

最大カット問題に対するベクトル割り当て解法

真下 征章 (指導教員:平田 富夫)
名古屋大学 工学部

1. 概要

本研究ではグラフの最大カット問題に対する発見的解法を与える。最大カット問題を2次元ベクトル割り当て問題に変換することで、性能のよい解を求める。ベクトルの割り当て方を工夫することでアルゴリズムの性能がどのように変化するかを調べた。

2. 問題の定義

2.1 最大カット問題

最大カット問題とは、各辺に対する非負の重み w_{ij} が与えられた無向グラフ $G = (V, E)$ に対し、頂点集合 V を2つの部分集合 V_1, V_2 に分割する時、 V_1 と V_2 の間にある辺の重みの総和 (カット値)

$$\sum_{(i,j) \in E, i \in V_1, j \in V_2} w_{ij}$$

が最大となる分割を求める問題である。

なお、この問題はNP困難であることが知られている。

2.2 2次元ベクトル割り当て問題

上記の問題を2次元ベクトル割り当て問題 [1] に変換する。2次元ベクトル割り当て問題とは、与えられた無向グラフ G の各頂点 $i \in V$ に2次元単位ベクトル $v_i = (\cos\theta_i, \sin\theta_i) (0 \leq \theta_i < 2\pi, \forall i \in V)$ を割り当て、

$$\sum_{(i,j) \in E} w_{ij} \times \text{diff}(\theta_i, \theta_j)$$

を最大化する問題である。ただし $\text{diff}(\theta_i, \theta_j) = \min\{|\theta_i - \theta_j|, 2\pi - |\theta_i - \theta_j|\}$ である。つまり、より重い辺で結ばれている頂点どうしのベクトルを遠ざけることでその頂点対が別々の部分集合に入る可能性を高くすることである。以下では頂点 i とそのベクトル v_i を同一視することがある。

文献 [2] では最大カット問題を半正定値計画問題に緩和して近似アルゴリズムを与えている。半正定値計画問題では、各頂点に対し n 次元単位ベクトルを割り当てて、本論文ではこれを2次元に制限している。

2.3 最大カット問題への変換

2次元ベクトル割り当て問題の解から最大カット問題の解を得るのは次のように行う。隣り合うベクトルとベクトルの間を2箇所選ぶことで頂点集合を2つに分割し、それを最大カット問題の解とする。

3. 提案アルゴリズム

まず全体の流れを説明する。グラフ G が与えられたら、初期解として各頂点 $i \in V$ に2次元単位ベクトルを $0 \leq \theta_i < 2\pi$ の範囲からランダムに割り当てる。そして各頂点 i に対対応するベクトルの位置を調整する。この操作をある回数繰り返し2次元ベクトル割り当て問題の解とする。その後2.3の方法で2分割を全通り実行し、それぞれのカット値を計算して最大値を出力する。

ベクトルの位置の調整は次の3つの方法で行う。

■Ⅰ.(最大角の最小化)

頂点 i に隣接する頂点 j どのの隣り合う角度が1番大きいところを見つけ、その真ん中に頂点 i を移動させる。なおこの方法は辺の重みを考慮していない。

■Ⅱ.(重みを考慮した最大角の最小化)

$\sum_{j:(i,j) \in E} w_{ij} \times \text{diff}(\theta_i, \theta_j)$ が最大となるような θ_i を計算し、そこに頂点 i を移動させる。

■Ⅲ.(隣接する頂点の移動)

頂点 i に隣接するすべての頂点 j について、そのベクトルをそれぞれ辺の重さに応じて頂点 i のベクトルから遠ざける。ただし π 以上にはしない。

4. 実験結果

問題例は Helmberg と Rendl[3] が作ったグラフを元にして、それぞれのグラフに対し辺の重みが「A:すべて1」「B:1~10からランダム」「C:1~50からランダム」の3種類を作成した。反復回数は500回とし、結果は各問題例について初期解を変えて10回実行したときのカット値の平均を求めた。なお実験に使用した計算機は Pentium(R)4 2.80GHz 512MB RAM である。実験結果を抜粋して表1に示す。

グラフ	頂点数	辺の数	I	II	III
A2	800	19175	10478.7	11443.1	11368.7
B2				64434.9	64494.7
C2				262522.6	297909.8
A5	2000	11778	7038.8	7509.6	7299.8
B5				43038.0	43082.3
C5				199967.7	199958.8

表1 実験結果 (抜粋)

重みがすべて1のグラフでは、Ⅱ.(重みを考慮した最大角の最小化)が一番いい結果を出し、Ⅰ.(最大角の最小化)はあまりいい結果が出なかった。重みがついているグラフでは、ⅡとⅢ.(隣接する頂点の移動)はあまり差がなかった。また、計算時間はⅢが非常に速く、Ⅰがその約2~3倍、ⅡはⅢより大幅に時間がかかった。

まとめとして、“最大角の最小化”は遅くはないが、あまりいい結果が出せなかった。“重みを考慮した最大角の最小化”は時間はかかるが、特に重みのないグラフでいい結果が出せた。“隣接する頂点の移動”は非常に高速で、平均的にいい結果が出せた。

今後の課題としては、他の研究結果との比較や提案アルゴリズムの性能の向上が求められる。

参考文献

[1] Burer, S., Monteiro, R.D.C., Zhang, Y.: Rank-two relaxation heuristics for max-cut and other binary quadratic programs. SIAM Journal on Optimization 12(2), 503-521 (2002)

[2] Michel X. Goemans and David P. Williamson. .878-Approximation Algorithms for MAX CUT and MAX 2SAT. In *Proceedings of the Twenty-Sixth ACM Symposium on Theory of Computing*, 1994.

[3] Helmberg, C., Rendl, F.: A Spectral Bundle Method for Semidefinite Programming. SIAM Journal of Computing, 10:673:696, 2000.