最大カット問題に対する Semidefinite Programming 緩和

02501870 東京理科大学大学院 古屋 貴行* FURUYA Takayuki
02501520 東京工業大学大学院 藤江 哲也 FUJIE Tetsuya
02501480 東京工業大学大学院 藤沢 克樹 FUJISAWA Katsuki
01103520 東京工業大学大学院 小島 政和 KOJIMA Masakazu

1. はじめに

本稿では，整数計画問題に対する Semidefinite Programming (SDP) 緩和を扱う．整数計画問題の例としては，最大カット問題・最大クリーク問題・最大安定集合問題・グラフ分割問題などがある．本稿では，例として，最大カット問題に対する SDP 緩和 [3, 5] を示す．

2. 最大カット問題

2.1. 定式化

- $G=(V,E)$ を，頂点集合 V，枝集合 E からなる無向グラフとし，頂点数 $|V|=n$ とする．
- (V_1,V_2) を，頂点集合 V を 2 分割した場合の部分集合の組とする．つまり，$V_1,V_2 \subseteq V, V_1 \cup V_2 = V, V_1 \cap V_2 = \emptyset$ を満たすものとする．

最大カット問題とは，“無向グラフ $G=(V,E)$”と，“各枝 (i,j) に対する非負の重み w_{ij}”が与えられたときに，

$$w(V_1,V_2) \stackrel{\mathrm{def}}{=} \sum_{i\in V_1, j\in V_2} w_{ij}$$

の最大値と，このときの頂点集合の組 (V_1,V_2) を求める問題である．

ここでは，頂点に変数を対応させた場合の定式化を示す．頂点に対する特性ベクトル $\boldsymbol{x}$ の各成分を

$$x_i \stackrel{\mathrm{def}}{=} \begin{cases} +1 & \text{for } \forall i \in V_1, \\ -1 & \text{for } \forall i \in V_2 \end{cases}$$

とする．

最大カット問題を以下の整数計画問題に定式化する [3, 5].

$$\begin{aligned} W_{IP} = \text{maximize} \quad & \frac{1}{2}\sum_{\substack{(i,j)\in E\\ i<j}} w_{ij}(1-x_ix_j), \\ \text{subject to} \quad & x_i \in \{-1,+1\} \ \text{ for } \forall i \in V. \end{aligned}$$

2.2. SDP 緩和

最大カット問題に対する SDP 緩和問題は以下のようになる [3, 5].

$$\begin{aligned} W_{SDP} = \text{maximize} \quad & \frac{1}{2}\sum_{\substack{(i,j)\in E\\ i<j}} w_{ij}(1-X_{ij}), \\ \text{subject to} \quad & \boldsymbol{E}_{ii} \bullet \boldsymbol{X} = 1 \ \text{ for } \forall i \in V, \\ & \boldsymbol{X} \succeq \boldsymbol{O}. \end{aligned}$$

ただし，$\boldsymbol{E}_{ii} \stackrel{\mathrm{def}}{=} \boldsymbol{e}_i\boldsymbol{e}_i^T$ (for $\forall i \in V$) とする．

さらに，表記を簡略化するため，ラプラス行列

$$\boldsymbol{L} \stackrel{\mathrm{def}}{=} \frac{1}{2}\begin{pmatrix} \sum_{j\neq 1} w_{1j} & -w_{12} & \cdots & -w_{1n} \\ -w_{12} & \sum_{j\neq 2} w_{2j} & \ddots & \vdots \\ \vdots & \ddots & \ddots & -w_{(n-1)n} \\ -w_{1n} & \cdots & -w_{(n-1)n} & \sum_{j\neq n} w_{nj} \end{pmatrix}$$

を導入すると，最大カット問題に対する SDP 緩和問題は以下のようになる [3, 5].

$$\begin{aligned} W_{SDP} = \text{maximize} \quad & \frac{1}{2}\boldsymbol{L} \bullet \boldsymbol{X} \\ \text{subject to} \quad & \boldsymbol{E}_{ii} \bullet \boldsymbol{X} = 1 \ \text{ for } \forall i \in V, \\ & \boldsymbol{X} \succeq \boldsymbol{O}. \end{aligned}$$

このとき，

*東京理科大学大学院工学研究科経営工学専攻修士 2 年，
E-mail:furuya@ms.kagu.sut.ac.jp

- 整数計画問題の最適値 $W_{IP} \leq$ SDP 緩和問題の最適値 W_{SDP} (つまり，上界値),
- W_{SDP}と W_{IP}との理論的な誤差

$$\delta \stackrel{\text{def}}{=} \frac{|W_{SDP} - W_{IP}|}{W_{SDP}} \leq 1 - 0.87856 = 0.12144$$

が成立する [3].

3. 数値実験

SDP に対する主双対内点法 [4] を用いて，最大カット問題に対する SDP 緩和問題を解いた．プログラミング言語は C++ [2]，計算機は Sun SPARC station 20 を使用した．

表 1に，例としてサイクルの結果を示す．簡単のため，枝の重み $w_{ij} = 1$ (for $\forall(i,j) \in E$) とした．

表 1より，偶サイクルの場合には，すべて最適解が得られることがわかる．また，奇サイクルの場合には，すべて上界値が得られることがわかる．

4. おわりに

本稿では，整数計画問題に対する Semidefinite Programming 緩和の例として，最大カット問題を扱った．詳細は，発表当日に示す予定である．

参考文献

[1] T.Fujie, M.Kojima, "Semidefinite Programming Relaxation for Nonconvex Quadratic Programs," Research Reports B–298, Dept. of Mathematical and Computing Science , Tokyo Institute of Technology, 1995.

[2] K.Fujisawa, M.Kojima, "SDPA (Semidefinite Programming Algorithm) - User's Manual -," Research Report B-308, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, 1995.

[3] M.X.Goemans, D.P.Williamson, "Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming," unpublished manuscript, 1994.

[4] M.Kojima, S.Shindoh, S.Hara, "Interior-Point Methods for the Monotone Semidefinite Linear Complementarity Problem in Symmetric Matrices," Research Reports B–282, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, 1994, revised 1995.

[5] S.Poljak, F.Rendl, "Node and Edge Relaxations of the Max-Cut Problem," Report 266, CDLDO–28, Technische Universität Graz, 1993.

表 1: サイクルの結果 ($w_{ij} = 1$ for $\forall(i,j) \in E$)

$\|V\|$	最適値 W_{IP}	SDP 緩和 W_{SDP}	反復回数	time (秒)	誤差 δ(%)
3	2	2.250000	15	0.0167	11.111
4	4	4.000000	16	0.0167	0.000
5	4	4.522542	16	0.0333	11.554
6	6	6.000000	16	0.0833	0.000
7	6	6.653391	16	0.1000	9.820
8	8	8.000000	16	0.1167	0.000
9	8	8.728617	16	0.1167	8.347
10	10	10.000000	17	0.1500	0.000
11	10	10.777211	16	0.1833	7.212
12	12	12.000000	17	0.2333	0.000
13	12	12.811121	16	0.2500	6.331
14	14	14.000000	17	0.3167	0.000
15	14	14.836106	16	0.3833	5.636
16	16	16.000000	17	0.4500	0.000
17	16	16.855271	16	0.4833	5.074
18	18	18.000000	17	0.5667	0.000
19	18	18.870432	17	0.6833	4.613
20	20	20.000000	17	0.8333	0.000