A-004

大規模な最大クリーク問題に対する k-opt 局所探索法の性能評価

Performance Analysis of k-opt Local Search for Large Maximum Clique Problems

濱本　明宏[†]　　片山　謙吾[†]　　成久　洋之[†]
Akihiro HAMAMOTO　　Kengo KATAYAMA　　Hiroyuki NARIHISA

あらまし　我々は既に，最大クリーク問題 (maximum clique problem, MCP) に対して，可変深度探索のアイデアに基づく k-opt 局所探索法を提案している．一般にアルゴリズムの性能の評価は，主としてベンチマークグラフやランダムグラフを対象とした結果にゆだねられることが多い．k-opt 局所探索法は DIMACS のベンチマークグラフに対して，既に報告されている最先端なメタ戦略アルゴリズムの性能を上回るもしくは匹敵する性能を有することが明らかにされている．本論文では，既存の研究で対象とされた問題規模よりもはるかに大規模な 30000 節点までのランダムグラフを対象に k-opt 局所探索法の性能を評価し，平均的に良質な解を算出することを示す．

キーワード　局所探索，可変深度探索，最大クリーク問題，組合せ最適化

1. はじめに

節点 (vertex) の集合 $V = \{1, 2, ..., n\}$，枝 (edge) の集合 $E \subseteq V \times V$ からなる無向グラフを G= (V,E) とし，V の部分集合 S による生成部分グラフを $G(S) = \{S, E \cap S \times S\}$ とする．すべての節点対の間に枝をもつ無向グラフ $G = (V, E)$ を完全グラフ (complete graph) という．V の部分集合 C による生成部分グラフ $G(C)$ において，すべての節点対に枝をもつとき，C をクリーク (clique) という．最大クリーク問題 (maximum clique problem, MCP) とは，与えられたグラフから節点数最大のクリークを求める問題である．

MCP は，NP 困難であり，組合せ最適化問題における基本問題の一つである．また，種々の実用的応用も図られており ([5] の 7 章 Applications)，多くの研究者により厳密解法やヒューリスティックにもとづく解法の研究が行われている．

近年，MCP の実用的応用として，通信分野や生物情報学を対象とした大規模な問題例に対する研究結果が報告されている [1, 2]．文献 [1] では，遠距離通信の実データに対して最大クリーク問題に対する近似解を算出する greedy randomized adaptive procedure(GRASP) が提案されている．この実問題はランダムグラフと異なるものの節点数が約 5377 万と非常に多いが，枝数が約 17 億と枝密度にして 0.17×10^{-6} と極めて疎なグラフである．文献 [2] では，富田らによって提案された分枝限定法 (branch-and-bound method) に基づく効率的な厳密解法 MCQ を採用し実用的応用を試みている．しかしながら，MCQ のような分枝限定法で適用可能な問題サイズは，疎なグラフに対しては 15000 節点程であり，密なグラフに対しては 1000 節点程である．節点数が多い密なグラフに対しては，問題サイズの大規模化に伴い莫大な計算時間を必要とすることが知られている．

文献 [1, 2] で対象とされている現実の問題は節点数は多いが疎なグラフである．現実の問題がそのようなグラフ的な特徴を持っている場合は，それに適した解法を採用するのが望ましい．しかし，近年，我々が取り扱うデータの数や種類は急激に増加傾向にあり，それに伴い問題のサイズは爆発的に拡大することが十分予想できる．

そのような状況を踏まえ，近年，メタ戦略 (metaheuristics) にもとづくアルゴリズムが注目されている [15]．メタ戦略とは，局所探索法 (local search, local improvement) などの基本戦略よりも多少時間がかかっても，より良質の解を求めるような解法の一般的な枠組みを与えるものである．一般に，局所探索法はメタ戦略に属する殆どのアルゴリズムでも多くの場合に採用されるため，高性能な局所探索法を開発することは極めて重要であり，メタ戦略の研究に対して重要な指針を与える [9]．実際に，高性能な局所探索法を組み込んだメタ戦略は，低性能な局所探索法を組み込んだものよりも，より良質な近似解を算出することが報告されている [7, 9]．

我々は MCP に対して，Lin と Kernighan による可変深度探索 (varable depth search, VDS) [11, 12] のアイデアに基づく k-opt 局所探索法を既に提案している．文献 [10] では，MCP に対する k-opt 局所探索法の有効性を示すために，DIMACS [8] のベンチマークグラフ群のそれぞれに対して，k-opt 局所探索法の適用を試みた．その結果，k-opt 局所探索法は，MCP に対する最先端のメタ戦略アルゴリズムである，Marchiori により提案された遺伝的局所探索法 (genetic local search) や反復局所探索法 (itetated local search) [13] よりもテストしたすべての問題例に対して，平均的により良好な近似解をより短時間に算出可能であった．さらに，Battiti と Protasi により提案された reactive local search (RLS) [3] との比較を行った．RLS はタブー探索法 (tabu search) に基づく，MCP に対する最強のメタ戦略の 1 つであることが広く知られており，DIMACS のほとんどの問題例に対して，高品質な解を非常に短い時間で算出可能である．k-opt 局所探索法は，得られる解の平均値において，RLS にひけを取らない性能であることを示した．

以上をふまえ，本論文では，既存研究では扱われなかったような 30000 節点までの大規模な問題例に k-opt 局所探索法を適用しその性能を評価する．その結果，実用的な時間内に，高品質な解を算出可能であることを示す．

2. add move による 1-opt 局所探索法

MCP における実行可能領域 F は与えられた無向グラフ $G(V, E)$ のすべてのクリークの集合 X である．MCP の目的関数 f はクリーク $x \in X$ のサイズ $f(x) = |x|$ である．MCP における最も素朴な近傍 $N(x)$ とは 1 つの

[†]岡山理科大学工学部情報工学科
Department of Information and Computer Engineering,Okayama University of Science

節点を加えたり，取り除いたすることによって x から得られるすべてのクリークである．前者の近傍操作を **add move**，後者の近傍操作を **drop move** を呼ぶ．また，2 つの近傍操作を総称して **1-opt move** と呼ぶ．次節で述べる k-opt 局所探索法は add move と drop move の両方を適用するが，本節では add move だけの 1-opt 局所探索法について簡単に考察する．

グラフ $G(V,E)$ と拡大可能な初期クリーク x が与えられたとする．局所探索法は一般により大きいクリークを構成するために，各反復ごとに現在のクリークのサイズを 1 つずつ拡大することを試みる．現在のクリークを拡大する素朴な操作は，拡大可能な節点集合からランダムに 1 つの節点を 1 つ選択することである．しかしながら，そのような操作で結果的に得られるクリークは質の悪いことが既によく知られている [3]．

現在のクリークのサイズを 1 つ拡大する 1 つの節点を選択する場合には，節点の次数を考慮する方が良いと考えられる．なぜならば，高い次数をもつ節点は一般により大きなクリークを構成する可能性が高いからである．

我々の局所探索法では，与えられたグラフの節点の次数を考慮するのではなく，現在のクリークを拡大可能な節点集合からなる生成部分グラフにおける最大次数をもつ 1 つの節点を選択する．最大次数をもつ節点が複数ある場合には，その中からランダムに 1 つ選択する．その後，次の反復処理のために現在のクリークを拡大可能な節点集合とその節点集合からなる生成部分グラフに含まれる各節点の次数を更新する．このような add move による操作は現在のクリークを拡大することができる節点集合が空集合になるまで繰り返される．つまり，これ以上解が改善できなくなるまで繰り返される．

3. MCP に対する k-opt 局所探索法

可変深度探索 (variable depth search, VDS) は，巡回セールスマン問題 (traveling salesman problem, TSP) やグラフ分割問題 (graph partitioning problem, GPP) に対して Lin と Kernighan により提案された巧妙な近傍探索のアイデアである [11, 12]．TSP と GPP に対する彼らの両局所探索法は現在もなお解改善法として優れた探索性能を誇っている．また，他の組合せ最適化問題に適用され成功を収めた例として [9, 16] が挙げられる．以下，MCP に対する VDS のアイデアに基づく我々の k-opt 局所探索法 [10] について簡単に述べる．

MCP に対する VDS に基づく我々の k-opt 局所探索法の近傍は，現在のクリークに上記の 1-opt move を連続的に適用して得られる解の集合を 1 つの大きな近傍と定義している．前節で述べたように 1-opt move は 2 つの近傍操作 (add move と drop move) の総称である．したがって，k-opt 局所探索法の各反復では連続的に add move と drop move が適切に切り替わりながら適用される．k-opt 局所探索の探索過程において，現在のクリークに add move が適用される処理では，前述のように現在のクリークを拡大することができる節点の次数を考慮する．一方，drop move の操作は現在のクリークから 1 つの節点を取り除く．現在のクリークから 1 つ節点を取り除くとクリークのサイズは 1 つ小さくなるが，1 つ小さくなったクリークに対して新しい拡大可能な節点集合が発生する．drop move が適用される処理では，このように新しく発生する拡大可能な集合のサイズが最大となるような節点を現在のクリークから 1 つ選び，現在のクリークから取り除く．k-opt 局所探索は 1-opt move を連続的に適用して得られる解の集合内に改善解を発見できなくなるまで探索を進め，得られた最良解を出力し探索を終了する．なお，k-opt 局所探索法はパラメータを一切使用してないため，煩わしいパラメータの設定が必要がない．詳細は [10] を参照されたい．

4. 大規模な問題例に対する k-opt 局所探索法の性能の評価実験

巡回セールスマン問題のように非常に研究が進んでいる組合せ最適化問題では，ランダムに作成された大規模な問題例に対して近似解法の性能評価が行なわれる [7]．より大規模な問題においては，一般に最適解自体がわからないことが多い．そのため，文献 [7] では，理論的な最適解の下界値に対する解の質と計算時間によって近似解法の性能を評価する方法が採用されている．

最大クリーク問題は巡回セールスマン問題と同様に代表的な組合せ最適化問題である．しかしながら，大規模な最大クリーク問題に対する近似解法の性能についての研究やその報告は我々が知る限りない．

本論文は，大規模な最大クリーク問題に対する近似解法の性能評価のはじめての試みであり，最大クリーク問題に対する研究を大きく前進させることを目的とする．その目的のために，後述するランダムに作成された 30000 節点までの大規模な問題例に対する k-opt 局所探索法の探索性能を検証し，k-opt 局所探索法が高品質な解を算出可能であることを示す．

4.1 ランダムグラフと最大クリークサイズの見積

節点の集合 V が与えられたとき，すべての節点対について，それぞれの節点対の間に確率 p で一様に枝が張られたグラフをランダムグラフといい，$G_{n,p}$ で表す．

以下では，ランダムグラフに対する理論的事実を利用し，パラメータ n と p にもとづく最大クリークのサイズの見積に関して議論する．

ランダムグラフに対する研究は広く行なわれており，多くの理論的事実が報告されている．Bollobas と Erdos [4] は，ランダムグラフ $G_{n,p}$ に最も高い確率で出現する最大クリークのサイズ $X_n(G)$ は，$b = 1/p$ とすると，

$$X_n(G) = 2\log_b n - 2\log_b \log_b n + 2\log_b(\frac{1}{2}e) + 1$$

となり，$X_n(G)$ の取りうる範囲 X_n は，

$$[Z_n(G) - 2\frac{\log\log n}{\log n \log b}] \leq X_n \leq [Z_n(G) + 2\frac{\log\log n}{\log n \log b}]$$

となることを証明している．

上記のランダムグラフに対する理論的事実を考察すると，$X_n(G)$ の取りうる範囲 X_n は n と p によって決定することから，n と p から X_n の下界値および上限値を得ることが可能であり，n が増大すればするほど，下界値と上限値が近づきより精度の高い最大クリークのサイズ（最適解の値）の見積もりが可能となる．また，p の

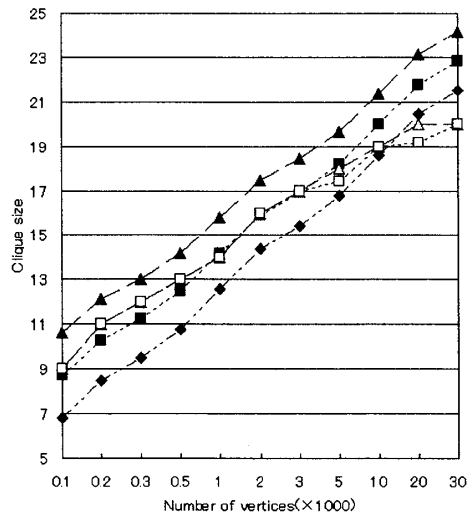

図 1: 算出された解と理論的な最適解

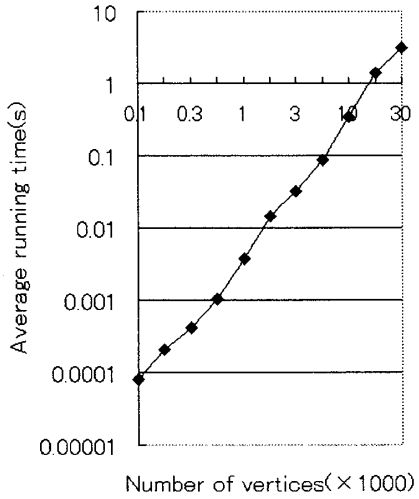

図 2: k-opt 局所探索法の 1 回あたりの平均計算時間

値が 0.9 のような密なグラフでは節点数が大きくとも最適解の見積もりは煩雑になる．

4.2 実験方法

文献 [7] では，理論的な最適解の見積もりにもとづいて，大規模な巡回セールスマン問題に対する近似解法の性能の評価実験が行なわれている．本実験でも同じような方法で，大規模な最大クリーク問題に対する近似解法の性能評価を行なう．

なお，本論文で扱う全ての問題例は，p を 0.5 とし，節点数 n={100, 200, 300, 500, 1000, 2000, 3000, 5000, 10000, 20000, 30000} とした計 11 問のランダムグラフを問題例とする．

本実験では，上記の全 11 問それぞれに対して，異なる初期解に k-opt 局所探索法を適用することを繰り返した．我々はこのような多スタート法 (multi-start method) を HP ワークステーション xw6000 (Xeon 2.8GHz) 使用のもとで実行した．アルゴリズムは C によりコード化し，最適化オプション-**O2** を使用した **gcc** コンパイラでコンパイルした．なお，多スタート法の試行は 5 回である．

k-opt 局所探索法を組み込んだ多スタート法は以下のように実行した．多スタート法の各試行ではそれぞれ異なった乱数の種を使用し，与えられたグラフの節点数を n とすると，異なる n 個の各初期解からスタートする k-opt 局所探索法を繰り返し実行した．ただし，多スタート法の計算時間が **cpu time** で 3600 秒以上費やしたとき，例え n 回の k-opt 局所探索法が実行されていない場合であっても，処理を打ち切った．このような計算打ち切り時間を設けた理由は，問題のサイズが大規模になればなるほど 1 回の局所探索法に費やす計算時間が増大し，指定した回数の局所探索法を実行するにはかなりの計算時間を必要とすることが予備実験から明らかになったためである．多スタート法が最終的に算出する解は，上述の終了条件のいずれかを満たすまでに算出された最良解を結果として出力する．

k-opt 局所探索法に与える初期解の生成法は非常に単純である．1 つの節点を初期解とする方法であり，与えられたグラフの節点の次数が高い節点を先に初期解とする．ただし，1 試行あたりの各初期解はそれぞれ異なるものとする．したがって，各試行において，k-opt 局所探索法に与える初期解は同じものとなる．しかしながら，k-opt 局所探索法は 3. 章で述べたように 1-opt move の際にランダム選択が施されるため，同じ初期解が与えられたとしても最終的に得られる解や探索は異なったものとなる．

4.3 結果と考察

図 1 は，p=0.5 の場合について所定の節点数 (Number of vertices) のランダムグラフそれぞれに対して，k-opt 局所探索法を組み込んだ多スタート法により得られた結果である．図 1 の横軸は節点数を 1/1000 に縮小して示し，縦軸に上述の実質的最適解 $X_n(G)$，その下界，その上限をそれぞれ■，◆および▲でプロットし，本多スタート法を 5 回試行して得られた最良解および平均の値をそれぞれ△および□でプロットしてある．図 1 より，k-opt 局所探索法を組み込んだ多スタート法は節点数 100 から節点数 10000 までの 11 問中 9 問の問題例までは，4.1 節で述べた見積りから最適解に極めて近い高品質な近似解が得られたものと考えられる．節点数 20000 および 30000 の問題例に対しては，得られる解の質が若干悪くなる傾向が観測できる．

比較のため，2. 節で述べた add move だけの 1-opt 局所探索法の実験結果についても以下に簡単に述べる．1-opt 局所探索法の実験方法は k-opt 局所探索法とまったく同じ方法である．1-opt 局所探索法は節点数 3000 までの問題例に対しては k-opt 局所探索法と同質の解を算出可能であった．しかしながら，それ以上に節点数が多い問題例については，1-opt 局所探索法の結果は k-opt 局所探索法の結果よりも平均的に悪くことが観測された．具体的には，1-opt 局所探索法の得られた解の平均値は節点数 5000，10000，20000 および 30000 のときそれぞれ 17.0，18.2，19.0，20.0 であった．

k-opt 局所探索法による多スタート法の計算打ち切り時間 3600 秒によって計算が打ち切られた問題例は節点数が 20000 および 30000 の 2 問だけであった．そのときの k-opt 局所探索法の平均適用回数はそれぞれ 2584.2 回，1178.60 回であり，節点数 n に対する割合はそれぞれ 12.9%，3.93%であり，他の 9 問の場合よりも局所探索法の回数が極端に少ない．

そこで，得られる解の質が悪かった 2 つの問題例に対して，計算打ち切り時間を 5 倍の 18000 秒 (5 時間) まで許容し，本実験方法と同じ要領で k-opt 局所探索法を組み込んだ多スタート法を 5 回試行した．その結果，5 回の試行で得られた最良解の最大値，平均値，各試行で最良解を得るまでの平均計算時間および k-opt 局所探索法の平均適用回数は，節点数 20000 の問題例に対しては，各々20，20.0，5047.8 秒および 13132.0 回であり，節点数 30000 の問題例では，各々21，20.2，3416.6 秒および 5849.4 回であった．

この結果から，k-opt 局所探索法が適用された回数は指定された回数 n に対して十分ではないが，より長い計算時間を許容することでより良好な解が平均的に得られることが観測された．

また，図 2 は全 11 問の問題例それぞれに対する k-opt 局所探索法の単位実行あたりの平均計算時間である．図 2 より，p=0.5 の場合，節点数 100 から節点数 10000 までの問題例に対しては非常に高速である．節点数が 20000，30000 といった大規模な問題例に対する 1 回の局所探索法の計算時間は 1 秒以上となるが，そのような大規模なグラフに対しても十分に実用的な時間と言え，上述の結果をふまえて良質なクリークを算出可能であると言える．

5. 結 論

本論文では，最大クリーク問題 (MCP) における既存研究の多くで適用された問題サイズをはるかに上回る大規模な MCP の問題例を準備し，DIMACS のベンチマーク問題例に対し成功を収めた可変深度探索 (VDS) に基づく k-opt 局所探索法の性能を評価した．大規模な問題例に対する k-opt 局所探索法の性能評価は，グラフ理論にもとづく理論的定理を応用し求めた最適解の理論値に基づいて行った．その結果，k-opt 局所探索法が節点数 10000 までの問題例に対して，実質的な最適解もしくはそれに近い高品質な近似解を頻繁に短時間で算出可能であることを示した．さらに，節点数が 20000，30000 のより大規模な問題例に対しては，より長い計算時間を許容することでより良好な解が平均的に得られることを示した．このことから，MCP に対する k-opt 局所探索法の性能は主要なメタ戦略の枠組みを有しないにも関わらず，比較的短時間で精度の高い解を算出可能であるといえる．

参考文献

[1] J. Abello, P.M. Pardalos, and M.G.C. Resendė, "On Maximum Clique Problems In Very Large Graphs," AT&T Labs Research Technical Report: TR 98.32.1.

[2] D. Bahadur, T. Akutsu, E. Tomita, T. Seki, and A. Fujiyama, "Point matching under non-uniform distortions and protein side chain packing based on efficient maximum clique algorithms, " Genome Informatics, No.13, pp.143–152, 2002.

[3] R. Battiti and M. Protasi, "Reactive local search for the maximum clique problem," Algorithmica, vol. 29, no. 4, pp. 610–637, 2001.

[4] B. Bollobas and P. Erdos, "Cliques in random graphs," Math. Proc. Cambridge Philos. Soc. 80, pp.419–427,1976

[5] I.M. Bomze, M. Budinich, P.M. Pardalos, and M. Pelillo, "The maximum clique problem," In D.-Z. Du and P. M. Pardalos, editors, Handbook of Combinatorial Optimization, Kluwer Academic Publishers, Boston, MA, 1999.

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

[7] D.S. Johnson and L.A. McGeoch, "The traveling salesman problem: A case study," Local Search in Combinatorial Optimization, John Wiley & Sons, pp. 215–310, 1997.

[8] D.S Johnson and M.A. Trick, (Eds.): "Cliques, Coloring, and Satisfiability," DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol.26, American Mathematical Society, 1996.

[9] 片山謙吾, 成久洋之, "大規模な最大多様性問題に対する遺伝的局所探索," 情報処理学会論文誌：数理モデル化と応用, vol. 45, no. SIG 2 (TOM 10), pp. 99–109, 2004.

[10] K. Katayama, A. Hamamoto, and H. Narihisa, "Solving the Maximum Clique Problem by k-opt Local Search," Proc. of the 2004 ACM Symposium on Applied Computing , vol. 2, pp. 1021–1025, 2004.

[11] B.W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," Bell System Technical Journal, vol. 49, pp. 291–307, 1970.

[12] S. Lin and B.W. Kernighan, "An effective heuristic algorithm for the traveling salesman problem," Operations Research, vol. 21, pp. 498–516, 1973.

[13] E. Marchiori, "Genetic, iterated and multistart local search for the maximum clique problem," Applications of Evolutionary Computing, Springer, LNCS 2279, pp. 112–121, 2002.

[14] E. Tomita and T. Seki, "An Efficient Branch-and-Bound Algorithm for finding a Maximum Clique," Discrete Mathematics and Theoretical Computer Science, LNCS 2731, pp. 278-289, 2003.

[15] 柳浦睦憲, 茨城俊秀：組合わせ最適化–メタ戦略を中心として–. 朝倉書店 (2001)

[16] M. Yagiura, T. Yamaguchi and T. Ibaraki, "A variable depth search algorithm for the generalized assignment problem," Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization, (S. Voss, S. Martello, I.H. Osman and C. Roucairol, eds.), Kluwer Academic Publishers, pp. 459–471, 1999.

付 録 MACHINE BENCHMARKS

最後に参考として，本実験で使用した計算機のベンチマーク結果を示す．以下の表に示すのは，DIMACS プロジェクトにおいて公開されているアルゴリズム dfmax のソースコードをコンパイルし実行した計測時間 (秒) の結果である．

Type of Machine : Xeon CPU 2.8GHz
Compiler and flags used : gcc -O2

r100.5	r200.5	r300.5	r400.5	r500.5
2.00×10^{-3}	5.80×10^{-2}	0.487	2.42	13.1