OR 研究の最前線

半正定値計画を用いた最大カット問題の.878近似解法

松井　知己

1. はじめに

ここ数年, 組合せ最適化の分野では, 近似解法が流行っているが, それはＯＲの教科書に載っているような従来の解法とだいぶ異なっている. 例えば論文題目を見ても, 0.878 近似解法とか, 0.931 近似解法などと奇妙な数字が並んでいる. 本稿では, Goemans and Williamson によって 1994 年に提案された, 半正定値計画を用いた最大カット問題に対する 0.878 近似解法について解説する.

この論文は, 0.878 といった奇妙な数字が出現し始めた理由の１つであり, 半正定値計画 (線形行列不等式) という非線形計画の話題と, 最大カット問題という組合せ最適化の話題を結びつけ, 現在も続く大きなムーブメントを起こした. また提案された解法では, 半正定値緩和とランダマイズド算法という, 新しい概念が見事に融合している.

この解法の出現以降, 非線形の研究者と組合せ最適化の研究者が交じり合う研究会も珍しくなくなり, 両方の分野の知識を必要とする研究が瞬く間に広まった. 若手の研究者の間には, このような雰囲気のもとで育ち, 非線形計画と組合せ最適化という分野の区別無く 議論を自在に進められる頼もしい人たちも増えている.

本稿では, この解法の近似比率である 0.878 という奇妙な数字の由来に限定して解説する. 知りたい事をこれだけに絞れば, 組合せ最適化や半正定値計画に関する専門的な知識は必要無い. 必要な知識は「なんと！」 確率と行列と平面幾何学に関する基本的な知識である.

2. 近似解法

ある解法によって求められる解が, 最適解からある程度の距離しか離れていない事が保証できるとき, その解法を近似解法と呼ぶ事が多い. 他方, そのような保証ができないとき, その解法を発見的解法と呼ぶ事が多い. しかしながら, 上記のような「近似解法」と「発見的解法」の使い分けは, 研究者の間でも必ずしも同意されている訳ではない. 目的関数を最大化する最適化問題に対するある解法が, 最適値の α 倍以上の目的関数値を持つ許容解を必ず見つけるとき, この解法は α 近似解法であると呼ばれる. ただし α は $0 \leq \alpha \leq 1$ を満たす数であり, 近似比率と呼ばれる. 最適値が負の値を取る可能性のある問題に対しては, この定義はあまり意味を持たない. 目的関数を最小化する問題でも, 近似解法とその近似比率を同様に定義できる. ただしこの場合は, 近似比率は 1 以上の数字となる.

3. 最大カット問題

最大カット問題とは以下のような問題である. 無向グラフ $G = (V, E)$ と各枝 $e = \{i, j\} \in E$ に対し, 非負の枝重み $w_e = w_{ij} = w_{ji}$ が与えられているとする. 頂点の部分集合 $U \subseteq V$ に対するカット $\delta(U)$ を, 枝の部分集合 $\delta(U) = \{\{i, j\} \mid i \in U \not\ni j$ または $i \notin U \ni j\}$ と定義し, カットの重み $w(U)$ をカット中の枝の重みの総和 $w(U) = \sum_{\{i,j\} \in \delta(U)} w_{ij}$ と定義する. 最大カット問題は, 重み最大のカットを求める問題である.

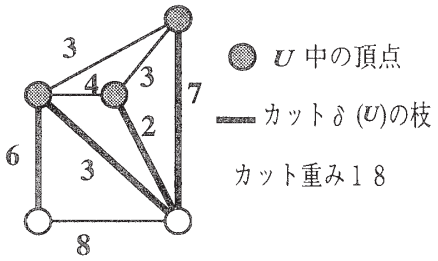

図 1: カットの定義.

最大カット問題は NP- 困難と呼ばれる難しい問題であることが知られている. 電気回路の基板の設計においてビア (表面から裏面に配線を通すために空ける穴) の数を最小にする問題 が最大カット問題の変種となる事が知られている. また, 全ての変数が 0 または 1 の値を取るとき, 2 次関数 (凸である必要はない) を最大化 (最小化でも構わない) する問題は, 最大カット問題に変形されることが知

まつい ともみ 東京大学大学院工学系研究科計数工学専攻
〒 133-8656 東京都文京区本郷 7-3-1
URL: http://www.misojiro.t.u-tokyo.ac.jp/~tomomi/

られている．ただし最大カット問題に変形するには，「全ての変数が0または1の値を取る」という以外には制約があってはならない．

最大カット問題の1/2近似解法は簡単に作ることができる．この手法は，後に出てくるランダマイズド算法の理解の際に役立つので，お付き合いいただきたい．算法は簡単で，「各頂点を1/2の確率で選んだ頂点集合をUとし，カット$\delta(U)$を解とする」というだけである．これを何度も繰り返し実行し，得られたカットの中で最も重みの大きなカットを出力する．繰り返しの回数は，時間が許す限り行なう事にする．では，解として得られるカットの重みはどうなるだろう．そこで上記の算法を1回実行したとき得られるカットの重みの期待値を算定しよう．任意の枝に対しその両端点の内ただ1つのみがUに入る確率は1/2となる．ゆえに，任意の枝に対し その枝が算法を(1回)実行して得られるカットに含まれる確率は1/2であり，カットの重みの期待値は枝重みの総和の1/2となる．枝重みが非負であることから，最大カット重みは，枝重みの総和以下である．すなわち

$$\begin{pmatrix}\text{カットの}\\ \text{期待値}\end{pmatrix} = \frac{1}{2}\begin{pmatrix}\text{枝重みの}\\ \text{総和}\end{pmatrix} \geq \frac{1}{2}\begin{pmatrix}\text{最大カット}\\ \text{重み}\end{pmatrix}$$

が成り立つ．上記の算法を何度も繰り返せば，期待値以上の重みを持つカットが得られる確率は限りなく1に近づく．非常に簡単だが，これがランダマイズド算法の典型的な例であり，またランダマイズド算法の近似比率の算定の一例である．この例の様に，ランダマイズド算法の1回の実行時間が，問題例の入力サイズの多項式時間であるとき，多項式時間のランダマイズド算法であると言う．

4. 最大カット問題の 0.878 近似解法

本節ではdを任意の正整数として議論を進める．与えられたグラフの頂点集合を$V=\{1,2,\ldots,n\}$とする．d次元座標空間R^dの原点を中心とする，半径$1/\pi$のd次元球の表面をSとする．$1/\pi$という半径は，大円の半円弧の長さが1になるように定められている．本節の議論は正整数dがいくつであっても成り立つのだが，Sとして3次元球の表面の例を図示したので，それらを参考に以下を読んでいただければ幸いである．球面S上の任意の2点$\boldsymbol{v},\boldsymbol{v}'$に対し，2点を通る大円の弧のうち短い方の長さを$d(\boldsymbol{v}\boldsymbol{v}')$と書く．

与えられたグラフGの頂点を球面S上に配置して，グラフを球面上に貼り付けてみよう．このとき「グラフの各枝の重さと枝の両端点の距離の積の総和」を最大化するような頂点の配置を求める問題を考える．正確には，以下のように記述される問題である，

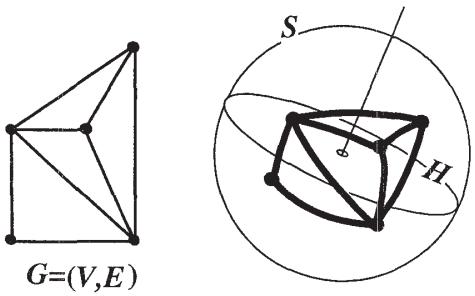

図 2: 球面上にグラフを張り付ける．

$$\text{MC: } \max\left\{\sum_{\{i,j\}\in E} w_{ij}\, d(\boldsymbol{v}_i\boldsymbol{v}_j) \mid \boldsymbol{v}_i \in S \ (\forall i \in V)\right\}.$$

実はこの問題は最大カット問題と本質的に等しい．正確には以下が成り立つ．

定理 1 最大カット問題の最適解(の1つ)を$\delta(U)$とする．球面S上の任意の点$\boldsymbol{v}^*$に対し，

$$\boldsymbol{v}_i^* = \begin{cases} \boldsymbol{v}^* & (i \in U), \\ -\boldsymbol{v}^* & (i \in V \setminus U), \end{cases}$$

と定義される解$(\boldsymbol{v}_1^*,\ldots,\boldsymbol{v}_n^*)$は問題$MC$の最適解である．

この定理だけでも十分美しい．球面上の頂点の配置問題は，最適解では頂点が球面全体に星空のように広がると予想されるのだが，それを裏切って，北極と南極のような対蹠点対(「たいせきてんつい」または「たいしょてんつい」)の位置に頂点が集中することになる．またこの定理は，最大カット問題という離散的な問題から，球面上の点の配置という連続的な非凸最適化問題への変形を示しており，近年の大域的最適化の研究の変化に対応している事にも注目されたい．

この定理の証明は，秀抜なアイデアに基づいており美しい．専門的な知識を前提としないので，是非読み進んでいただきたい．さらに証明自体がこの後構築する算法を示唆しており，アルゴリズムの魅力も備えている．

証明：頂点配置$(\boldsymbol{v}_1^*,\ldots,\boldsymbol{v}_n^*)$に対する問題MCの目的関数値は，球面$S$上の大円の半円弧の長さが1であることから，最大カット問題の最適値に等しい．

以下で，MCの任意の許容解$(\boldsymbol{v}_1,\ldots,\boldsymbol{v}_n)$に対し，その目的関数値 $\sum_{\{i,j\}\in E} w_{ij}\ d(\boldsymbol{v}_i\boldsymbol{v}_j)$ が最大カットの重

み以下であることを示す．この事実を示すことができれば，最大カットの重みと同じ目的関数値を持つ頂点配置 $(v_1^*, \ldots, v_n^*)$ が，最大化問題 MC の最適解 (の 1 つ) である事が分かる．

ここで，原点を通る (超) 平面を 1 つ思い浮かべていただきたい．球面上に貼り付けられたグラフの頂点は，この超平面のどちら側にあるかで 2 分割される (超平面上には頂点は無いとする)．この超平面を横切るグラフの枝全てを集めると，これはグラフのカットとなる．以下ではこのカットについて詳細に議論しよう．

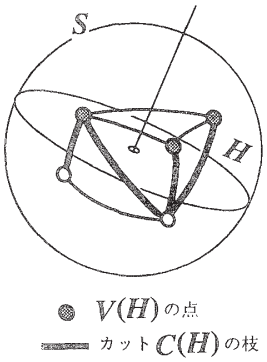

図 3: 超平面による 2 分割で生成されるカット．

原点を境界に含む閉半空間 H に対し，球面上の点集合 $\{v_1, \ldots, v_n\}$ のうち H に含まれるものに対応する頂点集合を $V(H)$ とする．また $V(H)$ に対応するカットを $C(H)$，カット $C(H)$ の重みを 閉半空間 H に対応するカットの重みと呼ぶ．原点を境界に含む閉半空間 H をすべて等確率で発生させるとき，対応するカットの重みの期待値 z について議論しよう．

以下では，許容解 (球面上の点) $(v_1, \ldots, v_n)$ に対応する目的関数値 $\sum_{\{i,j\}\in E} w_{ij}\ d(v_i v_j)$ が，上記で定義した期待値 z となる事を証明する．上記の確率分布に従って発生させた閉半空間を H としよう．グラフ中の枝 $\{i,j\}$ がカット $C(H)$ に含まれる確率は，閉半空間 H の境界である超平面が，v_i と v_j を分離する確率である．2 点 v_i と v_j を含む大円を D とすると，上記の確率は，大円 D を 2 つの半円で分割する際に，2 点 v_i と v_j が分離する確率に等しい．(ただし半円への分割の方法は，全てが等確率で起こるとする.) 簡単に分かるように，この確率は v_i と v_j を結ぶ円弧の長さが，大円の半円弧に対して占める割合であり，(半円弧の長さが 1 である事より) 長さ $d(v_i v_j)$ に等しい．

以上より，得られるカットの重みの期待値は，各枝に関

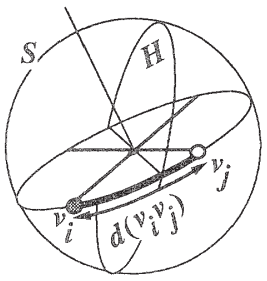

図 4: 超平面により v_i と v_j が分離する．

して，その枝が閉半空間 H によって生成されるカットに含まれる確率と 枝重みの積を計算し，全ての枝について足し合わせたものに等しい．すなわち，期待値 z の値は $z = \sum_{\{i,j\}\in E} w_{ij}\ d(v_i v_j)$ となる．この値は，頂点配置 $(v_1, \ldots, v_n)$ に対応する問題 MC の目的関数値に他ならない．

上記より，任意の許容解 $(v_1, \ldots, v_n)$ に対し，その目的関数値 $\sum_{\{i,j\}\in E} w_{ij}\ d(v_i v_j)$ は，原点を通る閉半空間から生成されるカットの重みの期待値に等しく，それは (いくつかのカットの重みの重み付き平均値であるから) 最大カット重み以下である．■

以下では，頂点配置 $(v_1^*, \ldots, v_n^*)$ を 最大カット配置と呼ぶ．問題 MC は最大カット問題と本質的に同じであり，依然として効率良く解く事は困難である．

次に，2 点 v_i, v_j 間の距離を直線距離 (ユークリッド距離) の 2 乗で置き換えた問題について議論する． 球面 S 上の任意の 2 点 v, v' に対し，2 点のユークリッド距離の 2 乗の $\pi^2/4$ 倍を $f(vv')$ と書く，すなわち

$$f(vv') = (\pi^2/4)||v - v'||^2$$

である．ちなみに

$$f(vv') = (1/2)(1 - \pi^2 v^\top v')$$

と書くこともできる．この距離は，$d(vv') = 1$ の時，すなわち v, v' が直径の両端となっているとき，$f(vv') = 1$ となるように設定されている．

2 本の (位置) ベクトル v と v' の角度を θ[rad] とすると，$d(vv') = \theta/\pi$，$f(vv') = (1 - \cos\theta)/2$ となることを，初等的な平面幾何学の知識を用いて導くことができる．すると，θ の値を 0 から π まで変化させた時の，距離 $d(vv') = \theta/\pi$ と $f(vv') = (1 - \cos\theta)/2$ の値は，図 6のような直線と破線で表される．図から読み取れるよ

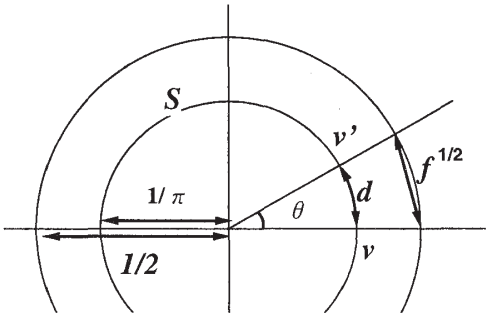

図 5: 距離 $f(vv')$ と $d(vv')$.

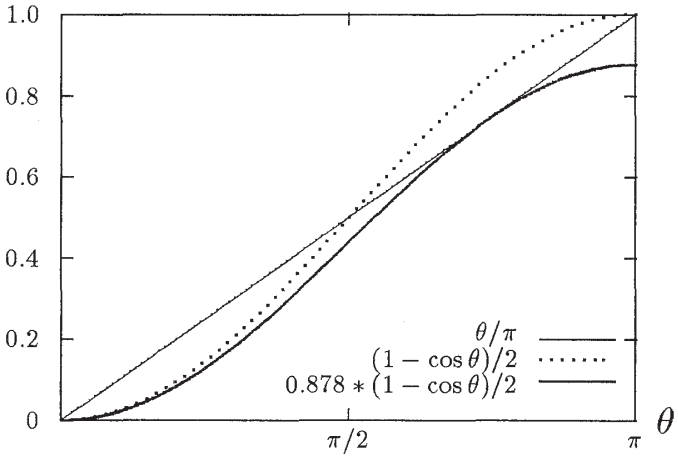

図 6: 関数 θ/π と $(1-\cos\theta)/2$ の関係.

うに, この 2 つの距離は θ が $0, \pi/2, \pi$ の 3 つの値で一致する. さらに, 図 6から予想される様に, $f(\boldsymbol{v}\boldsymbol{v}')$ で測った距離の 0.878 倍は, $d(\boldsymbol{v}\boldsymbol{v}')$ で測った距離よりいつも短くなる. すなわち以下が成り立つ.

補題 **2** 球面 S 上の任意の 2 点 $\boldsymbol{v}, \boldsymbol{v}'$ に対し, $d(\boldsymbol{v}\boldsymbol{v}') \geq \left(\min_{0\leq\theta\leq\pi} \frac{2}{\pi}\frac{\theta}{1-\cos\theta}\right) f(\boldsymbol{v}\boldsymbol{v}')$ が成り立つ. またこのとき, $\min_{0\leq\theta\leq\pi} \frac{2}{\pi}\frac{\theta}{1-\cos\theta} > 0.878$ である.

正確には $\min_{0\leq\theta\leq\pi} \frac{2}{\pi}\frac{\theta}{1-\cos\theta} = 0.87856\cdots$ であり, 正確な値は数値計算の手法を用いて求めなければならない.

最大カット配置 $(\boldsymbol{v}_1^*, \ldots, \boldsymbol{v}_n^*)$ については, 球面上の任意の頂点対は, 同一場所か対蹠点にあることから, どちらの距離で測っても値は一致し, $\sum_{\{i,j\}\in E} w_{ij}\ d(\boldsymbol{v}_i^*\boldsymbol{v}_j^*) = \sum_{\{i,j\}\in E} w_{ij}\ f(\boldsymbol{v}_i^*\boldsymbol{v}_j^*)$ が成り立つ.

ここで問題 MC における距離 $d(\boldsymbol{v}_i\boldsymbol{v}_j)$ を $f(\boldsymbol{v}_i\boldsymbol{v}_j)$ で測った距離で置き換えた, 以下の問題

$$\overline{\mathrm{MC}}: \max\{\sum_{\{i,j\}\in E} w_{ij}\ f(\boldsymbol{v}_i\boldsymbol{v}_j) \mid \boldsymbol{v}_i \in S\ (\forall i \in V)\}.$$

について議論しよう. 距離 $f(\boldsymbol{v}_i\boldsymbol{v}_j)$ と $d(\boldsymbol{v}_i\boldsymbol{v}_j)$ の定義より, 以下の性質が成り立つ.

定理 **3** 問題 MC の最適解 $(\boldsymbol{v}_1^*, \ldots, \boldsymbol{v}_n^*)$ および問題 $\overline{MC}$ の最適解 $(\boldsymbol{v}_1', \ldots, \boldsymbol{v}_n')$ に対し,

$$\sum_{\{i,j\}\in E} w_{ij}\ d(\boldsymbol{v}_i'\boldsymbol{v}_j') \geq 0.878 \sum_{\{i,j\}\in E} w_{ij}\ f(\boldsymbol{v}_i'\boldsymbol{v}_j')$$

$$\geq 0.878 \sum_{\{i,j\}\in E} w_{ij}\ f(\boldsymbol{v}_i^*\boldsymbol{v}_j^*) = 0.878 \sum_{\{i,j\}\in E} w_{ij}\ d(\boldsymbol{v}_i^*\boldsymbol{v}_j^*)$$

が成り立つ.

証明： 最初の不等式は補題2から導かれる. 次の不等式は, 配置 $(\boldsymbol{v}_1', \ldots, \boldsymbol{v}_n')$ が 問題 $\overline{\mathrm{MC}}$ の最適解である事から導かれる. 最後の等式は, 最大カット配置では 2 つの距離が一致することから導かれる. ■

上記の定理より, 問題 $\overline{\mathrm{MC}}$ を解いて得られる配置では, MC の目的関数値が問題 MC の最適値の 0.878 倍以上となっている事が保証される. すなわち, 問題 $\overline{\mathrm{MC}}$ を解く事で, 最大カット問題 MC の最適値の近似値を知る事ができる. 以下では, 近似値だけでなく近似解を得る方法について議論する.

実はその方法は, 定理1の証明そのものである. 問題 $\overline{\mathrm{MC}}$ を解いて得られる配置に対して, 定理1の証明で用いた手法に従い, 原点を境界に含む半閉空間をすべて等確率として 1 つ発生させて, 対応するカットを得るだけで良い. この方法で得られるカットの重みの期待値は, 問題 $\overline{\mathrm{MC}}$ を解いて得られる配置 $(\boldsymbol{v}_1', \ldots, \boldsymbol{v}_n')$ に対する目的関数値 $\sum_{\{i,j\}\in E} w_{ij}\ d(\boldsymbol{v}_i'\boldsymbol{v}_j')$ に他ならない. またこの期待値が最大カットの 0.878 倍以上となっている事は, 定理 3で示した.

以上で, 0.878 という値の由来の説明は終了である. では半正定値計画は何処に関わるのか？ それは, 問題 $\overline{\mathrm{MC}}$ を解く際である. 次節では, これについて簡単に解説しよう.

5. 半正定値計画

前節の近似解法を実行するには, 問題 $\overline{\mathrm{MC}}$ を解く必要がある. 距離 $f(\boldsymbol{v}\boldsymbol{v}')$ の定義から問題 $\overline{\mathrm{MC}}$ は

$$\begin{aligned}\overline{\mathrm{MC}}\text{-}1:\ \max.\quad & (1/2)\textstyle\sum_{\{i,j\}\in E} w_{ij}(1-\pi^2\boldsymbol{v}_i^{\mathsf{T}}\boldsymbol{v}_j)\\ \text{s. t.}\quad & \boldsymbol{v}_i \in S\ (\forall i \in V),\end{aligned}$$

と書くことが出来る. 上記の問題は制約式に 2 次等式の入った 2 次計画である.

以下では, この問題が半正定値計画に変形される事を示す. 変数 y_{ij} $(i,j \in \{1,\ldots,n\})$ を新たに導入し, $y_{ij} =$

$\pi^2 v_i^\top v_j$ と変数変換を行なうと, 行列 $Y = (y_{ij})$ は次の性質；(1) $\forall i \in V$, $y_{ii} = 1$, (2) Y は半正定値対称行列, を満たす. これより以下の問題

$$\begin{aligned} \overline{\text{MC}}\text{-2: max.} \quad & (1/2)\sum_{\{i,j\}\in E} w_{ij}(1-y_{ij}) \\ \text{s. t.} \quad & y_{ii} = 1 \ (\forall i \in V), \\ & Y = (y_{ij}) \text{は半正定値対称行列}, \end{aligned}$$

は $\overline{\text{MC}}$-1 の緩和問題となる. すなわち $\overline{\text{MC}}$-2 の最適値は $\overline{\text{MC}}$-1 の最適値以上である. $\overline{\text{MC}}$-2 の最適解 (の 1 つ) を Y^* としたとき, 行列 $(1/\pi^2)Y^*$ を Cholesky 分解することにより, $(1/\pi^2)Y^* = X^\top X$ が得られたとする.

ここで, 前節での議論の際, 座標空間の次元 d を固定しておかなかった理由が出現する. 問題 MC を定義した座標空間の次元 d を, 行列 X の行数に (天下り的に) 設定する. X の列ベクトルを $(v_1, \ldots, v_n)$ とすると, これは問題 $\overline{\text{MC}}$-1 の許容解となり, 対応する目的関数値は $\overline{\text{MC}}$-2 の最適値と一致する. ゆえに, X の列ベクトル集合は問題 $\overline{\text{MC}}$-1 の最適解である. 上記より, 問題 $\overline{\text{MC}}$ の最適解は, 半正定値計画問題 $\overline{\text{MC}}$-2 を解き, 得られた行列を Cholesky 分解することで得られる. 半正定値計画は, 内点法等を用いて多項式時間で解けることが知られている. 詳しくは, [5,7] 等を参照されたい.

6. 計算機実験

Goemans and Willimason は、計算機実験についても報告をしている. 表 1は論文 [2] からの抜粋である. 4 種類のグラフを生成し, 計算時間と解の精度を計っている. 問題タイプ等の詳細は [2] をご覧いただきたい. 解の精度は, 実際に得られたカットの重さを, 半正定値計画問題を用いて得られた上界値で割った値である. ランダマイズド算法におけるカットの生成は, 閉半空間をランダムに 50 個生成して最も良いものを選択している. 半正定値計画問題を解くソフトは, Vandelbei によって作られたものを用いたとある. 実験で確認された解の精度は, すべて 0.9 以上となっている.

表 2と表 3は, 京都大学大学院工学研究科建築学専攻の藤沢克樹氏による実験結果である. 表 2は [3] に記載されたものの抜粋である. 表 2, 3は各サイズのグラフ 1 つについて解いた結果である.「SDP」と「メモリ」は, 半正定値計画問題を解くのに要した時間とメモリ量である. RANDOM, SDP-R, TABU という 3 つの解法の性能を比較している. RANDOM は, 2 節で紹介した 1/2 近似解法である. SDP-R が 0.878 近似解法である. TABU は, 典型的な発見的解法である, 詳しくは [3] をご覧いただきたい. 表には, 各々の解法を用いて得られたカットの重さを, 半正定値計画問題を解いて得られた上界で割った値を記した. TABU が非常に良い解を得ていることが分かる. RANDOM, TABU の 2 つは, 100 秒間実行して得られたカットの中で, 最も良いものを出力として選んでいる. SDP-R は、半正定値計画問題を解いた後, ランダムなカット生成を 100 秒間行って得られたカットのうち, 最も良いものを出力として選んでいる. SDP-R は, ある程度良い解を出力しているが, 半正定値計画問題を解くのに非常に長い時間がかかっている. このため, 1996 年当時この結果が RAMP シンポジウムで発表されたときは, 0.878 近似解法の実用性は低いと感じられた. しかし現時点では, 状況がまったく異なっている. 表 3は藤沢克樹氏の御厚意により, 表 2と同じ問題を解いた際の, 最新の実験結果をお知らせいただいたものを記載している. また, 頂点数 500 と 1000 の問題に対する追加データもいただいたので, 一緒に記載した. 数年間で, 半正定値緩和問題の求解時間が驚異的に短くなっていることが分かる. 半正定値計画の解法と計算機の急速な進歩が, 組合せ最適化の研究の方向に強い影響を与えることが, この表 1 つから読み取れる. このデータでさえ, すでに過去のものとなりつつある.

表 1: 計算実験 : Vanderbei : Sun SPARC Station 1

問題タイプ	頂点数	問題数	精度	時間 [秒]
A	50	50	0.96988	36.28
	100	20	0.96783	323.08
	200	5	0.97209	4629.62
B	50	50	0.97202	23.06
	100	20	0.97097	217.42
	200	5	0.97237	2989.00
C	50	50	0.95746	23.53
	100	20	0.94214	306.84
	200	5	0.92362	2546.42
D	50	50	0.95855	27.35
	100	20	0.93984	355.32
	200	5	0.93635	10709.42

最大カット問題の近似解法として解く半正定値計画は, 与えられたグラフの枝密度が, 扱う半正定値行列の密度と密接に関わる. 与えられたグラフの枝密度が疎であれば, 半正定値計画のソフトの中でも, 疎行列を扱うデータ構造を使用しているものが効率的になる. 半正定値計画問題を解くソフトについては, [7] や, 内点法ホームページ
http://www-c.mcs.anl.gov/home/otc/InteriorPoint/
を参照されたい. 下記の Ye 教授のホームページでは, 大規模で疎なグラフに対応したソフトが公開されている.
http://dollar.biz.uiowa.edu/col/ye/

7. おわりに

論文 [2] では, 他にも MAX 2SAT 問題の 0.878 近似解法, MAX SAT 問題の 0.7584 近似解法, MAX DICUT 問題の 0.79607 近似解法が提案されている. またその後の研究により, 近似比率がより良い解法も開発されている. 同時に, 近似比率の限界に対する研究もされており, 例えば最大カット問題を解く $83/84 = 0.99809\cdots$ より大きい近似比率を持つ多項式時間解法が存在する可能性は 絶望視されている (正確には, もしそのような解法が存在するならば NP=P となる事が示されている). 近年の近似解法の研究の紹介としては, 例えば解説記事 [6,4] などをお勧めする.

最大カット問題の近似解法については, いまだ分からない事も多い. Goemans and Williamson も論文で挙げている問題として, 0.878 近似解を求めるのに半正定値計画を解く必要ば本当にあるのか? という疑問がある. 半正定値計画という枠組は, 近似比率の算定に必要なだけであって, もっと簡単な算法で近似比率 0.878 の解が求まる可能性が考えられる. また, 球面上の配置問題 $\overline{\mathrm{MC}}$ は, 何次元の球面ならば解けるのかについても, 現在のところ詳細は不明である. 1 次元球面 (線分の両端点) ならば, 球面上の配置問題はもとの最大カット問題に本質的に等しい. しかし例えば 2 次元の球面 (すなわち円周) では, 配置問題が多項式時間で解けるかは不明である. さらに, 球面上の配置問題という連続的な問題に変形する事により, いままでとは全く異なった近傍に基づいた局所探索法が開発される可能性もある.

最近は, 組合せ最適化問題の算法として, 本稿で紹介した算法のように, 適当な緩和問題を解いて, それを確率的な変動を与えながら丸める, randomized rounding method (ランダム化丸め算法) の研究が数多くされている. 例えば [4] を参照されたい. 算法としても設計が容易であり, 他の手法との併用も可能である. 他の発見的解法との比較と, 実務問題への適用の研究が望まれる.

始めに述べたように, 非線形計画と組合せ最適化の分野が融合した研究が広まりつつある. 研究者の知識と視野の広さが要求される時代が急速に到来している事を実感する.

参考文献

[1] M. X. Goemans and D. P. Williamson: .878-approximation Algorithms for MAX CUT and MAX 2SAT. *Proceedings of 26th STOC*, Montreal, Quebec, Canada, 422–431, 1994.

[2] M. X. Goemans and D. P. Williamson: Improved Approximation Algorithms for Maximum Cut and Satifiability Problems Using Semidefinite Programming. *J. ACM*, **42** (1995), 1115–1145.

[3] 藤沢克樹: 組合せ最適化問題に対する近似解法. 第 8 回 RAMP シンポジウム論文集, 東京大学, 139–154, 1996.

[4] 浅野孝夫: 論理システム解析のための高性能近似アルゴリズム — 四角い問題を丸くして解く —. 情報処理, **39** (1998), 683–688.

[5] 小島政和: 半正定値計画とその組合せ最適化問題への応用. 離散構造とアルゴリズム V, 近代科学社, 203–249, 1998.

[6] 玉木久夫: 巡回セールスマン問題の近似アルゴリズム: 天才アローラによる 20 年ぶりの急進展. 情報処理, **39** (1998), 566–573.

[7] 土谷隆: 最適化アルゴリズムの新展開 — 内点法とその周辺 IV — 半正定値計画問題 II. システム／制御／情報, **42** (1998), 550–559.

表 2: 計算機実験 (1996 年 8 月) : SDPA Ver 1.0 : Sun SPARC Station 20 : Super SPARC II 125MHz

頂点数	枝密度 [%]	SDP [秒]	メモリ [Mbyte]	RANDOM 精度	SDP-R 精度	TABU 精度
124	2	636.5	5.1	0.6761	0.9648	0.9648
124	4	610.3	5.1	0.7040	0.9374	0.9485
124	8	637.1	5.1	0.7781	0.9470	0.9534
124	16	607.0	5.1	0.8052	0.9625	0.9648
250	1	7948.5	23.5	0.6303	0.9518	0.9612
250	2	7407.6	23.5	0.6612	0.9516	0.9438
250	4	7290.7	23.5	0.7318	0.9244	0.9448
250	8	7505.7	23.5	0.7712	0.9454	0.9567

表 3: 計算機実験 (1999 年 12 月) : SDPA Ver 5.00 : DEC ALPHA 21164 600MHz

頂点数	枝密度 [%]	SDP [秒]	メモリ [Mbyte]
124	2	3.0	5.0
124	4	2.8	5.0
124	8	2.8	5.0
124	16	2.8	5.0
250	1	22.3	20.0
250	2	21.0	20.0
250	4	21.3	20.0
250	8	21.3	20.0
500	1	208.0	79.0
1000	5	2989.1	314.0