{推移確率行列 } P \\ \text{目的の定常分布 } \pi \end{cases}$

状態空間

### サンプリング法

1. 目的の分布を定常分布にもつマルコフ連鎖を設計する。  
2. 十分な回数推移させて、定常分布からサンプリングする。

⇒ 何回推移させれば十分か？

- > 近似サンプリング
  - mixing time (mixing rate)
  - total variation distance
- > パーフェクトサンプリング
  - CFTP (Coupling From The Past)  
Propp and Wilson [1996]
  - 誤差が無い → パラメータ調整不要

状態空間

### 更新関数 (update function)

マルコフ連鎖  $M_C$  (エルゴード的)

- > 状態空間:  $s_1, s_2, s_3$ ; 有限 (3 状態)
- > 推移規則: 乱数  $\lambda \in \{1, \dots, 6\}$  に対して決まる。

更新関数			
	$s_1$	$s_2$	$s_3$
1	$s_3$	$s_1$	$s_2$
2	$s_2$	$s_3$	$s_1$
3	$s_2$	$s_1$	$s_3$
4	$s_1$	$s_1$	$s_3$
5	$s_1$	$s_2$	$s_1$
6	$s_2$	$s_3$	$s_3$

例 5  $\rightarrow \lambda$

時刻  $t$   $t+1$

### CFTPアルゴリズムと定理

マルコフ連鎖  $M_C$ :  $\begin{cases} \text{有限離散の状態空間 } \Omega \\ \text{更新関数 } \Phi_i(x, \lambda) \\ \text{エルゴード性を満たす。} \end{cases}$

**CFTPアルゴリズム**

1. set  $T = -1$ ; set  $\lambda$ : 空列;
2. generate  $\lambda[T]$ : 乱数; put  $\lambda := (\lambda[T], \lambda[T+1], \dots, \lambda[-1])$ ;
3.  $\Omega$  の全ての状態について、時刻  $T$  から 0 まで  $\lambda$  を用いてマルコフ連鎖  $M_C$  を推移させる。
  - a. if coalesce ( $\exists y \in \Omega, \forall x \in \Omega, y = \Phi_y^T(x, \lambda)$ )  $\Rightarrow$  return  $y$ ;
  - b. otherwise, set  $T := T - 1$ ; step 2. に戻る;

**CFTP定理**

上のアルゴリズムが停止して値を返す時、その値はマルコフ連鎖  $M$  の定常分布に厳密に従う確率変数を返す。

CFIPのアイデア 128

◆無限の過去から推移を続ける仮想的なマルコフ連鎖を考える。

- 現在(時刻0)の状態は厳密に定常分布に従う。

◆現在(時刻0)の状態は何か？

- 最近の乱数列から推理する。
- ⇒ 現在の状態を一意に特定する証拠を見つける。

coalescence

CFIPのアイデア 129

無限の過去から推移を続ける仮想的なマルコフ連鎖を考える。

現在(時刻0)の状態は厳密に定常分布に従う。

現在(時刻0)の状態は何か？

最近の乱数列から推理する。

⇒ 現在の状態を一意に特定する証拠を見つける。

時刻

T = -4からのシミュレーション 130

6 3 2 5 → λ

時刻

CFIPのアイデア 131

◆無限の過去から推移を続ける仮想的なマルコフ連鎖を考える。

- 現在(時刻0)の状態は厳密に定常分布に従う。

◆現在(時刻0)の状態は何か？

- 最近の乱数列から推理する。
- ⇒ 現在の状態を一意に特定する証拠を見つける。

乱数列と対応する推移に依存!!

coalescence

CFIPのアイデア 132

初期状態を知る必要がない!!

4 1 6 3 2 5 → λ

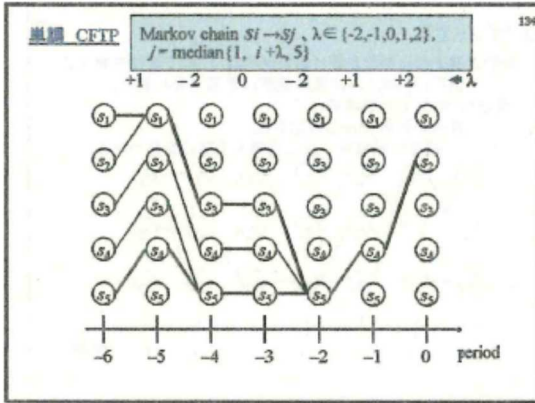
時刻

定義: 単調マルコフ連鎖 (monotone Markov chain)

定義: 単調 (monotone) マルコフ連鎖

- 状態空間中に半順序関係が存在する。
- 最大元と最小元が存在する。
- 更新関数が半順序関係を壊さない。

効率的にcoalescenceを確認



**monotone CFIPアルゴリズムと定理**

マルコフ連鎖  $M$ :  $\left\{ \begin{array}{l} \text{有限離散の状態空間 } \Omega \\ \text{update function } \Phi_\lambda(x, \lambda) \\ \text{エルゴード性を満たす。} \end{array} \right.$

**CFIPアルゴリズム**

1. set  $T = -1$ ; set  $\lambda$ : 乱列;
2. generate  $\lambda[T], \dots, \lambda[7/2 - 1]$ : 乱数; put  $\lambda := (\lambda[T], \dots, \lambda[7/2 - 1], \lambda[7/2], \dots, \lambda[-1])$ ;
3.  $x_0$  と  $x_1$  について、時刻  $T$  から 0 まで  $\lambda$  を用いてマルコフ連鎖  $M$  を推移させる。
  - a. if **coalesce** ( $\exists y \in \Omega, y = \Phi_\lambda(x_0, \lambda) = \Phi_\lambda(x_1, \lambda)$ )  $\Rightarrow$  return  $y$ ;
  - b. otherwise, set  $T := 2T$ ; step 2.に戻る;

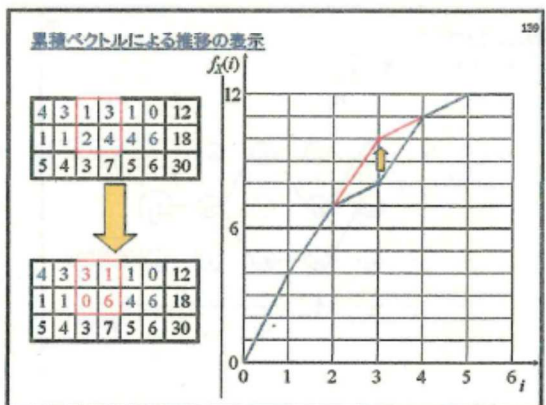
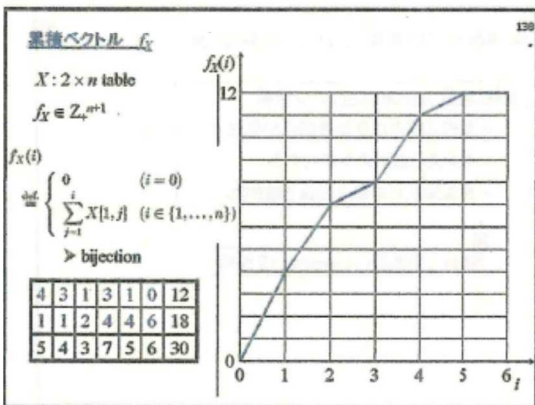
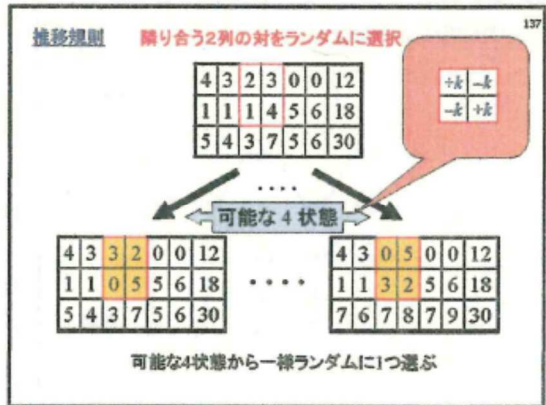
**CFIP定理**  
 上のアルゴリズムが停止して値を返す時、その値はマルコフ連鎖  $M$  の定常分布に数値に従う確率変数を表現する。

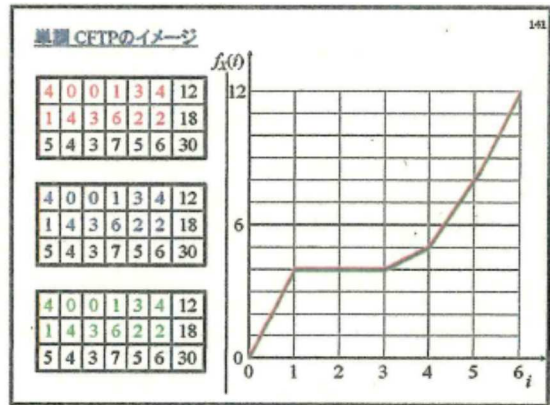
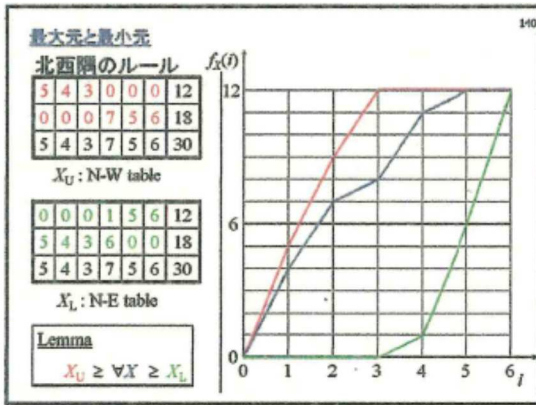
**2行分割表のサンプリング**

- 以下のサンプリングを行う
  - > 行数が2の分割表。
  - > セルは、周辺和を満たす非負整数値。
  - > 一様分布。

						12
						18
5	4	3	7	5	6	30

- 独立性の検定に用いる  $p$  値の計算。
- マルコフ連鎖を用いたサンプリング。





主結果 142

目的: 周辺和を満たす、2行分割表の一様サンプリング

結果: perfect sampling algorithmの提案 (単調CFTP algorithm)

実行する推移回数期待値  $O(n^3 \ln N)$ .

$n$ : 列数,  $N$ : 分割表内の数値の総和.