B-05

最大重みクリーク抽出法における分枝順序の検討

Study on Branching Ordering in an Algorithm for the Maximum Weight Clique Problem

森中 諒太† 清水 悟司 † 山口 一章 † 増田 澄男 †
Ryota Morinaka Satoshi Shimizu Kazuaki Yamaguchi Sumio Masuda

1. まえがき

最大クリーク問題とは，与えられたグラフ中から頂点数が最大となるクリークを求める問題であり，組合せ最適化問題の一つである．最大クリーク問題はバイオ情報処理やパターン認識，符号理論や並列計算などの分野への応用が知られている [1],[2]．最大重みクリーク問題は，各頂点に自然数の重みが付与し，グラフの中から重みの和が最大のクリークを求めよという形に最大クリーク問題を一般化したものである．最大クリーク問題と最大重みクリーク問題は NP 困難であることが知られている．

最大重みクリーク問題に対する厳密解法はこれまでにいくつか提案されているが，本稿では文献 [3] の手法（以下 Kumlander のアルゴリズムと記す）に着目する．Kumlander のアルゴリズムは，最初に入力のグラフ $G = (V, E)$ の頂点彩色を求め，その彩色を元に分枝順序を決定し，分枝限定法を実行する．

頂点彩色を求める際，各 S_i が独立頂点集合でありさえすればどのようなものであっても正しく動くことが保証されており，かつ，頂点彩色の良否が計算時間を大きく左右することが知られている．本稿では，いくつかの頂点彩色を用いて系列を作ったときの，Kumlander のアルゴリズムの性能を比較し，どのような彩色法が良いかを検討する．

2 準備

2.1 諸定義

グラフ $G = (V, E)$ において，$C \subseteq V$ なる C の任意の 2 頂点間に辺が存在するとき，C をクリークと呼ぶ. グラフ $G = (V, E)$ に存在するクリークのうちで，もっとも頂点数の多いものを最大クリークと呼び，各頂点に重み $w(\cdot)$ が与えられたとき，含まれる頂点の重みの合計が最大になるクリークを最大重みクリークと呼ぶ.

以上の用語について図 1 を用いて説明する．アルファベットは頂点名であり頂点の横の数字は重みを表している．図 1 のグラフにおいて $\{b, c, d, e\}$ の 4 頂点を考える．この 4 頂点の中で任意の 2 頂点は隣接しているので，これは頂点数が 4 のクリークである．$\{a, b, c, d\}$ は，a と d が隣接してないため，クリークではない．このグラフには $\{b, c, d, e\}$ や $\{a, b, c\}$，それらの部分集合などのクリークがあるが，最大重みクリークは $\{a, b, c\}$ でありその重みは 20 である．

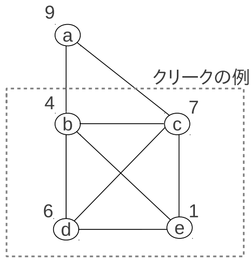

図 1　クリークの例

グラフ $G = (V, E)$ において $v \in V$ に隣接する頂点の集合を $N(v)$ と記す．グラフ $G = (V, E)$ の 2 頂点をランダムに選んだときにそれらの間に辺が存在する確率をそのグラフの辺密度と呼ぶ．辺密度はグラフ $G = (V, E)$ に対して

$$(\text{辺密度}) = \frac{2|E|}{|V| \times (|V| - 1)}$$

で計算される．

2.2 Greedy 彩色

Greedy 彩色は，使用可能な最も小さな番号の色を順に付けていくことで彩色を行う．まず頂点に何らかの順序を付ける．その順に $v_1, \cdots, v_n$ とする．次に，$S_1 = \emptyset$, $k = 1$ とし，その後，v_1, v_2 から順に，以下の処理を行う．

> 処理する頂点を v_i とする．$S_1, S_2, \cdots, S_k$ に入れられるかどうか（隣接している頂点が存在するか否か）調べる．入れられる集合の中で最も添え字の小さい集合に v_i を入れる．

最終的な k の値（使用した色数）は，頂点をどう順序付けするかに依存するが，頂点の次数（隣接する頂点数）の降順が良いことが経験的に知られている．

†神戸大学，Kobe University

2.3 飽和次数彩色 (DSATUR)

頂点 v の飽和次数とは，いくつかの頂点に色が付けられたグラフにおいて，v に隣接する頂点につけられた色の種類の数である．DSATUR [4] は，彩色されていない頂点の中で飽和次数が最大の頂点に，割り当て可能な最小の色番号を割り当てる．一つの頂点に彩色をする度に各頂点の飽和次数飽和次数を計算し直すので，DSATUR は Greedy 彩色と比べ計算時間が掛かるが，多くの場合，得られた彩色の色数は Greedy 彩色の色数以下となることが知られている．

3. 従来法 (Kumlander による最大重みクリーク抽出法)

Kumlander による最大重みクリーク抽出法は，最初に頂点を彩色し，各カラークラス毎に分枝を行い最大重みクリークを抽出する．本稿では頂点彩色の部分の改良を行うので，分枝限定法の部分の詳細については省略する．Kumlander のアルゴリズムにおける彩色の処理は以下の手順からなる．

1. 入力として与えられたグラフ $G = (V, E)$ について，重み $w(\cdot)$ の降順に頂点を Greedy 彩色する．

2. カラークラスごとに頂点を重みの降順にソートした系列を作る．

3. 色番号の小さいカラークラスを先頭に，全てのカラークラスの頂点系列をくっつけた一つの系列を作り，その系列中の頂点を順に $v_1, v_2, \cdots, v_n$ とする．

ここで，$V_i = \{v_i, v_{i+1}, \cdots, v_n\}$ とする．以降，文献 [5] のアルゴリズム中の backtrack search を改良したものを用いて，$i = n, n-1, \cdots, 2, 1$ の順に，V_i による頂点誘導部分グラフの最適解を順に求めていく．$V_1 = V$ なので，最終的に元の問題の最適解が求まる．

Kumlander のアルゴリズムでは，1. の頂点彩色は，以降の分枝限定法の分枝順序を定めると同時に，限定操作における上界計算の良否にも影響するので，計算時間は彩色の良否に大きく左右される．

4. 彩色法の検討

様々な彩色を実装し，その特徴について考察する．以下では Kumlander のアルゴリズムで用いられている重み降順の Greedy 彩色と比較する 4 つの彩色法を示す．

4.1 (重み $*\, n +$ 次数) の降順

この順序は，重みが異なる 2 頂点については重みの降順とし，同じ場合は次数の降順になるようにすることを意味する．重みのないグラフに対して Greedy 彩色を用いる場合は，次数の降順に行うと良いことが経験的に知られており，本方法はその自然な拡張の一つであると考えられる．

なお，Kumlander のアルゴリズム中，限定操作のために彩色を用いて最大重みクリークの重みの上界を計算するが，この順序付けによる上界は重み降順によるものよりも良かった．

4.2 飽和次数彩色 (DSATUR)

重みのないグラフを飽和次数彩色で彩色した場合，2.3 でも述べた通り DSATUR は色数を抑えることが出来る．Kumlander のアルゴリズムは各カラークラス毎に分枝が進んでいくので，カラークラスが少なくなれば計算時間を減らすことが出来るのではないかと考えた．本発表では，彩色後，各カラークラスにおける頂点は重みの降順に並べた．

ちなみに，DSATUR を使って上界計算を行ったところ，あまり良い値は得られなかった．

4.3 (重み $*\, n -$ 次数) の降順

この方法は，重みの大きな頂点をできるだけ最初のカラークラスに詰め込むことで，分枝が進んだ後に出てくる頂点の重みが小さくなり，枝刈りが起こりやすくなることを期待した．

ただし，上界計算に用いた場合は，単純な重みの降順よりも若干，悪くなった．

4.4 重みの昇順

ここでいう重みの昇順というのは，彩色の際に，v_n の側に置く頂点を先に定める方法である．他の彩色法が v_1 の側から順に決めていくのとは逆順になる．これは，部分問題中に頻繁に出現する，添字の大きな頂点に，できるだけ重みの小さなものを集めることで部分問題の重みを小さくし，枝刈りが起きやすくなることを目指したものである．

5. 計算機実験

Kumlander のアルゴリズム中の 1. の彩色について前述の 4 つの彩色法に置き換えたものの計算時間を，従来法の計算時間と比較した．実験で用いたグラフはランダムグラフで，頂点数は 100 から 100 刻みの 1000 まで，辺密度は 0.1 から 0.1 刻みの 0.9 までとした．計 90 パターンの条件のランダムグラフそれぞれ 100 個，計 9000 個に対し

表 1　実験結果

(a) 計算時間 (sec)

頂点数	辺密度	重みの降順	(重み $*n+$ 次数) の降順	DSATUR	(重み $*n-$ 次数) の降順	重みの昇順
	0.1	0.00019	0.00031	0.00067	0.0003	0.0003
	0.2	0.00023	0.00033	0.0011	0.0004	0.00029
	0.3	0.00037	0.0004	0.00126	0.00044	0.00032
	0.4	0.00059	0.0006	0.00176	0.0006	0.00042
100	0.5	0.00096	0.00105	0.00305	0.00103	0.00066
	0.6	0.00189	0.00201	0.00488	0.00195	0.00114
	0.7	0.00496	0.00499	0.015	0.00509	0.00273
	0.8	0.02065	0.0209	0.11682	0.01911	0.00835
	0.9	0.1267	0.14384	3.63848	0.1202	0.02551
	0.1	0.00035	0.00053	0.00168	0.00054	0.0005
	0.2	0.00075	0.00088	0.00248	0.00086	0.00061
	0.3	0.00171	0.0018	0.00485	0.00181	0.0013
	0.4	0.00406	0.00419	0.01046	0.00412	0.0031
200	0.5	0.01226	0.0125	0.03319	0.01258	0.00968
	0.6	0.05265	0.05654	0.19408	0.05272	0.042
	0.7	0.40424	0.41184	2.34546	0.40368	0.24445
	0.8	8.01634	8.51906	TimeOver	7.49781	4.92717
	0.9	TimeOver	TimeOver	TimeOver	TimeOver	TimeOver
	0.1	0.00071	0.00084	0.0027	0.00081	0.00066
	0.2	0.00194	0.00215	0.00547	0.00208	0.00153
	0.3	0.00564	0.00575	0.01356	0.00567	0.00426
	0.4	0.0187	0.01975	0.04924	0.01921	0.01657
300	0.5	0.08864	0.09119	0.2953	0.08908	0.06989
	0.6	0.663	0.66828	3.18029	0.62285	0.52492
	0.7	9.69891	10.16643	TimeOver	9.1537	7.45066
	0.8	TimeOver	TimeOver	TimeOver	TimeOver	TimeOver
	0.9					
	0.1	0.00142	0.00154	0.0045	0.0016	0.00125
	0.2	0.00424	0.00434	0.01055	0.00436	0.00323
	0.3	0.01507	0.01584	0.0372	0.01571	0.01355
	0.4	0.0694	0.06968	0.19223	0.06971	0.06655
400	0.5	0.42624	0.43832	1.61271	0.4137	0.45591
	0.6	4.75745	4.85241	27.8068	4.48343	5.29237
	0.7	TimeOver	TimeOver	TimeOver	TimeOver	TimeOver
	0.8					
	0.9					
	0.1	0.00198	0.00214	0.00651	0.00212	0.00145
	0.2	0.00818	0.00839	0.01962	0.00851	0.00607
	0.3	0.03553	0.03606	0.08695	0.0347	0.03132
	0.4	0.19233	0.19468	0.60923	0.195	0.19054
500	0.5	1.57864	1.58757	6.96477	1.54822	1.5941
	0.6	TimeOver	TimeOver	TimeOver	TimeOver	TimeOver
	0.7					
	0.8					
	0.9					

表 2　実験結果

(a) 計算時間 (sec)

頂点数	辺密度	重みの降順	(重み $*n+$ 次数) の降順	DSATUR	(重み $*n-$ 次数) の降順	重みの昇順
600	0.1	0.0028	0.00304	0.00936	0.00303	0.00216
	0.2	0.01437	0.01497	0.03494	0.01472	0.01171
	0.3	0.07296	0.07477	0.19105	0.07283	0.06917
	0.4	0.45914	0.47558	1.57726	0.47025	0.51423
	0.5	4.74399	4.84946	23.69246	4.7292	5.38932
	0.6	TimeOver	TimeOver	TimeOver	TimeOver	TimeOver
	0.7					
	0.8					
	0.9					
700	0.1	0.00408	0.00447	0.01269	0.0044	0.00307
	0.2	0.02468	0.02489	0.05748	0.02443	0.02075
	0.3	0.14385	0.1406	0.38259	0.14469	0.14765
	0.4	1.02651	1.04601	3.89294	1.01427	1.2014
	0.5	TimeOver	TimeOver	TimeOver	TimeOver	TimeOver
	0.6					
	0.7					
	0.8					
	0.9					
800	0.1	0.00579	0.00623	0.01744	0.00611	0.00441
	0.2	0.03807	0.03892	0.09189	0.03878	0.03539
	0.3	0.2524	0.25979	0.7058	0.24739	0.27283
	0.4	2.10147	2.07424	8.31957	2.09044	2.56654
	0.5	TimeOver	TimeOver	TimeOver	TimeOver	TimeOver
	0.6					
	0.7					
	0.8					
	0.9					
900	0.1	0.00775	0.00827	0.02173	0.00825	0.00623
	0.2	0.05655	0.05789	0.14211	0.05763	0.05734
	0.3	0.42852	0.41795	1.21173	0.41113	0.48167
	0.4	3.92452	4.05998	15.79449	3.84738	4.99722
	0.5	TimeOver	TimeOver	TimeOver	TimeOver	TimeOver
	0.6					
	0.7					
	0.8					
	0.9					
1000	0.1	0.01013	0.01072	0.02846	0.01096	0.00818
	0.2	0.08157	0.08271	0.20594	0.08226	0.08535
	0.3	0.6678	0.6823	2.01236	0.68452	0.80724
	0.4	7.08207	7.05282	30.24249	6.99857	9.64976
	0.5	TimeOver	TimeOver	TimeOver	TimeOver	TimeOver
	0.6					
	0.7					
	0.8					
	0.9					

て実験を行い，その実行時間の平均値を求めた．ただし，計算時間が 60sec を越えるものは TimeOver とする．実験環境の CPU は Intel(R)Core(TM)i7-2600 (3.40GHz)，プログラム言語は JAVA，OS は Ubuntu11.10(maverick) を用いた．

表 1，2 より以下のことが分かった．

- 上界計算法に用いた場合に最も良かった (重み$*n+$次数) の降順については，Kumlander のアルゴリズムに組み込んだときの計算時間が従来手法よりも長くなった．このことより，計算時間の良否は，彩色による上界の良否と必ずしも一致しないことがわかった．
- DSATUR については，色数は抑えることができるものの，上界の値が悪いデメリットの方が大きかったようで，良い結果は得られなかった．
- 辺密度が小さいときは重みの昇順が他に比べ高速であった．この結果より，辺密度が小さいときは，小さな重みの頂点を添字の大きな頂点にすることが効果的であると推測される．
- (重み*n-次数) の降順に関しては，辺密度が大きいときには，重みの降順と比べると Kumlander のアルゴリズムにおいて良い結果が得られた．

6. あとがき

本発表では，Kumlander のアルゴリズムに様々な頂点彩色法を組み込んでそれぞれの計算時間を検討した．Kumlander のアルゴリズムはその分枝順序においてまださまざまな工夫が可能であるが，実験結果より，簡単な変更により高速化できることを確認した．また，辺密度によってどの方法が優位になるかが変わることも確認した．

今後は，入力に応じてより良い彩色を求める方法について検討していく．例えば，入力によって彩色方法を使い分けるなど工夫することなどが挙げられる．

参考文献

[1] Malod-Dognin,Noel,Andonov,Rumen,Yanev,Nicola,“Solving Maximum Clique Problem for Protein Structure Similarity” Serdica Journal of Computing, Vol. 4, No 1, (2010), 93p-100p

[2] Samuel Rota Bulo and Marcello Pelillo, “A New Spectral Bound on the Clique Number of Graphs” Lecture Notes in Computer Science, 2010, Volume 6218/2010, 680-689

[3] D.Kumlander, “An exact algorithm for the maximum-weight clique problem based on a heuristic vertex-coloring,” Proc. 5th International Conference on Modeling, Computation and Optimization in Information Systems and Management Sciences, pp.202-208, 2004.

[4] Daniel Brelaz ”New Methods to Color the Vertices of a Graph,” Communications of the ACM, vol.22, no.4, pp.251-256, 1979.

[5] P.R.J. Östergård, “A New algorithm for the maximum-weight clique problem,” Nordic Journal of Computing,vol. 8, pp.424-436, 2001