OR メモランダム

このコラムは，ORにかかわる概念，知識(手法，原理)，それらの図解，よい教材や問題，実学ORの実施経験，そこから得られた知恵やアドバイス，失敗談と教訓，新しい視点，視座，フレームワーク，未だ解けていない問題，面白い研究テーマなどを，"新鮮に"，しかも"コンパクトに"表現し，提示していただくものです．ユニークなアイディア，フレッシュな見方，発想，だれかと意見をたたかわせたい問題提起など，ふるってご投稿ください．(原稿は，刷り上がり，半ページから3ページに納まるようにお書きください．簡単に／ 加筆訂正をお願いする場合があります)

最大クリーク問題の近似解法に関するコメント

久保　幹雄

近似解すら$\mathcal{NP}$-完全である難しい問題において，本稿のような大きな対象についても，多項式時間で驚くべき高い近似精度の達成を確認することができた．
猪瀬，山田，富田，高橋[1]

1. はじめに

平成5年9月6日の読売新聞の1面トップで「コンピュータの難問解明，集団内の最大派閥，超高速で選別」という記事が載っていた．ここで言う最大派閥問題とは，組合せ最適化およびグラフ理論における標準的な問題の1つである最大クリーク問題 (Maximum Clique Problem) のことである．筆者は，この記事および，その発端になった研究発表論文[1]を読んだ後，最適化問題に関与している研究者の1人として少なくともOR学会員の方々には，正しい情報を流す義務があると感じた．以下では，最大クリーク問題の現状と，この記事で紹介されている内容の問題点を示す．

2. 定義と計算量

まず，用語と問題の定義を説明しよう．完全グラフとは，すべての点の間に枝があるグラフである．無向グラフ$G=(V, E)$ (Vは点集合，Eは枝集合を表わす)が与えられたとき，点の部分集合C ($\subseteq V$) は，Cによって導かれたグラフが完全なときクリークと呼ばれる．最大クリーク問題とは，クリークサイズ$|C|$で最大になるクリークを求める問題である．もちろん，この問題は$\mathcal{NP}$-困難である[6]．

記事のもとになった研究発表[1]に「近似解すら$\mathcal{NP}$-完全である難しい問題」という記述がある．この言葉の真意を説明しよう．1992年に Arora et al. [2,3]は，最大3充足問題 (MAX3-SAT) のe-近似解を求める多項式時間のアルゴリズムは，$\mathcal{P}\neq\mathcal{NP}$ の下では存在しないことを証明した．この結果を用いることによって，任意のϵ (>0) に対して，最大クリーク問題の最適値の$|V|^{\epsilon}$ 分の1以上の近似値を算出する多項式時間のアルゴリズムが$\mathcal{P}\neq\mathcal{NP}$ の仮定のでは存在しないことが示される．

3. 現状

最大クリーク問題に対する現状を知るための最もよい情報源はDIMACS である．DIMACS とは DIscrete MAthematics and theoretical Computer Science の略称であり，Rutgers University, Princeton University, AT & T Bell Laboratories, Bellcore を母体とした研究センターである．毎年テーマを決めて，

くぼ　みきお　東京商船大学　流通情報工学
〒135 江東区越中島2－1－6
e-mail: kubo@ship2. ipc.tosho-u.ac.jp

そのテーマに対して世界中の研究者たちが協力して集中的に研究を行なっている．1993年度のテーマは，最大クリーク問題，グラフ彩色問題，充足可能性問題の3つであり，ベンチマーク問題の収集，アルゴリズム開発および性能テスト，サーベイ論文の整備，理論的な結果の発表など，テーマ別に精力的に研究が行なわれている．最大クリーク問題に関する成果は無記名 ftp サイト dimacs. rutgers.edu の pub/challenge/graph に集められており，誰でも自由に得ることができる（以下では，directory 名のみを記すことにする）．

最大クリーク問題に対する最適解法をいくつか紹介しよう．簡単な陰的列挙法が Carraghan と Pardalos によって作られている[5]．彼らの解法はグラフが疎なときは効率的であり，$n=3000$までの実際問題を解くことに成功している．彼らの論文の付録には，FORTRAN で書かれたプログラムがついている．このプログラムは /contributed/pardalos から得られる． ここには，Pardalos のグループによる凹関数の最適化による下界を組み込んだ分枝限定法[7]のプログラムと最新のサーベイもある．また，Applegate と Johnson によるC-言語で記述された効率的な最適化手法と近似解法のプログラムも/solvers から得ることができる．

最大クリーク問題に対する数値実験は，通常ランダムグラフを用いて行われる．ここで，ランダムグラフ$G_{n,p}$とは，点の数$n(=|V|)$，定数$0\leq p\leq 1$を与えたとき，すべての2点間に独立に確率pで枝を引くことによって構成されるグラフである．ランダムグラフ$G_{n,p}$に含まれるサイズ$k(=|C|)$のクリークの数の期待値は，k点部分集合の数がn個からk個を選ぶ組合せの数であり，k点のランダムグラフがクリークになる確率が$p^{k(k-1)/2}$であることから

$$ {}_nC_k p^{k(k-1)/2} \qquad (1) $$

となることがわかる．したがって，$G_{n,p}$に対する最大のクリークサイズの推定値（確率的推定値）は(1)が1以上になる最大のkの値となる[4]．たとえば，後で比較に用いる$n=4000$, $p=0.75$のランダムグラフでは，$k=24$, 25, 26, 27のときのクリークの数の期待値を式(1)によって計算すると，それぞれ7377.9, 111.3, 1.21, 0.01個である．したがって，サイズが24や25のクリークは簡単に見つかるが，サイズが26になると探すのは容易ではなくなり，サイズが27になると非常に高い確率で存在しなくなる．

最近では，最大クリーク問題に対するアルゴリズムの評価を行なうには，ランダムグラフによる実験だけでは不完全であるというのが，研究者の間のコンセンサスである．DIMACS では，最大クリーク問題の応用から生まれた，いろいろなタイプの構造をもったグラフを/benchmarks で配布している． 例としては，符号理論から生まれたHamming グラフや Johnson グラフ，幾何学の未解決問題を解くために生まれた Kellerグラフ，信頼性への応用から生まれたC-Fat Ringグラフなどがある．開発したアルゴリズムの頑強性を確認するためには，これらのグラフに対する数値実験も必要不可欠である．また，実験用として大きなクリークを隠したランダムグラフの作成法もいくつか提案されている．

表1 文献[1]の実験で得られた値と確率推定値

p	0.25	0.50	0.75	0.90
Clique Size	7	13	25	52
確率的推定値	7	13	26	55

4. 実験結果の分析

読売新聞に掲載されていたアルゴリズムは近似解法である．実験ではすべてランダムグラフに対して行なわれているが，従来の近似解法との比較ではなく，最適化手法と計算時間の比較を行なっている．当然のことであるが，近似解法の評価をするときに，最適解法に比べて速いことだけで満足してはいけない．

近似解の性能は最適値からどの程度離れているかによって評価できる．しかし，問題の規模が大きくなると最適解を求めるのが困難になるので，最適値の代わりとして上述の確率的推定値を用いるのが普通である．しかし，[1]の実験では，「厳密解が求まっていないので，本アルゴリズムを異なる乱数系列で100回試行し，状況的な証拠から求めた推定値をもととして」最適値を推測する方法を用いている．このような方法によって最適解の保証を得ることは難しいことが予想される．おそらく[1]の実験で最適値と推定した値は，本来の最適値とは異なるものであることが推測される．以下に詳しい理由をあげよう．

まず，上述した確率的推定値との差を見てみる．文献[1]に記載されていた$n=400$のランダムグラフに対する実験結果を表1にあげる．$p=0.25$, 0.50のときは確率的推定値と同じ値を算出してはいるが，$p=0.75$の場合は推定値より1だけ，$p=0.90$の場合は3も小さい値しか算出していない．nが小さくpが0.5から離れた場合には，ランダムグラフ内のクリークサイズの推定値は正確ではないが，彼らの言うところの「実験を100回行な

表 2 Applegate と Johnsonの近似解法を1000回試行したときの近似値とその頻度：$n=400$, $p=0.8$, 確率的推定値は31

Clique Size	24	25	26	27	28	29	30
頻度	3	76	338	399	168	14	2

表 3 Applegate と Johnsonの近似解法を1000回試行したときの近似値とその頻度：$n=8000$, $p=0.5$, 確率的推定値は20

Clique Size	15	16	17	18
頻度	26	718	246	10

ったときの状況的な証拠」よりは，根拠のある数値であると考えられる．

文献[1]のアルゴリズムは手元にないので，どの程度の証拠が実験から生まれてくるかは筆者にはわからない．以下では，Applegate と Johnson の行なった類似の実験結果を利用して「状況的証拠」の妥当性について考えてみよう．

Applegate と Johnson は，彼らが作成したランダム化を用いた近似解法(dmclique) を1000回試行したときの度数分布を$n=100$から8000のランダムグラフに対して調べている(結果のすべては/solvers/results.dmcにある)．一例として[1]で解かれた最大規模の問題の近い$n=100$, $p=0.8$の度数分布表をあげる(**表 2**)．1000回の実験による状況証拠から判断すると，最大のクリークサイズは30であることが予想されるが，残念ながら確率的推定値は31である．このような現象は，問題のサイズが大きくなると顕著になる．たとえば，$n=8000$, $p=0.5$の問題の場合(**表 3**)では，確率的推定値が20であるにもかかわらず，1000回の試行で得られた最良値は18である．このように，近似解法を1000回実行したときの最良値より最適値（または確率的推定値）が大きくなる現象が，かなりの頻度で発生することが Applegate と Johnson の結果を観察することによっていえる．すなわち，近似解法を何度も試行したときの「状況証拠」は，アルゴリズムのよさの証拠にはならないのである．

以下では，筆者が作成した Tabu Search を用いた近似解法と比較を行なう．この近似解法は，現在世界中で最良のものというわけではないが，よい解を短時間で算出するアルゴリズムの1つである．実験は[1] と同じ SPARCstation 2で行ない言語はCである．[1]では，実験はすべてランダムグラフに対して行なわれている

表 4 比較実験：$n=400$, $p=0.75$のランダムグラフ，単位は秒，( )内の数値は得られたクリークのサイズ

文献 [1]	Tabu Search
80 (25)	0.28 (25), 46.3 (26)

が，計算時間と得られた解の値の同者が記述されているのは，$n=400$, $p=0.75$の場合だけである．したがって，この問題に対してのみ比較を行なう．結果を**表 4**にあげる．明らかに Tabu Search の方が高速に解を算出していることがわかる．クリークサイズ25を探索するまでの計算時間は200分1の以下であり，さらに，Tabu Search は確率的推定値であるクリークサイズ26を出すことにも成功している．アルゴリズム実現の容易さも比較の対象になるので簡単に触れる．上の実験で用いた Tabu Search のインプリメンテーションは容易でありデータ入力の部分を除けば，わずか100行程度のコードで記述できる．

他のランダムグラフに対する結果を**表 5** に，構造をもった問題に対する結果を**表 6** にあげる．表6の問題は，すべて/benchmarks から得ることができる．もし今後，最大クリーク問題に対する近似解法を作成して比較を行なうときには，この値を目安にして評価することが望ましい．計算に用いたアルゴリズムおよび実験の詳しい内容（パラメータ設定，他のグラフに対する実験結果等）に関する論文は，筆者に連絡をとることによって得ることができる．また，実験に用いたプログラムも研究用としてなら喜んで提供する．

5. おわりに

上で述べたことから考えると，読売新聞に掲載されていた近似解法は，現在までに開発されてきた解法と比べて優れているとはいえず，また数値実験も不完全であると考えられる．組合せ最適化に限らず，ＯＲに関連することがマスコミに掲載されることは，喜ばしいことではあるが，正しい情報を流すように注意を与えることが，学会としての義務であると考えられる．

参 考 文 献

[1] 猪瀬，山田，富田，高橋，近似最大クリークを抽出する確率アルゴリズムの実験的評価．電子情報通信学会秋季大会予稿集，6－3，1993.

[2] S.Arora and S. Safra, Probabilistic checking of proofs;a new characterization of NP.

表 5 ランダムグラフ$G_{n,p}$に対する実験(の一部):計算時間 (CPU の欄) は SPARCstation 2 の CPU time, 単位は秒

n	p	確率的推定値	Clique Size	CPU
100	0.5	9	9	0.01
200	0.5	11	11	0.1
300	0.5	12	12	4.1
400	0.25	7	7	0.72
400	0.5	13	13	5.25
400	0.75	26	25	0.28
400	0.75	26	26	47.01
500	0.25	8	8	3.6
500	0.5	13	13	4.43
500	0.75	27	26	2.97
500	0.75	27	27	62.5
1000	0.5	15	15	3.13
1500	0.5	16	16	162.08

In *Proceedings 33rd IEEE Symposium on Foundations of Computer Science*, pages 2—13, IEEE Computer Society, Los Angeles, CA, 1992.

[3] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. In *Proceedings 33rd IEEE Symposium on Foundations of Computer Science*, pages 14—23, IEEE Computer Society, Los Angeles, CA, 1992.

[4] B, Bollobas, *Random Graphs*, Academic Press, 1985.

[5] R. Carraghan and P. M. Pardalos, An exact algorithm for the maximum clique problem, *Operations Research Letters*, 9: 375—382, 1990.

[6] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, 1979.

[7] P. M. Pardalos and G. P. Rodgers, A branch and bound algorithm for the maximum clique problem, *Computers and Operations Research*, 19: 363—375, 1992.

表 6 応用から生まれたグラフ (DIMACS Benchmarks) に対する実験(の一部), 計算時間(CPU の欄) は SPARCstation 2 のCPU time, 単位秒, 得られたすべての解の Clique Size は最適値またはBest Known Value に一致している.

File Name	$n = \|V\|$	$\|E\|$	Clique Size	CPU
c-fat200-1	200	1534	12	0.01
c-fat200-2	200	3235	24	0.05
c-fat200-5	200	8473	58	0.15
c-fat500-1	500	4459	14	0.05
c-fat500-10	500	46627	126	0.52
c-fat500-2	500	9139	26	0.03
c-fat500-5	500	23191	64	0.23
johnson16-2-4	120	5460	8	0.02
johnson32-2-4	496	107880	16	0.03
johnson8-2-4	28	210	4	0.01
johnson8-4-4	70	1855	14	0.01
keller4	171	9435	11	0.07
keller5	776	225990	27	17.27
keller6	3361	4619898	59	2113.2
hamming10-2	1024	518656	512	7.37
hamming10-4	1024	434176	40	0.56
hamming6-2	64	1824	32	0.03
hamming6-4	64	704	4	0.01
hamming8-2	256	31616	128	0.43
hamming8-4	256	20864	16	0.01
san1000.clq	1000	250500	15	819.18
san200_0.7_1.clq	200	13930	30	5.45
san200_0.7_2.clq	200	13930	18	4.72
san200_0.9_1.clq	200	17910	70	0.28
san200_0.9_2.clq	200	17910	60	2.47
san200_0.9_3.clq	200	17910	44	1.53
san400_0.5_1.clq	400	39900	13	50.62
san400_0.7_1.clq	400	55860	40	57.15
san400_0.7_2.clq	400	55860	30	0.62
san400_0.7_3.clq	400	55860	22	6.52
san400_0.9_1.clq	400	71820	100	17.78
sanr200_0.7.clq	200	13868	18	0.12
sanr200_0.9.clq	200	17863	42	1.5
sanr400_0.5.clq	400	39984	13	8.42
sanr400_0.7.clq	400	55869	21	7.50
p_hat300-1.clq	300	10933	8	3.13
p_hat300-2.clq	300	21928	21	0.12
p_hat300-3.clq	300	33390	36	0.92
p_hat500-1.clq	500	31569	8	0.43
p_hat500-2.clq	500	62946	36	1.03
p_hat500-3.clq	500	93800	50	8.18
p_hat700-1.clq	700	60999	9	7.07
p_hat700-2.clq	700	121728	33	0.37
p_hat700-3.clq	700	183010	62	0.82
p_hat1000-1.clq	1000	122253	10	12.62
p_hat1000-2.clq	1000	244799	27	0.62
p_hat1000-3.clq	1000	371746	68	17.15
p_hat1500-1.clq	1500	284923	10	70.02
p_hat1500-2.clq	1500	568960	24	6.60
p_hat1500-3.clq	1500	847244	94	5.73