

## 線形順序付け問題に対するラグランジュ緩和と釘付けテスト

筑波大学 システム情報工学研究科 鮎川 矩義 (Noriyoshi Sukegawa) \*

筑波大学 システム情報工学研究科 張 理遠 (Liyuan Zhang) †

Graduate School of Systems and Information Engineering,  
University of Tsukuba

筑波大学 システム情報系 山本 芳嗣 (Yoshitsugu Yamamoto) ‡

Faculty of Engineering, Information and Systems,  
University of Tsukuba

### 1 はじめに

線形順序付け問題は、有限集合とその集合に属する要素の 1 対比較に対する重みが与えられた下で、重みの総和を最大にする線形順序を求める NP 困難な組合せ最適化問題である。Kemeny の方法、産業連関分析や重み付き完了時刻和最小化 1 機械スケジューリングなどに応用がある。

この問題は 0-1 整数線形計画問題として定式化される。Charon and Hudry [1] は、ラグランジュ緩和を用いた分枝限定法を提案し、乱数に基づく入力例に対し数値実験を行った。ラグランジュ乗数の総数は、素朴に実装すると、問題サイズ  $n$  の 3 乗に比例するため、大規模な問題に対する実装が困難であることを彼らは指摘している。

本論文では、ラグランジュ乗数の取り扱いを改良したラグランジュ緩和法を提案し、その性能を数値実験により検証する。また、問題の規模を縮小するための道具として良く知られた釘付けテストを、線形順序付け問題の構造を用いて強化する方法を提案し、その性能を数値実験により検証する。

本論文の構成は以下のとおりである。2 節で線形順序付け問題を説明し、3 節で 0-1 整数線形計画問題としての定式化を紹介する。4 節で本論文で提案する緩和について説明し、5 節で通常の釘付けテストと本論文で提案する改良釘付けテストについて説明する。6 節で提案する解法について説明をし、7 節でその性能を数値実験により検証する。

### 2 線形順序付け問題

まず線形順序付け問題の応用例である Kemeny の方法を紹介しよう。項目の集合  $N = \{1, 2, \dots, n\}$  が与えられて、 $K$  人の投票者が  $N$  の上の同点を許さない選好順序、(すなわち線形順序) を申告しているとする。彼らの選好順序を、 $\sigma_1, \dots, \sigma_K$  とする。ここでの目標は、この複数の投票者が持つ選好順序を 1 つの選好順序にまとめることである。

Kemeny [4] で提案された通り、与えられたすべての選好順序に「近い」選好順序が最終的な選好順序として選ばれるべきであろう。そこで、ある選好順序が、与えられている複数の選好順序とどの程度近いかを測る

---

\* e-mail: sukega60@sk.tsukuba.ac.jp

† e-mail: zhangliyuan120@gmail.com

‡ e-mail: yamamoto@sk.tsukuba.ac.jp

ための指標を定義する。まず、準備として、定数  $\alpha \in [0, 1]$  に対し、

$$c_{ij}^{(1)} := \alpha |\{ \kappa \mid \sigma_{\kappa}(i) < \sigma_{\kappa}(j) \}| - (1 - \alpha) |\{ \kappa \mid \sigma_{\kappa}(i) > \sigma_{\kappa}(j) \}|$$

$$c_{ij}^{(2)} := \sum_{\kappa=1}^K \alpha [\sigma_{\kappa}(j) - \sigma_{\kappa}(i)]^+ - (1 - \alpha) [\sigma_{\kappa}(i) - \sigma_{\kappa}(j)]^+$$

を定義する。ここで  $|\cdot|$  は対応する集合の位数を表し、 $[t]^+ = \max\{t, 0\}$  とする。定数  $\alpha$  は「勝」と「負」のどちらを重視するかに依存して決まるものである。

全単射  $\pi : N \rightarrow N$  で最終的な選好順序を表す。すなわち、 $i$  が最終的な選好順序において  $j$  より上位であることと  $\pi(i) < \pi(j)$  が成立することが同値であるとする。先ほど定義した係数  $c_{ij}^{(\nu)}$  ( $\nu = 1, 2$ ) は、最終的な選好順序  $\pi$  において  $\pi(i) < \pi(j)$  としたときに、どの程度の同意が得られるかを表している。従って、この値の総和

$$\sum_{(i,j):\pi(i)<\pi(j)} c_{ij}^{(\nu)}$$

を最大にする選好順序は、与えられた複数の選好順序に最も近い選好順序とみなすことができる。Kemeny の方法は、以下の最適化問題

$$(LOP) \quad \left\{ \begin{array}{l} \text{maximize} \quad \sum_{(i,j):\pi(i)<\pi(j)} c_{ij}^{(\nu)} \\ \text{subject to} \quad \pi \text{ は線形順序} \end{array} \right.$$

を解き、その最適解を最終的な選好順序とするものである。この最適化問題 (LOP) が本論文で注目する問題である。

### 3 定式化

この節では、線形順序付け問題の 0-1 整数線形計画問題としての一般的な定式化をまず紹介し、次に、メモリの節約を考慮した定式化を紹介する。

#### 3.1 0-1 整数線形計画問題

与えられた線形順序  $\pi$  に対し、すべての順序対  $(i, j)$  について、0-1 変数  $x_{ij}$  を以下のように定義する。

$$x_{ij} = \begin{cases} 1 & \text{if } \pi(i) < \pi(j) \\ 0 & \text{otherwise} \end{cases}$$

すると、線形順序付け問題は以下のように定式化される。

$$(LOP) \quad \left\{ \begin{array}{l} \text{maximize} \quad \sum_{(i,j) \in N_{\neq}^2} c_{ij} x_{ij} \\ \text{subject to} \quad x_{ij} \in \{0, 1\} \quad \text{for all } (i, j) \in N_{\neq}^2 \quad (0-1) \\ x_{ij} + x_{ji} = 1 \quad \text{for all } (i, j) \in N_{\neq}^2 \quad (\text{反対称性}) \\ x_{ij} + x_{jk} + x_{ki} \leq 2 \quad \text{for all } (i, j, k) \in N_{\neq}^3 \quad (\text{推移性}) \end{array} \right.$$

ここで、

$$N_{\neq}^2 := \{(i, j) \mid i, j \in N, i \neq j\}$$

$$N_{\neq}^3 := \{(i, j, k) \mid i, j, k \in N, i \neq j, j \neq k, k \neq i\}$$

である。最適化問題 (LOP) は  $n(n-1)/2$  個の等式制約 (反対称性制約) と  $n(n-1)(n-2)/3$  個の不等式制約 (推移性制約) を持つため、問題サイズ  $n$  の増加に伴い、非常にたくさんの制約を持つようになることが分かる。従ってこれらの制約をどのように扱うかが、解法を考える上で重要になる。

### 3.2 変数削減

反対称性制約を用いて  $x_{ji}$  で  $i < j$  を満たすものを  $1 - x_{ij}$  で書き換えることで、以下の等価な最適化問題 (P) が得られる。

$$(P) \quad \begin{cases} \text{maximize} & \sum_{(i,j) \in N_{<}^2} \bar{c}_{ij} x_{ij} \\ \text{subject to} & x_{ij} \in \{0, 1\} & \text{for all } (i, j) \in N_{<}^2 \\ & x_{ij} + x_{jk} - x_{ik} \leq 1 & \text{for all } (i, j, k) \in N_{<}^3 \quad (\text{型 1}) \\ & -x_{ij} - x_{jk} + x_{ik} \leq 0 & \text{for all } (i, j, k) \in N_{<}^3 \quad (\text{型 2}) \end{cases}$$

ここで、

$$\begin{aligned} \bar{c}_{ij} &:= c_{ij} - c_{ji} \\ N_{<}^2 &:= \{(i, j) \mid i, j \in N, i < j\} \\ N_{<}^3 &:= \{(i, j, k) \mid i, j, k \in N, i < j < k\} \end{aligned}$$

である。決定変数が半減し、等式制約が消え、推移性制約が 2 種類に分かれている。本論文では、1 つ目の推移性制約を型 1、2 つ目の推移性制約を型 2 と呼ぶ。また、この書き換えによって目的関数に定数項  $\sum_{(i,j) \in N_{<}^2} c_{ji}$  が表れるが、これは省略する。すなわち、最適化問題の最適値を  $\omega(\cdot)$  で表すとすれば、 $\omega(P) + \sum_{(i,j) \in N_{<}^2} c_{ji} = \omega(\text{LOP})$  である。以降では、最適化問題 (P) に対する解法について議論する。

## 4 緩和

この節では、まず前節の最適化問題 (P) に対するいくつかの緩和を提案し、それらの性質についてまとめる。次に本論文で提案する緩和法について説明をする。

### 4.1 推移性制約の緩和

まず、最適化問題 (P) において、問題サイズ  $n$  の増加に伴い急速に増加する推移性制約を緩和する。すなわち  $U$  と  $V$  をそれぞれ  $N_{<}^3$  の適当な部分集合として、以下の制約緩和問題 (P(U, V)) を解く。

$$(P(U, V)) \quad \begin{cases} \text{maximize} & \sum_{(i,j) \in N_{<}^2} \bar{c}_{ij} x_{ij} \\ \text{subject to} & x_{ij} \in \{0, 1\} & \text{for all } (i, j) \in N_{<}^2 \\ & x_{ij} + x_{jk} - x_{ik} \leq 1 & \text{for all } (i, j, k) \in U \\ & -x_{ij} - x_{jk} + x_{ik} \leq 0 & \text{for all } (i, j, k) \in V \end{cases}$$

明らかに、制約緩和問題 (P(U, V)) を解いた結果得られる最適解が、偶然すべての推移性制約を満たしている場合、その解は元問題 (P) の最適解である。

## 4.2 ラグランジュ緩和

制約緩和問題 ( $P(U, V)$ ) は, ( $U, V$ ) に特殊な構造が仮定されない限り, 一般には依然として解きにくい問題である. そこで, 更にこの問題を緩和した問題を解くということが考えられる. 本論文では, ラグランジュ緩和を用いる.

型 1 に属するすべての制約に対し非負の乗数  $u_{ijk}$  を, 型 2 に属するすべての制約に対し非負の乗数  $v_{ijk}$  を導入し, 以下のような制約が 0-1 制約のみのラグランジュ緩和問題 ( $LR(\mathbf{u}, \mathbf{v})$ ) を解く.

$$(LR(\mathbf{u}, \mathbf{v})) \quad \begin{cases} \text{maximize} & \sum_{(i,j) \in N_{\leq}^2} \bar{c}_{ij} x_{ij} + \sum_{(i,j,k) \in U} u_{ijk} (1 - x_{ij} - x_{jk} + x_{ik}) \\ & + \sum_{(i,j,k) \in V} v_{ijk} (0 + x_{ij} + x_{jk} - x_{ik}) \\ \text{subject to} & x_{ij} \in \{0, 1\} \quad \text{for all } (i, j) \in N_{\leq}^2 \end{cases}$$

簡単のため, どのような  $U$  と  $V$  を選んでいるかは明記せずに, この最適化問題を単純に ( $LR(\mathbf{u}, \mathbf{v})$ ) と書く. ここで,  $\mathbf{u}$  と  $\mathbf{v}$  はそれぞれ乗数のベクトル  $(u_{ijk})_{(i,j,k) \in U}$  と  $(v_{ijk})_{(i,j,k) \in V}$  を表しており, ラグランジュ乗数ベクトルと呼ばれる. また,  $r(\mathbf{u}, \mathbf{v})_{ij}$  を目的関数における  $x_{ij}$  の係数とする. ラグランジュ緩和問題 ( $LR(\mathbf{u}, \mathbf{v})$ ) は, 0-1 制約のみの問題であるため, 最適解の 1 つ  $\mathbf{x}(\mathbf{u}, \mathbf{v}) = (x(\mathbf{u}, \mathbf{v})_{ij})_{(i,j) \in N_{\leq}^2}$  は, 以下のように計算できる.

$$x(\mathbf{u}, \mathbf{v})_{ij} = \begin{cases} 1 & \text{if } r(\mathbf{u}, \mathbf{v})_{ij} > 0 \\ 0 & \text{if } r(\mathbf{u}, \mathbf{v})_{ij} \leq 0 \end{cases} \quad (1)$$

ラグランジュ緩和問題 ( $LR(\mathbf{u}, \mathbf{v})$ ) の最適値  $\omega(LR(\mathbf{u}, \mathbf{v}))$  は元問題 ( $P$ ) の最適値の上界を与える.

重要な点は, 通常のラグランジュ緩和で取り扱うべき乗数の総数が  $n(n-1)(n-2)/3$  であったのに対し, 本研究で提案するラグランジュ緩和では  $|U| + |V|$  だけである点である. もし, うまく ( $U, V$ ) を選んで, すべての乗数を考慮した場合と同程度の上界を得ることができれば, メモリの節約と計算時間の短縮につながる.

ラグランジュ緩和問題 ( $LR(\mathbf{u}, \mathbf{v})$ ) の最適解がすべての推移性制約を満たしていたとしても, 制約緩和問題 ( $P(U, V)$ ) とは異なり, 元問題 ( $P$ ) の最適解であることは必ずしも保証できない [3].

## 5 釘付けテスト

この節では, 問題規模の縮小のために用いる釘付けテストを紹介する. まず, 通常の釘付けテストを説明し, 次に本論文で提案する釘付けテストを紹介する.

### 5.1 通常の釘付けテスト

ラグランジュ緩和問題 ( $LR(\mathbf{u}, \mathbf{v})$ ) の最適解  $\mathbf{x}(\mathbf{u}, \mathbf{v})$  から得られる情報を用いて, ある変数が元問題 ( $P$ ) の最適解において 0 を取る変数なのかそれとも 1 を取る変数なのかを判別できる.

ある添え字  $(s, t) \in N_{\leq}^2$  について元問題 ( $P$ ) が  $x_{st} = \xi \in \{0, 1\}$  となる最適解を持っていたとする. このとき, 明らかに, 元問題 ( $P$ ) と  $x_{st} = \xi$  なる制約を ( $P$ ) に追加した問題 ( $P|x_{st} = \xi$ ) は, それぞれの最適値が一致するという意味で, 等価である. 元問題 ( $P$ ) のある下界を  $\omega_{\text{low}}$  とすると,  $\omega(P|x_{st} = \xi) = \omega(P) \geq \omega_{\text{low}}$  が成立する. 制約緩和問題 ( $P(U, V)$ ) は元問題 ( $P$ ) の緩和問題であり, ラグランジュ緩和問題 ( $LR(\mathbf{u}, \mathbf{v})$ ) は制

約緩和問題を更に緩和した問題であるので,

$$\omega(LR(\mathbf{u}, \mathbf{v})|x_{st} = \xi) \geq \omega(P(U, V)|x_{st} = \xi) \geq \omega(P|x_{st} = \xi),$$

が成立する. よって,  $\omega(LR(\mathbf{u}, \mathbf{v})|x_{st} = \xi) \geq \omega_{\text{low}}$ . が得られる. 以上をまとめると以下の補題が得られる.

**補題 1.** ある定数  $\xi \in \{0, 1\}$  に対し, もし  $\omega(LR(\mathbf{u}, \mathbf{v})|x_{st} = \xi) < \omega_{\text{low}}$  が成立するならば, 元問題 (P) の任意の最適解において,  $x_{st} = 1 - \xi$  が成立する.

ラグランジュ緩和問題 ( $LR(\mathbf{u}, \mathbf{v})$ ) を解いて得られる最適解  $\mathbf{x}(\mathbf{u}, \mathbf{v})$  において  $x(\mathbf{u}, \mathbf{v})_{st} = 0$  が成立したと仮定する. このとき, 簡単な計算から以下の関係式が成立することが分かる.

$$\omega(LR(\mathbf{u}, \mathbf{v})|x_{st} = 1) = \omega(LR(\mathbf{u}, \mathbf{v})) + r(\mathbf{u}, \mathbf{v})_{st}.$$

最適解  $\mathbf{x}(\mathbf{u}, \mathbf{v})$  において  $x(\mathbf{u}, \mathbf{v})_{st}$  が 0 であるので, 対応する係数  $r(\mathbf{u}, \mathbf{v})_{st}$  は非正である. 同様にして,  $x(\mathbf{u}, \mathbf{v})_{st}$  が 1 の場合には以下が成立する.

$$\omega(LR(\mathbf{u}, \mathbf{v})|x_{st} = 0) = \omega(LR(\mathbf{u}, \mathbf{v})) - r(\mathbf{u}, \mathbf{v})_{st}$$

このとき,  $r(\mathbf{u}, \mathbf{v})_{st}$  は正である. 以上をまとめると以下の定理が得られる.

**定理 2.** もし,  $\omega(LR(\mathbf{u}, \mathbf{v})) - \omega_{\text{low}} < |r(\mathbf{u}, \mathbf{v})_{st}|$  が成立するならば, 元問題 (P) の任意の最適解  $\mathbf{x}^*$  において,  $x_{st}^* = x(\mathbf{u}, \mathbf{v})_{st}$  が成立する.

定理の条件が成立するとき, 決定変数  $x_{st}$  は  $x(\mathbf{u}, \mathbf{v})_{st}$  に固定されたという.

## 5.2 改良釘付けテスト

いくつかの決定変数が 0 または 1 に固定されている状況を考える. 集合  $P_0$  と  $P_1$  を, それぞれ前節の釘付けテストにより 0 と 1 に固定された決定変数に対応する添え字の集合とする. 本論文では, これらの固定された変数の情報と, 元問題 (P) が持つ推移性制約を組み合わせて, 通常の釘付けテストを強化する方法を提案する.

釘付けテストによって固定された変数から次のように有向グラフを定義する. 頂点集合を  $N$  とし, 有向枝集合  $A(P_0, P_1)$  を,  $x_{ij}$  が 1 に, もしくは  $x_{ji}$  が 0 に固定された順序対  $(i, j)$  の集合とする. すなわち,  $A(P_0, P_1) := \{(i, j) \in N^2 \mid (j, i) \in P_0 \text{ または } (i, j) \in P_1\}$  とする.

**定義 3.**  $P_0$  と  $P_1$  が与えられて,  $N$  の要素  $i$  と  $j$  について, 有向枝集合  $A(P_0, P_1)$  上に  $i$  から  $j$  への有向道があるとき,  $i$  は  $j$  の祖先, または  $j$  は  $i$  の子孫であるという.

**定義 4.** ある決定変数の添え字  $(s, t) \in N^2 \setminus (P_0 \cup P_1)$  が与えられたとき, 以下を定義する.

$$\begin{aligned} S_1 &:= \{s\} \cup \{i \in N \mid i \text{ は } s \text{ の祖先}\}, & T_1 &:= \{t\} \cup \{j \in N \mid j \text{ は } t \text{ の子孫}\} \\ S_0 &:= \{s\} \cup \{i \in N \mid i \text{ は } s \text{ の子孫}\}, & T_0 &:= \{t\} \cup \{j \in N \mid j \text{ は } t \text{ の祖先}\} \end{aligned}$$

まだ固定されていない決定変数  $x_{st}$  を選んで, 一時的に 1 に固定してみる. すると, 任意の  $s$  の祖先は任意の  $t$  の子孫にとっての祖先となるので, 推移性制約から決定変数は以下を満たすことになる.

$$x_{ij} = \begin{cases} 1 & \text{for all } (i, j) \in (S_1 \times T_1) \cap N^2 \\ 0 & \text{for all } (i, j) \in (T_1 \times S_1) \cap N^2 \end{cases}$$

決定変数  $x_{st}$  を一時的に 0 に固定した場合は以下ようになる。

$$x_{ij} = \begin{cases} 1 & \text{for all } (i, j) \in (T_0 \times S_0) \cap N_{<}^2 \\ 0 & \text{for all } (i, j) \in (S_0 \times T_0) \cap N_{<}^2 \end{cases}$$

与えられたラグランジュ乗数ベクトル  $\mathbf{u}$  と  $\mathbf{v}$  に対し,  $x_{st} = 1$  を追加した以下のラグランジュ緩和問題を定義する。

$$(LR(\mathbf{u}, \mathbf{v}, P_0, P_1) \mid x_{st} = 1) \left\{ \begin{array}{l} \text{maximize} \quad \sum_{(i,j) \in N_{<}^2} \bar{c}_{ij} x_{ij} + \sum_{(i,j,k) \in U} u_{ijk}(1 - x_{ij} - x_{jk} + x_{ik}) \\ \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad + \sum_{(i,j,k) \in V} v_{ijk}(0 + x_{ij} + x_{jk} - x_{ik}) \\ \text{subject to} \quad x_{ij} \in \{0, 1\} \quad \text{for all } (i, j) \in N_{<}^2 \\ \qquad \qquad \qquad \quad \quad \quad \quad \quad \quad \quad \quad \quad x_{ij} = \begin{cases} 0 & \text{for all } (i, j) \in (T_1 \times S_1) \cap N_{<}^2 \cup P_0 \\ 1 & \text{for all } (i, j) \in (S_1 \times T_1) \cap N_{<}^2 \cup P_1 \end{cases} \end{array} \right.$$

この問題は,  $x_{st} = 1$  という制約を追加した元問題 (P) の子問題の緩和問題になっている。

**補題 5.** もし  $((S_1 \times T_1) \cap N_{<}^2 \cap P_0) \cup ((T_1 \times S_1) \cap N_{<}^2 \cap P_1) \neq \emptyset$ , ならば, 元問題 (P) の任意の最適解  $\mathbf{x}^*$  について  $x_{st}^* = 0$  が成立する。

**証明.** ある添え字  $(i, j)$  が集合  $((S_1 \times T_1) \cap N_{<}^2 \cap P_0) \cup ((T_1 \times S_1) \cap N_{<}^2 \cap P_1)$  に属したと仮定すると, 決定変数  $x_{ij}$  はラグランジュ緩和問題  $(LR(\mathbf{u}, \mathbf{v}, P_0, P_1) \mid x_{st} = 1)$  において同時に 0 と 1 になる必要がある。よってこの問題は実行不可能になり,  $x_{st} = 1$  となる元問題 (P) の最適解は存在しないことが分かる。 □

決定変数  $x_{st}$  を一時的に 0 に固定した場合には, 以下のラグランジュ緩和問題と補題が得られる。

$$(LR(\mathbf{u}, \mathbf{v}, P_0, P_1) \mid x_{st} = 0) \left\{ \begin{array}{l} \text{maximize} \quad \sum_{(i,j) \in N_{<}^2} \bar{c}_{ij} x_{ij} + \sum_{(i,j,k) \in U} u_{ijk}(1 - x_{ij} - x_{jk} + x_{ik}) \\ \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad + \sum_{(i,j,k) \in V} v_{ijk}(0 + x_{ij} + x_{jk} - x_{ik}) \\ \text{subject to} \quad x_{ij} \in \{0, 1\} \quad \text{for all } (i, j) \in N_{<}^2 \\ \qquad \qquad \qquad \quad \quad \quad \quad \quad \quad \quad \quad \quad x_{ij} = \begin{cases} 0 & \text{for all } (i, j) \in (S_0 \times T_0) \cap N_{<}^2 \cup P_0 \\ 1 & \text{for all } (i, j) \in (T_0 \times S_0) \cap N_{<}^2 \cup P_1 \end{cases} \end{array} \right.$$

**補題 6.** もし  $((T_0 \times S_0) \cap N_{<}^2 \cap P_0) \cup ((S_0 \times T_0) \cap N_{<}^2 \cap P_1) \neq \emptyset$ , ならば, 元問題 (P) の任意の最適解  $\mathbf{x}^*$  について  $x_{st}^* = 1$  が成立する。

ラグランジュ緩和問題  $(LR(\mathbf{u}, \mathbf{v}, P_0, P_1) \mid x_{st} = \xi)$  は, 元問題 (P) に  $x_{st} = \xi$  なる制約を追加した子問題の緩和問題であるので, 以下の補題が得られる。

**補題 7.** ある定数  $\xi \in \{0, 1\}$  と固定されていない決定変数  $x_{st}$  が与えられて, もし  $\omega(LR(\mathbf{u}, \mathbf{v}, P_0, P_1) \mid x_{st} = \xi) < \omega_{\text{low}}$  が成立するならば, 元問題 (P) の任意の最適解において  $x_{st} = 1 - \xi$  が成立する。

先ほどと同様にして, 補題 7 の  $\omega(LR(\mathbf{u}, \mathbf{v}, P_0, P_1) \mid x_{st} = \xi)$  は,  $\omega(LR(\mathbf{u}, \mathbf{v}))$  とラグランジュ緩和問題の目的関数の係数を用いて次のように計算できる

$$\omega(LR(\mathbf{u}, \mathbf{v})) - \omega(LR(\mathbf{u}, \mathbf{v}, P_0, P_1) \mid x_{st} = 1 - x(\mathbf{u}, \mathbf{v})_{st}) = \sum_{(i,j) \in A'} |r(\mathbf{u}, \mathbf{v})_{ij}|$$

ここで,  $x(\mathbf{u}, \mathbf{v})_{st} = 0$  の場合に  $A'$  は,

$$A' = (((S_1 \times T_1) \cap N_{<}^2 \cup P_1) \cap \{(i, j) \mid x(\mathbf{u}, \mathbf{v})_{ij} = 0\}) \cup (((T_1 \times S_1) \cap N_{<}^2 \cup P_0) \cap \{(i, j) \mid x(\mathbf{u}, \mathbf{v})_{ij} = 1\})$$

で与えられ,  $x(\mathbf{u}, \mathbf{v})_{st} = 1$  の場合に  $A'$  は,

$$A' = (((T_0 \times S_0) \cap N_{\geq}^2 \cup P_1) \cap \{(i, j) \mid x(\mathbf{u}, \mathbf{v})_{ij} = 0\}) \cup (((S_0 \times T_0) \cap N_{\geq}^2 \cup P_0) \cap \{(i, j) \mid x(\mathbf{u}, \mathbf{v})_{ij} = 1\})$$

である. 最初の部分集合は, ラグランジュ緩和問題  $(LR(\mathbf{u}, \mathbf{v}, P_0, P_1) \mid x_{st} = \xi)$  において 1 を取るべきであるのに,  $\mathbf{x}(\mathbf{u}, \mathbf{v})$  において 0 を取っている決定変数に対応する添え字の集合であり, 2 番目の部分集合は, ラグランジュ緩和問題  $(LR(\mathbf{u}, \mathbf{v}, P_0, P_1) \mid x_{st} = \xi)$  において 0 を取るべきであるのに,  $\mathbf{x}(\mathbf{u}, \mathbf{v})$  において 1 を取っている決定変数に対応する添え字の集合である. 補題 7 とそれに続く  $\omega(LR(\mathbf{u}, \mathbf{v}, P_0, P_1) \mid x_{st} = \xi)$  の計算方法を改良釘付けテストと呼ぶ.

釘付けテストによっていくつかの変数が固定されてるとき, 推移性制約を用いて, 更に多くの変数を固定できる. これは, 頂点集合を  $N$ , 枝集合を  $A(P_0, P_1)$  とする有向グラフの推移的閉包を作ることで実現できる. 本論文では, 改良釘付けテストの前処理として Warshall [6] の方法によりグラフの推移的閉包を作る.

## 6 提案解法

この節では, 本論文で提案する解法について説明する. 添え字集合  $U$  と  $V$  を固定した下で, ラグランジュ乗数ベクトルの最適化を行う劣勾配法についてまず説明し, 次に,  $U$  と  $V$  の更新方法を説明し, 最後にそれらを統合して提案解法の全体像について説明する.

### 6.1 ラグランジュ双対問題と劣勾配法

ここでは, 固定された変数の添え字集合  $P_0, P_1$  を省略して, ラグランジュ緩和問題  $(LR(\mathbf{u}, \mathbf{v}, P_0, P_1))$  の最適値  $\omega(LR(\mathbf{u}, \mathbf{v}, P_0, P_1))$  を  $\omega(\mathbf{u}, \mathbf{v})$  で表す. ラグランジュ双対問題  $(LD)$  は非負のラグランジュ乗数ベクトル  $(\mathbf{u}, \mathbf{v})$  で対応するラグランジュ緩和問題の目的関数値  $\omega(\mathbf{u}, \mathbf{v})$  を最小化する問題である. すなわち,

$$(LD) \quad \begin{cases} \text{minimize} & \omega(\mathbf{u}, \mathbf{v}) \\ \text{subject to} & \mathbf{u}, \mathbf{v} \geq \mathbf{0} \end{cases}$$

である. 目的関数  $\omega(\mathbf{u}, \mathbf{v})$  は区分解線形な凸関数であり, 微分不可能な点を含む. この問題に対する最も良く知られた解法の 1 つに劣勾配法がある [2].

**定義 8.**  $(\mathbf{g}^u, \mathbf{g}^v)$  が関数  $\omega$  の  $(\bar{\mathbf{u}}, \bar{\mathbf{v}}) \geq \mathbf{0}$  における劣勾配であるとは, 任意の非負ベクトル  $(\mathbf{u}, \mathbf{v})$  について  $\omega(\bar{\mathbf{u}}, \bar{\mathbf{v}}) + \langle \mathbf{g}^u, \mathbf{u} - \bar{\mathbf{u}} \rangle + \langle \mathbf{g}^v, \mathbf{v} - \bar{\mathbf{v}} \rangle \leq \omega(\mathbf{u}, \mathbf{v})$  が成立することをいう. ここで  $\langle \cdot, \cdot \rangle$  は内積を表す.

劣勾配の 1 つは以下のように, ラグランジュ緩和問題の最適解から簡単に計算できる.

**補題 9.** ラグランジュ緩和問題  $(LR(\mathbf{u}, \mathbf{v}, P_0, P_1))$  の最適解を  $\mathbf{x}(\mathbf{u}, \mathbf{v})$  とする. このとき以下で定義されるベクトル  $(\mathbf{g}^u, \mathbf{g}^v)$  は関数  $\omega$  の  $(\mathbf{u}, \mathbf{v})$  における劣勾配となる.

$$\begin{aligned} g_{ijk}^u &:= 1 - x(\mathbf{u}, \mathbf{v})_{ij} - x(\mathbf{u}, \mathbf{v})_{jk} + x(\mathbf{u}, \mathbf{v})_{ik} & \text{for } (i, j, k) \in U \\ g_{ijk}^v &:= 0 + x(\mathbf{u}, \mathbf{v})_{ij} + x(\mathbf{u}, \mathbf{v})_{jk} - x(\mathbf{u}, \mathbf{v})_{ik} & \text{for } (i, j, k) \in V \end{aligned}$$

本論文では以下のルールに従って手元にある乗数ベクトル  $(\mathbf{u}, \mathbf{v})$  から次の乗数ベクトル  $(\mathbf{u}^+, \mathbf{v}^+)$  を計算

する.

$$u_{ijk}^+ := \max \left\{ 0, u_{ijk} - \mu \frac{\omega(\mathbf{u}, \mathbf{v}) - \omega_{\text{low}}}{\|(\mathbf{g}^u, \mathbf{g}^v)\|^2} g_{ijk}^u \right\} \quad \text{for } (i, j, k) \in U \quad (2)$$

$$v_{ijk}^+ := \max \left\{ 0, v_{ijk} - \mu \frac{\omega(\mathbf{u}, \mathbf{v}) - \omega_{\text{low}}}{\|(\mathbf{g}^u, \mathbf{g}^v)\|^2} g_{ijk}^v \right\} \quad \text{for } (i, j, k) \in V \quad (3)$$

ここで  $\mu$  はステップサイズを調整するパラメータであり初期値を 2 とする. また  $\|\cdot\|$  はユークリッドノルムを表す. 上述の更新ルールにおける  $\omega_{\text{low}}$  を元問題の最適値  $\omega(P)$  で置き換えると, 生成される乗数ベクトルの列がラグランジュ双対問題 ( $LD$ ) の最適解に収束することが知られている. 目的関数値  $\omega(\mathbf{u}, \mathbf{v})$  は乗数ベクトルが更新されたときに必ずしも減少しない. そこで,  $\mu$  の初期値を 2 とし, 上界の改善に連続して失敗した回数を数え, 5 回に達したならば  $\mu$  を半分にする. そして,  $\mu$  が 0.005 よりも小さくなったならば終了する.

これにより,  $U$  と  $V$  を固定した下での乗数の最適化が行えるわけだが,  $U$  と  $V$  の更新をどのように行うかの問題が残っている.

## 6.2 $U$ と $V$ の更新

固定された  $U$  と  $V$  の下で, 劣勾配法におけるステップサイズを調節するパラメータ  $\mu$  が 0.005 を下回ったとき, 今の  $U$  と  $V$  の下では上界を今以上に改善できる可能性はないと考え,  $U$  と  $V$  を更に拡大する方針を取る. 具体的には, 劣勾配法が終了した時点で手元にあるラグランジュ緩和問題 ( $LR(\mathbf{u}, \mathbf{v}, P_0, P_1)$ ) の最適解  $\mathbf{x}(\mathbf{u}, \mathbf{v})$  が壊している推移性制約を探して, それらをすべて追加する.

## 6.3 提案解法

提案する解法は, 劣勾配法と釘付けテストを行う内側のループ (Step 2 から 7) と,  $U$  と  $V$  を更新する外側のループから構成される.

### Step 1 (初期化)

- (a) 入力である重み行列から各項目  $i$  について  $\sum_{j \in N} c_{ij} - \sum_{j \in N} c_{ji}$  を計算し, この値で降順に並べた線形順序を最初の実行可能解とする. また, その目的関数値を  $\omega_{\text{low}}$  とする.
- (b) 上で計算した線形順序で隣合うすべての 3 つ組  $(i, j, k)$  に関する推移性制約を初期の  $(U, V)$  とし,  $l \leftarrow 0, \mu \leftarrow 2.0, k \leftarrow 0, (\mathbf{u}, \mathbf{v}) \leftarrow (\mathbf{0}, \mathbf{0}), P_0, P_1 \leftarrow \emptyset, \omega_{\text{up}} \leftarrow +\infty$  とする.

### Step 2 (ラグランジュ緩和問題 ( $LR(\mathbf{u}, \mathbf{v}, P_0, P_1)$ ) を解く)

- (a) 目的関数の係数  $r(\mathbf{u}, \mathbf{v})_{ij}$  を計算し,  $\mathbf{x}(\mathbf{u}, \mathbf{v})_{ij}$  を (1) に従い計算する.
- (b)  $\omega_{\text{up}} \leftarrow \min\{\omega_{\text{up}}, \omega(LR(\mathbf{u}, \mathbf{v}, P_0, P_1))\}$  とし, もし  $\omega_{\text{up}}$  が改善されたならば,  $l \leftarrow l+1, k \leftarrow k+1$ . それ以外のとき,  $l, k \leftarrow 0$  とする.

### Step 3 (終了条件)

- (a)  $\omega_{\text{up}} - \omega_{\text{low}} < \varepsilon$  が成立するか,  $k \geq 500$  が成立すれば停止.

### Step 4 ( $\mathbf{x}(\mathbf{u}, \mathbf{v})$ の実行可能解への修正と近傍探索)

- (a) 各項目  $i$  について  $\sum_{i < j} \mathbf{x}(\mathbf{u}, \mathbf{v})_{ij} + \sum_{i > j} (1 - \mathbf{x}(\mathbf{u}, \mathbf{v})_{ji})$  を計算し, この値で降順に並べた線形順序を近傍探索の初期点とする.
- (b) 各要素について順位を  $\beta$  の範囲で上下させて目的関数値が増加する場所を探し, 見つければ解を更新する. 局所最適解に至ったとき,  $\omega_{\text{up}}$  よりも良い下界が見つかったら  $\omega_{\text{up}}$  を更新する.

### Step 5 (釘付けテスト)



- (a)  $(\omega_{\text{up}} - \omega_{\text{low}})/\omega_{\text{low}} < \eta$  かつ  $k$  が 500 の倍数のとき,
- (b)  $P_0 = P_1 = \emptyset$  のときは通常の釘付けテストを, それ以外のとき, 改良釘付けテストを行う.
- (c)  $P_0$  と  $P_1$  にそれぞれ 0 と 1 に固定された決定変数に対応する添え字集合を入れる.

Step 6 ( $\mu$  の更新)

- (a)  $\mu \leq 0.005$  が成立していれば,  $\mu \leftarrow 2.0$  とし Step 8 に行く.
- (b) もし  $l$  が 5 であれば,  $\mu \leftarrow \mu/2$ .

Step 7 (ラグランジュ乗数ベクトル  $(\mathbf{u}, \mathbf{v})$  の更新)

- (a)  $(\mathbf{u}, \mathbf{v})$  を (2) と (3) に従って更新.  $k \leftarrow k + 1$  とし, Step 2 に行く.

Step 8 (推移性制約の添え字集合  $U, V$  の更新)

- (a)  $\mathbf{x}(\mathbf{u}, \mathbf{v})$  によって壊されている推移性制約を探し, すべて  $U$  と  $V$  に追加する.
- (b) 新たに追加された添え字  $(i, j, k)$  について  $u_{ijk}, v_{ijk} \leftarrow 0$  とし, Step 2 に行く.

## 7 数値実験

この節では, 提案する改良釘付けテストの性能を数値実験により評価する. 解法は Java により実装し, 実験は CPU : Intel i3, 3.33 GHz, メモリ 2 GB の計算機上で行った. 以下の数値実験において,

- $\text{GAP} := \omega_{\text{up}} - \omega_{\text{low}}$ ,  $\text{RGAP} := (\omega_{\text{up}} - \omega_{\text{low}})/\omega_{\text{low}}$ ,
- $\%^{UV} := 100(|U| + |V|)/|N_{\leq}^3|$ ,  $\%^{PT} := 100(|P_0| + |P_1|)/|N_{\leq}^2|$ ,
- $\%^{LP} := 100(\omega_{\text{up}} - \omega_{\text{up}}^{LP})/\omega_{\text{up}}^{LP}$ ,  $\%^* := 100\omega_{\text{up}}/\omega^*$

である. ここで,  $\omega^*$  と  $\omega_{\text{up}}^{LP}$  はそれぞれ (LOP) の最適値と (LOP) の線形緩和問題の最適値を表す. また,  $\omega_{\text{up}}^{LP}$  は  $(U, V)$  としてすべての推移性制約を扱ったときのラグランジュ双対問題の最適値と一致する [3]. 以下の実験結果では,  $\omega^*$  と  $\omega_{\text{up}}^{LP}$  を Xpress Optimizer 21.01.06 (PC: Intel i7, 2.80GHz, 6 GB memory) により計算している. 乱数による問題例と実社会から得られた問題例についての実験結果を示す. どの問題例も目的関数の係数ベクトルが整数であるため, 上界値  $\omega_{\text{up}}$  はすべて整数値をとる. したがって提案解法の Step 3 の終了条件で  $\varepsilon := 1$  とした. 他のパラメータについては,  $(\beta, \eta, M) = (5, 0.01, 100,000)$  としている.

### 7.1 乱数による問題例

本論文では, 乱数による問題例として Chron and Hudry [1] によって解かれた Median39 と Judges100 を用いた. 以下の実験例では, それぞれ 10 個ずつ問題例を生成している.

問題例 Median39 は,  $c_{ij}$  と  $c_{ji}$  のどちらか一方は 0 で, もう一方が正となるように構成するものである. ここで, 正になる確率と 0 になる確率は等しく 0.5 で, 正になる場合には  $c_{ij}$  あるいは  $c_{ji}$  は, 1 から 10 の正整数のなかから一様にランダムに選ぶ. 実験結果を表 7.1 に示す.

問題例 Judges100 は, 審査員 100 人と候補者 50 人がいる状況を考え, 各審査員が候補者  $i$  を候補者  $j$  よりも好む確率を  $0.5 + 0.35(j - i)/(n - 1)$  とし, 係数  $c_{ij}$  を候補者  $i$  を候補者  $j$  より好む審査員の人数から候補者  $j$  を候補者  $i$  より好む審査員の人数を引いた人数とするものである. 実験結果を表 7.1 に示す.

列  $\%^{LP}$  から, 問題例 Median39 と Judges100 のどちらについても, 線形緩和問題で得られる上界とほぼ同等の上界が提案解法によって得られている. また, 列  $\%^*$  から, 短時間で非常に良い下界が見つかったことが分かる. しかしながら, 1 つの問題例 (Judges100d) を除いて, 双対ギャップ (GAP) が残ったために最

適性は保証できなかった。

表 1 Median39 の結果

問題例	提案解法					Xpress		
	GAP	RGAP	% $LP$	% $PT$	時間 (秒)	$\omega^*$	時間 (秒)	%*
Median39a	70.00	2.4e-2	1.7e-1	0.00	7.52	2898.00	257.40	99.34
Median39b	68.00	2.4e-2	1.7e-1	0.00	7.38	2889.00	397.60	99.72
Median39c	44.00	1.5e-2	8.0e-2	0.00	7.46	2966.00	169.80	99.87
Median39d	54.00	1.8e-2	1.1e-1	0.00	6.54	2958.00	245.00	100.00
Median39e	51.00	1.8e-2	8.9e-2	0.00	7.86	2868.00	243.00	100.00
Median39f	83.00	2.9e-2	4.2e-2	0.00	8.60	2905.00	404.00	99.72
Median39g	80.00	2.8e-2	8.2e-2	0.00	7.50	2900.00	1282.70	99.93
Median39h	94.00	3.3e-2	1.1e-1	0.00	8.27	2878.00	2784.70	99.83
Median39i	71.00	2.5e-2	9.0e-2	0.00	7.41	2902.00	489.20	99.90
Median39j	63.00	2.2e-2	1.1e-1	0.00	7.05	2860.00	397.90	99.90

表 2 Judges100 の結果

問題例	提案解法					Xpress		
	GAP	RGAP	% $LP$	% $PT$	時間 (秒)	$\omega^*$	時間 (秒)	%*
Judges100a	25.00	4.1e-4	3.8e-3	58.71	12.98	60834.00	118.00	99.99
Judges100b	21.00	3.4e-4	4.9e-3	65.45	12.26	61752.00	102.10	99.99
Judges100c	30.00	4.9e-4	6.4e-3	52.26	13.43	60598.00	111.10	99.97
Judges100d	0.00	0.0	0.0	91.45	9.81	61592.00	85.70	100.00
Judges100e	43.00	7.0e-4	9.4e-3	35.84	12.93	61120.00	113.70	99.95
Judges100f	20.00	3.3e-4	4.6e-3	65.11	12.36	60220.00	143.70	99.97
Judges100g	10.00	1.6e-4	4.1e-3	75.23	12.57	60888.00	107.60	99.99
Judges100h	33.00	5.5e-4	1.1e-2	48.99	13.98	60490.00	249.70	99.99
Judges100i	10.00	1.7e-4	1.1e-3	73.09	13.03	60400.00	130.40	99.99
Judges100j	60.00	1.0e-3	1.9e-2	11.70	14.76	59946.00	853.80	99.97

## 7.2 実社会から得られた問題例

問題例 DsumC [5] の目的関数の係数行列は NCAA 大学バスケットボールチームの得点差を基に作られており、60,031 個の 0-1 変数と 13,807,130 本の推移性制約を持つ。実験結果を表 7.2 に示す。

反復回数はラグランジュ乗数ベクトル  $(\mathbf{u}, \mathbf{v})$  を更新した回数である。316 回目の反復において最適解が見つかっている。しかしながら、その時点では上界がまだ緩く、2193 回目の反復終了後に双対ギャップ (GAP) が 0 になり最適性が証明された。

表 3 DsumC に対する実験結果

反復回数	$\omega_{up}$	$\omega_{low}$	GAP	$ U  +  V $	$\%^{UV}$	$ P_0  +  P_1 $	$\%^{PT}$	時間 (秒)
1	276183.00	276629.00	446.00	345	0.00	0	0.00	0.02
500	276219.00	276221.00	2.00	10410	0.08	50836	84.68	2.48
1000	276219.00	276220.00	1.00	10693	0.08	55953	93.21	8.66
1500	276219.00	276220.00	1.00	10882	0.08	56760	94.55	13.16
2000	276219.00	276220.00	1.00	11046	0.08	57156	95.21	17.36
2193	276219.00	276219.00	0.00	11091	0.08	57156	95.21	18.45

## 8 おわりに

本論文では、0-1 整数線形計画問題に定式化した線形順序付け問題に対し、ラグランジュ緩和法と釘付けテストを改善する方法を提案した。大規模な線形順序付け問題を解く際に、通常のラグランジュ緩和法では扱う乗数の個数が非常に多くなるため、それに伴い必要なメモリや劣勾配法にかかる時間も増加するが、提案解法ではそれらの節約を試みている。数値実験から、提案解法は特に実社会から得られた問題や、実社会の問題例を模した問題例について良い結果を示すことが確認できた。

また、最適解が必要となる場合には、提案した改良釘付けテストによって固定された変数の情報を用いて問題の規模を縮小し、その問題に対し既存の厳密解法を実行するという手段が考えられる。

## 参考文献

- [1] I. Charon and O. Hudry, "A branch-and-bound algorithm to solve the linear ordering problem for weighted tournaments", *Discrete Applied Mathematics* **154** (2006) 2097–2116.
- [2] M.L. Fisher, "The Lagrangian relaxation method for solving integer programming problems", *Management Science* **27** (1981) 1–18.
- [3] A.M. Geoffrion, "Lagrangian relaxation for integer programming", *Mathematical Programming Study* **2** (1974) 82–114.
- [4] J.G. Kemeny and J.L. Snell, *Mathematical Models in the Social Sciences*, Blaisdell, New York, 1972.
- [5] K. Pedings, A.N. Langville and Y. Yamamoto, "A minimum violations ranking method", to appear in *Optimization and Engineering*.
- [6] R. Sedgewick, *Algorithms in C*, 3rd Edition, Addison-Wesley, Boston, 2002.