

## 最大クリーク問題に対する $k$ -opt 局所探索法の解析

貞松 政史・片山 謙吾\*・南原 英生\*・成久 洋之\*

岡山理科大学大学院工学研究科情報工学専攻

\*岡山理科大学工学部情報工学科

(2006年10月2日受付、2006年11月6日受理)

### 1. まえがき

頂点 (vertex) の集合  $V = \{1, \dots, n\}$  とそれらの頂点の対を両端とする無向辺 (undirected edge) の集合  $E \subseteq V \times V$  が与えられた時,  $G = (V, E)$  を無向グラフという. 特に, 全ての2頂点間に1つの辺が存在する無向グラフを完全グラフという.  $V$  の部分集合  $V' \subseteq V$  による誘導部分グラフ  $G(V') = (V', E \cap V' \times V')$  が完全グラフの時, すなわち,  $\forall i, j \in V', i \neq j$  に対して  $(i, j) \in E$  である時,  $V'$  をクリークと呼ぶ. 最大クリーク問題 (Maximum Clique Problem, MCP)<sup>1)</sup> とは, 与えられたグラフ  $G$  に含まれるクリーク  $K$  の中で, 次の目的 (評価) 関数

$$f_{\text{MCP}}(K) = |K|$$

を最大にするクリークを求める問題である.

MCP は, 通信ネットワーク, 符号理論, 並列計算, パターン認識等の分野にあらわれる実用上重要な組合せ最適化問題として知られており<sup>1)</sup>, NP-困難<sup>2)</sup> である. 従って, 多項式時間で厳密解を算出するアルゴリズムは存在しないであろうと考えられている. また, MCP の良質な近似解を得ることすら NP-完全のクラスに準じるほど困難であることが知られており<sup>3)</sup>, その他にも MCP の難しさを示す否定的な報告がなされている<sup>4)5)</sup>.

最近我々は, MCP に対して, 可変深度探索 (Variable Depth Search, VDS)<sup>6)7)</sup> に基づく  $k$ -opt 局所探索法 ( $k$ -opt Local Search, KLS)<sup>8)</sup> を提案した. KLS は, 各反復において, 現在のクリーク (解) を対象に複数個の頂点を連鎖的に追加もしくは削除する操作 (それぞれを Add 移動, Drop 移動と呼ぶ) により構成され, 現在の解からそれらの操作によって生成可能な解の集合を改めて近傍と捉えることで, 局所探索を行うアルゴリズムである. その特長は, 比較的大きな近傍を効率よく探索する巧妙な近傍探索処理を施しているにも関わらず, パラメータ値の設定を必要としないシンプルさである. KLS の性能を検証するために, 第2回 DIMACS Implementation Challenge Workshop<sup>9)</sup> で利用されたベンチマークグラフに適用した結果, 多くのグラフにおいて, そのワークショップで報告されていた最適解と同等以上の良質な解の算出に成功した. 更に我々は, MCP に対する高性能なメタ戦略アルゴリズムとして知られている, Marchiori による遺伝的局所探索法と反復局所探索法<sup>10)</sup>, タブー探索法にもとづく Battiti らの Reactive Local Search<sup>11)</sup> との比較を通して, それらと同等もしくは平均的により高品質の近似解を比較的短時間で算出できることを示した.

上述のように, KLS は, 他の近似解法との比較においてその性能を検証しているが, 探索の動作特性など, 内部的な解析については, 未だ十分になされていない. 本論文では, MCP に対する KLS の解析の一端として, KLS における近傍探索処理を打切る操作の検討, および与えられたグラフ中での KLS の探索範囲について調査する.

### 2. MCP に対する $k$ -opt 局所探索法

本節では, 我々が既に提案している MCP に対する  $k$ -opt 局所探索法 (KLS) の概要などに関して述べる. なお, KLS の詳細については文献<sup>8)</sup> を参照されたい.

```

k-opt-Local-Search(CC,PA,OM,degG(PA))
begin
1  repeat
2    CCprev:=CC, D:=CCprev, P:={1,...,n}, g:=0, gmax:=0;
3    repeat
4      if |PA ∩ P| > 0 then // Add Phase
5        find a vertex v with maxv ∈ {PA ∩ P}{degG(PA ∩ P)(v)};
6        if multiple vertices with the same maximum degree are found
7          then select one vertex v among them randomly;
8        CC := CC ∪ {v}, g := g + 1, P := P \ {v}
9        if g > gmax then gmax := g, CCbest := CC;
10       else //Drop Phase (if{PA ∩ P} = ∅)
11         find a vertex v ∈ {CC ∩ P} such that the resulting |PA ∩ P| is maximized;
12         if multiple vertices with the same size of the resulting |PA ∩ P| are found
13           then select one vertex v among them randomly;
14         CC := CC \ {v}, g := g - 1, P := P \ {v};
15         if v is contained in CCprev then D := D \ {v};
16       endif
17       update PA, OM, and degG(PA)(i), ∀i ∈ PA ∩ P;
18     until D = ∅;
19     if gmax > 0 then CC := CCbest else CC := CCprev;
20   until gmax ≤ 0;
21   return CC;
end;

```

図1 MCP に対する  $k$ -opt 局所探索法の擬似コード

## 2.1 KLS の基本アルゴリズム

KLS は、可変深度探索 (VDS) のアイデアにもとづいている。VDS とは、与えられた解に対して比較的小さな近傍操作を連鎖的に適用することで到達可能な解の集合を改めて大きな近傍と捉える近傍探索のアイデアである。KLS は、各反復において、現在のクリーク (解) から複数の頂点を連鎖的に追加および削除する操作 (それぞれ Add 移動, Drop 移動と呼ぶ) により構成され、現在の解からそれらの操作によって生成可能な解の集合を改めて近傍と捉えることで、局所探索を行うアルゴリズムである。

KLS の擬似コードを図1に示す。KLS は外ループ (Line1-18) と内ループ (Line3-16) の処理を有する。以下において、外ループに関しては「反復」、内ループに関しては「繰り返し」と呼び区別する。

まず、本論文で頻出する、図1の中の重要な記号を説明する。 $CC^{(l)}$  は内ループの繰り返し  $l$  の時点における解 (クリーク) である。 $PA^{(l)}$  は  $CC^{(l)}$  の全頂点到隣接する、 $CC^{(l)}$  に追加可能な頂点の集合

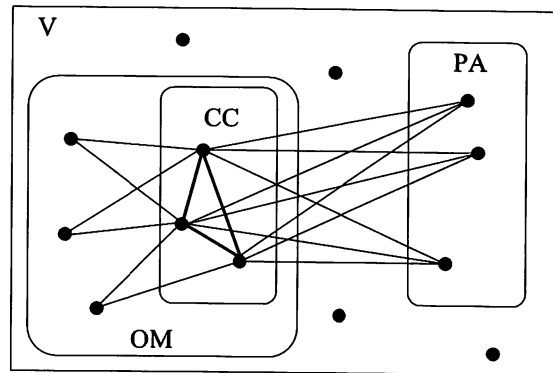
$$PA^{(l)} = \{v : v \in (V \setminus CC^{(l)}), (v, i) \in E, \forall i \in CC^{(l)}\}$$

である。 $OM^{(l)}$  は  $PA^{(l)}$  の定義を若干緩和した1辺不足集合と呼ぶ辺集合

$$OM^{(l)} = \{(v, i) : v \in V, i \in CC^{(l)}, (v, i) \notin E, (v, i) \in E, \forall j \in CC^{(l)}, j \neq i\}$$

である。なお、 $OM^{(l)}$  は  $CC^{(l)}$  に含まれる頂点群の中のいずれか一つの頂点  $i \in CC^{(l)}$  だけに辺が存在しない頂点の集合と捉えることもできる (なお、 $CC \subseteq OM$ ) (図2参照)。 $deg_{G(PA^{(l)})}$  は  $PA^{(l)}$  により誘導される部分グラフ  $G(PA^{(l)})$  内の各頂点  $v \in PA^{(l)}$  の次数である。

次いで、KLS の各反復における基本アルゴリズム ( $k$ -opt 局所探索処理) について簡潔に説明する。まず、与えられた初期クリーク (初期解)  $CC^{(0)}$  を対象として、複数の頂点を連鎖的に追加する操作 (Add 移動操作) および削除する操作 (Drop 移動操作) により到達可能な近傍解の集合  $CC^{(1)}, \dots, CC^{(k)}, \dots, CC^{(r)}$  を得る。(その生成途中では、移動候補頂点集合  $P$  (Line2) を利用することで、追加または削除された頂点

図2  $CC$ ,  $PA$  および  $OM$  の集合の一例

は再び追加・削除されることはない). その近傍解の集合から最良解  $CC^{(k)}$  ( $1 \leq k \leq r$ ) を選び (Line8), 次反復の初期クリーク  $CC^{(0)} := CC^{(k)}$  とする (Line17). KLS は常に実行可能領域を探索空間としており, 各反復の初期クリークに応じて,  $k$ -opt 近傍のサイズ (上記の  $r$  に対応) が適応的に変動する.

上記の  $k$ -opt 近傍探索処理は, Add 移動操作を施す「Add フェーズ」 (Line5-8) と Drop 移動操作を施す「Drop フェーズ」 (Line10-13) の二つのフェーズで構成される. 以下では, それぞれのフェーズにおいて核となる頂点選択方式 (Line5-6 および Line10-11) について記述する.

### 2.1.1 Add 移動頂点選択方式

$PA^{(l)} \cap P$  の中で次数  $deg_{G(PA^{(l)}(v))}$  が最大の頂点  $v$  を選択する. 但し, 同じ最大次数の頂点が複数個存在する場合は, それらの頂点群からランダムに一つの頂点を選択する.

### 2.1.2 Drop 移動頂点選択方式

$CC^{(l)}$  の中で,  $l+1$  の時点で得られる  $|PA^{(l+1)}|$  が最大となる頂点  $v$  を選択する. 但し,  $|PA^{(l+1)}|$  を最大とする頂点が複数個存在する場合は, それらの頂点群からランダムに一つの頂点を選択する.

## 3. KLS の解析に関する検討事項

本節では, 本論文の目的である, KLS の解析における 2 つの検討事項について記述する. まず, KLS における近傍探索処理の打ち切り操作について 3.1 節で記述し, 次いで KLS の探索範囲について 3.2 節で記述する.

### 3.1 近傍探索処理の打ち切り操作

KLS の処理における内ループの終了条件は  $D = \emptyset$  である (図 1 Line16).  $D$  は外ループの各反復における初期クリーク, すなわち前回の反復によって得られた最良解 ( $CC_{prev}$ ) で初期化されている (Line2). 前回の反復によって得られた最良解のクリークサイズが大きくなるほど,  $D$  のサイズが大きくなるため, 内ループの繰り返しの回数も比例して増加する可能性が高い. しかし, 繰り返しの回数が増加しても, KLS では移動候補頂点集合  $P$  (Line2) を利用しているため, 一度移動操作を行った頂点は, その反復内で再び移動操作を行うことはできない. そのため, 後半の繰り返しでは移動可能な頂点群が大幅に減少しており, 改善解が発見される可能性は低いと考えられ, この部分は KLS の探索において効率的でない. そこで, 探索の効率化を図るために, 繰り返しをある程度の回数で打切る操作を考える. 本論文では,  $D$ ,  $CC_{prev}$  のサイズを基準に以下の 2 パターンの打ち切り操作を提案する.

- 打ち切り 1:  $|D|$  が  $|CC_{prev}|$  の  $\frac{1}{2}$  になったとき, 内ループを終了する.
- 打ち切り 2:  $|D|$  が  $|CC_{prev}|$  の  $\frac{2}{3}$  になったとき, 内ループを終了する.

3.1.1 実験 I の詳細

上述 2 パターンの打切り操作の効果を検証するために実験を行う。打切り操作を行わない従来の KLS と 2 パターンの打切り操作 (打切り 1 と打切り 2) をそれぞれ有する 2 タイプの KLS の計 3 タイプの KLS を比較検討する。対象とするグラフは、MCP の標準的なベンチマーク問題としてよく知られる DIMACS ベンチマークグラフ (最大頂点数 4000, 最大辺数 5506380) から大規模もしくは厳密解の算出が困難な 37 グラフを対象とする。

本実験では、以下のマルチスタート法を採用する。与えられるグラフ  $G$  に含まれる 1 頂点をクリークと見なす時、 $|V| (= n)$  個の異なる初期クリークが生成可能であることから、 $n$  個の異なる (質の悪いクリーク) 解をそれぞれ初期解とする。従ってマルチスタート法の 1 試行あたり、 $n$  回の KLS が実行され、 $n$  個の局所最適解が得られる。KLS に与える初期クリークは、各頂点の度数にもとづき予めソートし、度数の高い頂点から非増加順に与える。KLS の純粋な性能を検証するためには、マルチスタート中、既に得られたクリークの情報を用いた後の KLS の実行の際に利用すべきではないため、KLS の各実行はそれぞれ独立させ、得られるクリーク群から最良解を選び出力する。なお、同じクリーク重みの最良解が複数個得られる場合は、最初に得られる最良解を出力する。3 タイプの KLS をそれぞれ有するマルチスタート法を各 10 回試行する。

全ての KLS は C 言語によってコード化し、使用コンパイラは、最適化オプション-O2 を付加した gcc (Ver.4.1.1) である。全ての実験は、Hewlett-Packard 社の計算機 HP xw4300 Workstation CPU:Pentium4 3.4GHz, 4GB RAM, OS:Fedora Core 5 上で行う。

3.1.2 実験 I の結果

結果を表 1 に示す。DIMACS benchmarks の欄には、問題例名 (Instance) を示した。打切りなしの欄には、打切り操作を施さない従来の KLS を有するマルチスタート法の各試行によって得られた最良解値の平均値 (Avg), 最良解を得るまでの平均計算時間 (Time(s)), 平均移動操作回数 (step), 平均 LS 回数 (ls\_n) を示した。打切り 1, 打切り 2 の欄には、打切り 1, 打切り 2 をそれぞれ施した KLS を有するマルチスタート法の結果を打切りなしの欄と同様の形式で示した。

表 1 の Avg に注目すると、全体的に見て、打切り 1, 打切り 2 共に、打切りなしと同程度の解を算出できていることがわかる。しかし、部分的に見ると、打切り操作を施した KLS の方が悪質な解、良質な解を算出している問題例があることがわかる。2 パターンの打切り操作は、繰り返し回数がある程度回数で打切

表 1 DIMACS ベンチマークグラフに対する 3 タイプの KLS による実験結果

DIMACS benchmarks	k-opt Local Search												
	打切りなし				打切り 1				打切り 2				
	Avg	Time(s)	step	ls_n	Avg	Time(s)	step	ls_n	Avg	Time(s)	step	ls_n	
Instance	34.00	0.001	242	1	34.00	0.000	149	1	34.00	0.000	141	1	
C125.9	44.00	0.028	5714	32	44.00	0.013	3461	31	44.00	0.014	3151	35	
C250.9	56.30	0.265	39371	159	56.10	0.107	13853	88	56.20	0.099	11402	91	
C500.9	66.10	1.076	75261	245	66.30	0.924	48372	249	66.40	1.226	55068	364	
C1000.9	74.80	5.015	113040	309	75.10	9.619	144603	632	74.80	3.865	47573	267	
C2000.9	13.00	0.018	737	19	13.00	0.019	784	32	13.00	0.014	487	23	
DSJC500.5	14.90	0.264	4966	117	15.00	0.412	5638	212	14.70	0.270	3239	141	
DSJC1000.5	16.00	1.121	8696	143	16.00	1.835	7287	246	16.00	0.903	3140	122	
C2000.5	17.10	18.887	33796	655	17.00	14.288	16566	507	17.00	14.414	14240	514	
C4000.5	126.00	0.030	4283	11	126.00	0.011	1898	7	126.00	0.018	2955	14	
MANN_a27	343.80	4.953	331493	297	343.50	3.814	238997	348	343.80	3.562	210869	368	
MANN_a61	1098.10	79.234	1302682	368	1098.00	14.483	240631	109	1098.00	18.371	282724	154	
brock200.2	11.00	0.002	518	17	11.00	0.002	337	17	11.00	0.001	278	16	
brock200.4	16.00	0.001	676	12	16.00	0.003	660	21	16.00	0.003	490	17	
brock400.2	24.80	0.056	7367	87	24.50	0.058	5502	100	24.60	0.065	5522	123	
brock400.4	25.00	0.005	816	9	25.00	0.008	769	13	25.00	0.007	499	11	
brock800.2	20.70	0.273	10720	162	20.70	0.264	7010	166	20.80	0.192	4442	127	
brock800.4	20.50	0.442	17305	267	20.60	0.552	14713	349	20.50	0.443	10361	266	
gen200.p0.9.44	44.00	0.018	6904	42	44.00	0.018	5597	52	44.00	0.024	6978	83	
gen200.p0.9.55	55.00	0.004	1626	9	55.00	0.004	1444	12	55.00	0.006	2138	22	
gen400.p0.9.55	52.50	0.202	40460	170	52.10	0.061	9876	66	52.00	0.092	13417	112	
gen400.p0.9.65	65.00	0.063	13140	51	65.00	0.060	10087	63	65.00	0.073	10958	80	
gen400.p0.9.75	75.00	0.035	7237	26	75.00	0.024	4364	25	75.00	0.069	10523	92	
hamming8-4	16.00	0.000	46	1	16.00	0.000	27	1	16.00	0.000	23	1	
hamming10-4	40.00	0.108	4266	33	40.00	0.177	4591	57	40.00	0.088	2084	30	
keller4	11.00	0.000	36	1	11.00	0.000	26	1	11.00	0.000	26	1	
keller6	27.00	0.020	951	10	27.00	0.019	625	11	27.00	0.024	683	14	
keller8	55.50	22.642	127147	643	55.60	36.885	132835	1079	55.50	33.466	106040	1000	
p_hat300-1	8.00	0.001	113	5	8.00	0.001	92	6	8.00	0.001	70	5	
p_hat300-2	25.00	0.000	102	1	25.00	0.000	75	1	25.00	0.000	80	1	
p_hat300-3	36.00	0.006	1186	9	36.00	0.006	1131	13	36.00	0.008	1119	16	
p_hat700-1	11.00	0.064	1930	72	11.00	0.059	1186	71	11.00	0.055	1012	68	
p_hat700-2	44.00	0.004	354	2	44.00	0.003	185	1	44.00	0.005	191	2	
p_hat700-3	62.00	0.006	646	2	62.00	0.005	528	3	62.00	0.008	395	2	
p_hat1500-1	12.00	1.346	12463	390	12.00	1.294	7857	390	12.00	1.280	6928	390	
p_hat1500-2	65.00	0.028	1003	3	65.00	0.030	889	4	65.00	0.032	756	4	
p_hat1500-3	94.00	0.266	12916	25	94.00	0.244	9413	29	94.00	0.344	11258	43	

ろうとする単純な操作なので、問題例によっては良質な解を発見する前に繰り返しが打切られ、悪質な解を算出してしまふと考えられる。良質な解を算出している問題例については、KLSにおける探索の動作自体が打切り操作を施すことによって変化していると推測される。次にtime(s)に注目すると、全体的に見て、打切り1、打切り2共に、打切りなしの探索時間に対して改善が見られない。そこでstep、ls\_nに注目すると、stepは打切り操作を施すことによって全体的に減少しているが、問題例によっては、ls\_nが打切り操作を施すことによって増加している。このことから、打切り操作を施すことによって探索の動作自体が変化していることが推測され、また、それによって、打切り操作が効率的に働いていないと考えられる。以上の結果より、KLSの効率化のためには、単純な繰り返し回数による打切り操作ではなく、より巧妙なルールによる打切り操作の設計が必要であると考えられる。

### 3.2 KLSの探索範囲

従来のKLSでは、実験Iで用いたマルチスタート法を採用しており、 $n$ 回のKLSを繰り返し適用する。このマルチスタート法によるKLSの各試行はそれぞれ独立であり、KLSによって得られたクリークの情報を全く利用していない。そのため、純粋なKLSの探索性能が重要であり、その動作特性を調査することは有益な検討事項であると考えられる。本節では、KLSの動作特性を調査する一端として、与えられたグラフ中でのKLSの探索範囲について調査する。

#### 3.2.1 実験IIの詳細

実験IIでは、3.1節で検討した打切り操作がKLSの探索に与える影響も付加的に調査するために、上述の3タイプのKLSを対象とし、与えられたグラフ中での各KLSの探索範囲について調査する。対象グラフは実験Iと同様に、DIMACSのベンチマークグラフから37グラフを対象とする。各タイプのKLSは、実験Iと同様のマルチスタート法を採用し、各1回試行する。実験環境は実験Iと同様である。

探索範囲を表す指標を、対象グラフの頂点数に対する、各KLSにおける移動操作(Add移動操作、Drop移動操作)の対象となった頂点数の割合、すなわち以下のように定義する。

$$\text{探索範囲} = \frac{\text{移動操作の対象となった頂点数}}{\text{対象グラフの頂点数}} \times 100[\%]$$

#### 3.2.2 実験IIの結果

結果を表2に示す。DIMACS benchmarksの欄には、問題例名(Instance)、各問題例の頂点数(Nodes)、辺密度(Density)を示した。打切りなしの欄には、打切り操作を施さない従来のKLSを有するマルチスタート法の試行における、KLSの各試行による探索範囲の最大値(Max)、平均値(Avg)とその標準偏差(s.dev)、最小値(Min)を示した。なお、単位は全て%である。打切り1、打切り2の欄には、打切り1、打切り2をそれぞれ施したKLSを有するマルチスタート法の結果を打切りなしの欄と同様の形式で示した。

表2の結果を見ると、KLSの探索範囲は、全体的に、対象とするグラフの頂点数、辺密度に関連する傾向が観測できる。辺密度が同程度のグラフでは、頂点数が多いグラフの方が探索の割合は小さい。すなわち、探索の対象となる頂点数が同程度のため、グラフの頂点数に対する探索範囲の割合が低くなっていると考えられる。頂点数が同程度のグラフでは、辺密度が高いグラフの方が探索範囲の割合が大きい。つまり、探索の対象となる頂点数は、辺密度に比例して増加する傾向にあると考えられる。頂点数、辺密度共に同程度であるにもかかわらず、探索範囲に差が見られるグラフに関しては、各頂点の次数に偏りがあるなど、グラフの構造の違いが影響していると考えられる。また、打切りなしと打切り1、打切り2の結果を比較すると、全体的に見て、打切りのタイミングが早い打切り操作を施すほど、探索の割合が小さくなっていることがわかる。このことから、3.1.2節で述べている推測と同様に、打切り操作を施すことで探索の動作自体が変化していると推測される。

与えられたグラフ中でのKLSの探索範囲について、さらに調査するために、打切りなしのKLSを有するマルチスタート法をDIMACSベンチマークグラフから選出した最適解の算出が困難な4グラフ(C2000.9, MANN\_a81, keller6, p\_hat1500-1)に各1回適用し、対象グラフにおける、各頂点毎の移動操作回数を調

表2 DIMACS ベンチマークグラフに対する3タイプのKLSによる探索範囲

DIMACS benchmarks			k-opt Local Search											
			打切りなし				打切り1				打切り2			
Instance	Nodes	Density	Max	Avg	(s.dev)	Min	Max	Avg	(s.dev)	Min	Max	Avg	(s.dev)	Min
C126_9	125	0.898	79.20	51.52	(6.49)	39.20	59.20	41.77	(4.41)	32.80	55.20	37.95	(3.77)	30.40
C250_9	250	0.899	58.80	34.92	(5.83)	24.80	41.60	26.94	(3.48)	20.80	36.40	24.08	(2.65)	19.20
C500_9	500	0.900	43.40	23.60	(4.22)	16.00	29.00	18.00	(2.47)	13.20	24.60	15.77	(1.81)	12.20
C1000_9	1000	0.901	27.60	14.35	(2.80)	9.70	19.00	11.02	(1.55)	7.80	15.30	9.52	(1.12)	7.30
C2000_9	2000	0.900	18.75	8.90	(1.79)	5.50	11.65	6.46	(0.95)	4.60	8.80	5.52	(0.67)	4.10
DSJC1000_5	500	0.501	8.00	4.40	(0.74)	3.20	5.00	3.21	(0.35)	2.60	4.80	3.06	(0.31)	2.40
DSJC1000_6	1000	0.500	5.00	2.48	(0.43)	1.80	2.90	1.79	(0.21)	1.40	2.90	1.70	(0.18)	1.40
C2000_5	2000	0.500	2.80	1.37	(0.24)	1.05	1.65	0.99	(0.11)	0.80	1.60	0.94	(0.09)	0.75
C4000_5	4000	0.500	1.55	0.75	(0.13)	0.57	0.95	0.54	(0.06)	0.42	0.87	0.50	(0.05)	0.42
MANN_a27	378	0.990	82.01	65.19	(2.64)	63.75	60.05	49.55	(0.87)	48.67	51.58	44.39	(0.55)	43.38
MANN_a46	1035	0.996	83.09	66.59	(3.82)	64.73	60.28	49.49	(0.98)	48.88	51.78	44.12	(0.66)	43.67
MANN_a81	3321	0.998	83.52	66.99	(3.72)	65.43	58.95	49.51	(0.66)	49.26	52.21	44.06	(0.57)	43.84
brock200_2	200	0.496	16.50	8.90	(1.46)	6.50	9.50	6.43	(0.73)	5.00	9.50	6.20	(0.71)	5.00
brock200_4	200	0.657	25.50	13.95	(2.46)	10.50	17.00	10.36	(1.31)	8.00	14.50	9.72	(1.06)	7.50
brock400_2	400	0.749	22.00	11.43	(2.15)	8.50	14.50	8.63	(1.15)	6.50	13.00	7.83	(0.89)	6.25
brock400_4	400	0.748	21.50	11.44	(2.10)	8.00	15.00	8.61	(1.15)	6.75	12.50	7.82	(0.90)	6.25
brock800_2	800	0.651	9.00	4.68	(0.89)	3.37	5.87	3.43	(0.43)	2.62	5.12	3.15	(0.34)	2.50
brock800_4	800	0.649	8.87	4.61	(0.85)	3.37	6.00	3.42	(0.43)	2.75	5.25	3.13	(0.34)	2.50
gen200_pp_9_44	200	0.900	61.50	39.51	(6.55)	29.50	51.00	30.93	(3.98)	24.00	42.00	27.55	(3.10)	22.50
gen200_pp_9_65	200	0.900	70.50	42.82	(7.74)	29.50	56.50	33.73	(5.22)	24.00	49.50	30.26	(4.50)	23.00
gen400_pp_9_55	400	0.900	45.75	27.95	(4.93)	18.00	33.75	21.33	(2.91)	15.25	29.75	18.65	(2.15)	13.75
gen400_pp_9_65	400	0.900	54.75	29.03	(5.50)	18.75	40.75	21.65	(3.52)	15.00	36.00	18.73	(2.59)	13.75
gen400_pp_9_75	400	0.900	58.00	30.03	(6.28)	19.25	42.75	22.20	(4.04)	15.50	37.75	19.11	(2.98)	14.00
hamming8_4	256	0.639	12.50	11.84	(0.83)	10.15	7.81	7.36	(0.41)	6.64	7.03	6.82	(0.19)	6.64
hamming10_4	1024	0.828	19.14	7.38	(1.92)	5.76	11.91	5.39	(0.94)	4.39	10.15	4.82	(0.67)	4.10
keller4	171	0.649	22.80	11.02	(1.44)	8.18	14.05	8.69	(0.89)	5.84	14.03	8.62	(0.84)	5.84
keller5	776	0.751	9.53	5.77	(0.92)	3.99	6.70	4.22	(0.31)	3.60	6.05	4.17	(0.29)	3.22
keller6	3361	0.818	6.06	2.79	(0.40)	2.02	3.39	2.05	(0.15)	1.81	3.03	2.01	(0.13)	1.63
p_hat300-1	300	0.243	9.00	4.51	(0.82)	3.33	5.00	3.07	(0.45)	2.00	5.00	3.14	(0.44)	2.00
p_hat300-2	300	0.488	24.00	15.85	(2.40)	10.33	20.33	13.16	(1.84)	7.66	17.66	12.04	(1.62)	7.33
p_hat300-3	300	0.744	35.66	22.54	(3.48)	15.66	26.66	18.27	(2.36)	13.33	25.00	16.44	(1.92)	12.33
p_hat700-1	700	0.249	5.71	2.36	(0.46)	1.57	3.28	1.64	(0.24)	1.14	3.14	1.63	(0.21)	1.14
p_hat700-2	700	0.497	18.71	12.30	(1.56)	8.28	15.28	10.22	(1.14)	7.00	13.71	9.31	(1.00)	6.14
p_hat700-3	700	0.748	24.85	17.21	(2.05)	12.28	21.28	14.42	(1.41)	10.85	18.42	13.07	(1.19)	9.71
p_hat1500-1	1500	0.253	2.66	1.26	(0.24)	0.86	1.80	0.89	(0.11)	0.66	1.80	0.86	(0.11)	0.66
p_hat1500-2	1500	0.506	13.46	9.00	(1.08)	6.13	11.20	7.43	(0.79)	5.13	9.60	6.71	(0.68)	4.46
p_hat1500-3	1500	0.753	18.46	13.05	(1.42)	8.86	15.13	10.79	(1.02)	7.60	13.80	9.70	(0.86)	7.20

査した。

その結果を図3に示す。横軸を対象グラフにおける各頂点の番号、縦軸を各頂点毎の移動操作回数および次数とする。なお、このグラフは、各頂点の番号と、それに対応する移動操作回数および次数を移動操作回数で降順にソートしてプロットしたものである。

図3の結果から、次数の高い頂点は移動操作回数が多くなっていることが観測できる。特に MANN\_a81 と keller6 のグラフでは、次数が落ち込んでいる部分に対応して、移動操作回数も同様に落ち込んでおり、KLSによって重点的に探索をする範囲が各頂点の次数に強く依存していることがわかる。従って、KLSの探索は、各頂点の次数の影響を強く受けており、次数が多い頂点の周りを重点的に探索する傾向にあると考えられる。

#### 4. むすび

本論文では、MCPに対するk-opt局所探索法(KLS)を解析する一端として、KLSにおける近傍探索処理を打切る操作の検討と探索範囲の調査を行った。KLSにおける近傍探索処理を打切る操作の検討では、内ループの繰り返しをある程度の回数で打切る操作を2パターン提案し、打切りなしのKLSとの比較実験を行った。その結果、単純な回数制限による打切り操作では、全体的な探索時間の効率化を実現することができないことを示した。KLSの探索範囲の調査では、対象グラフの頂点数に対する、KLSによる移動操作の対象となった頂点数の割合を調べる実験を行った。その結果、KLSの探索では、全体的に見て、対象グラフの頂点数と辺密度によって、探索範囲が増減する傾向があることを示した。また、上述の打切り操作によって、探索範囲が小さくなる傾向を示した。さらに、対象グラフにおける、各頂点毎の移動操作回数と次数を調査した結果、KLSが重点的に探索する範囲は、対象グラフにおける各頂点の次数が強く影響する傾向を示した。以下では、今後の課題・検討事項について記述する。

(1) 本論文で提案した打切り操作では、全体的な探索時間の効率化は実現できなかったが、今後、単純に繰り返しの回数を制限するルールではなく、KLSの探索途中で得られる情報を利用した巧妙なルールの設計について検討の余地がある。

(2) 本論文で示したKLSにおける探索範囲の傾向を利用して、さらに良質な解を算出するために、対象グラフに依存している探索範囲をうまく拡大するアイデアの検討も興味深い。

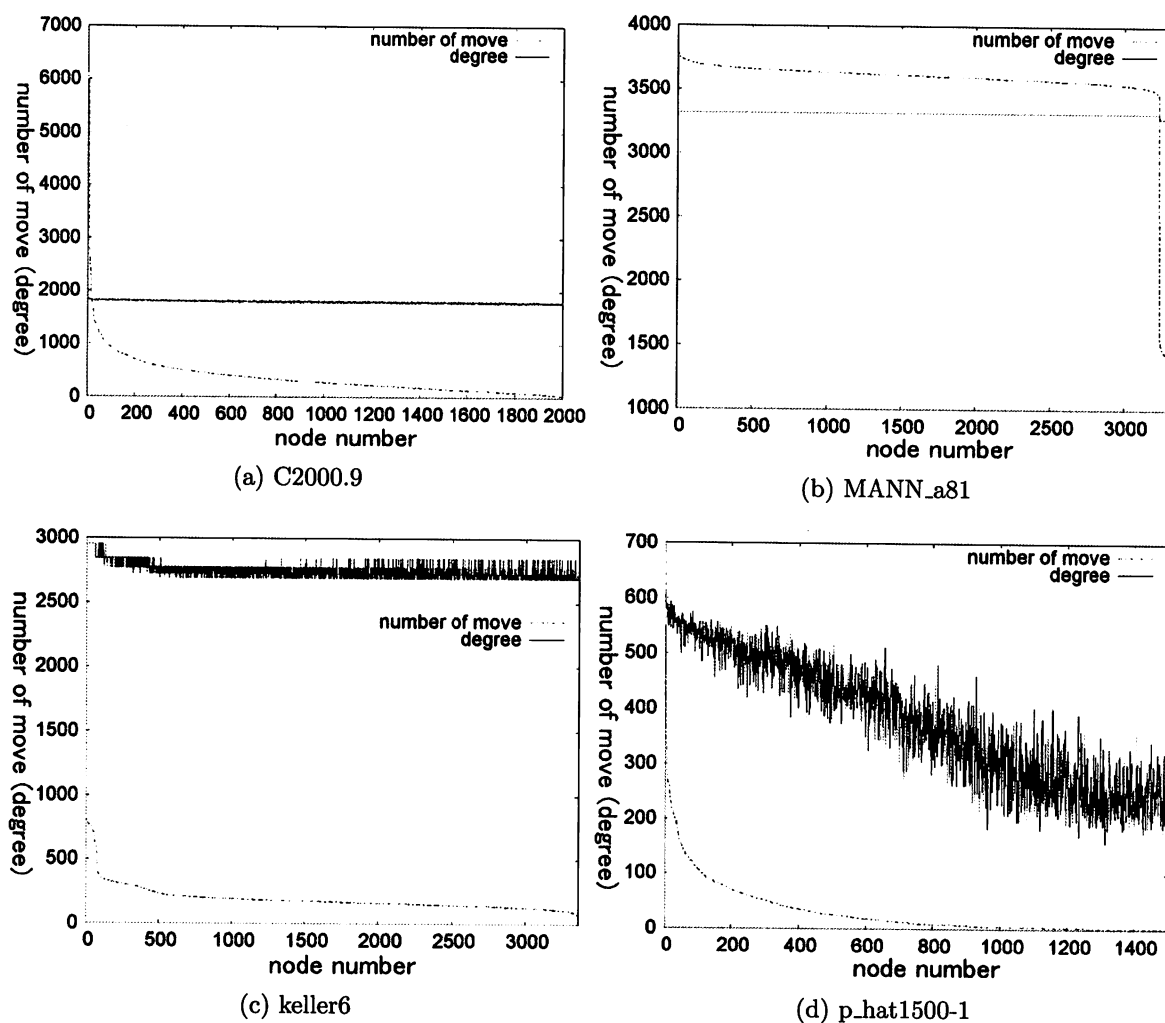


図3 対象グラフにおける各頂点の移動操作回数および次数

## 参考文献

- 1) I. Bomze, M. Budinich, P. Pardalos, and M. Pelillo, "The maximum clique problem," in Handbook of Combinatorial Optimization (suppl. Vol. A), D.-Z. Du, P.M. Pardalos (Eds.), pp.1-74, Kluwer, 1999.
- 2) M. Garey, and D. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness," Freeman, New York, 1979.
- 3) U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy, "Approximating clique is almost np-complete," Proc. the 32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, pp.2-12, 1991.
- 4) S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, "Proof verification and the hardness of approximation problems," Proc. the 33rd Annual Symposium on Foundations of Computer Science, Pittsburgh, PA, pp.14-23, 1992.
- 5) J. Håstad, "Clique is hard to approximate within  $n^{1-\epsilon}$ ," Acta Mathematica, vol.182, pp.105-142, 1999.
- 6) B. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," Bell System Technical Journal, vol.49, pp.291-307, 1970.
- 7) S. Lin and B. Kernighan, "An effective heuristic algorithm for the traveling salesman problem," Operations Research, vol.21, pp.498-516, 1973.

- 8) K. Katayama, A. Hamamoto, and H. Narihisa, "An effective local search for the maximum clique problem," *Information Processing Letters*, vol.95, no.5, pp.503–511, 2005.
- 9) D. Johnson and M. Trick, "Cliques, coloring, and satisfiability," *Second DIMACS Implementation Challenge*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, 1996.
- 10) E. Marchiori, "Genetic, iterated and multistart local search for the maximum clique problem," *Applications of Evolutionary Computing*, LNCS 2279, pp.112–121, Springer, 2002.
- 11) R. Battiti and M. Protasi, "Reactive local search for the maximum clique problem," *Algorithmica*, vol.29, no.4, pp.610–637, 2001.



## Analyses of $k$ -opt Local Search for Maximum Clique Problem

Masashi SADAMATSU, Kengo KATAYAMA\*, Hideo MINAMIHARA\*  
and Hiroyuki NARIHISA\*

*Graduate School of Engineering,  
\*Department of Information and Computer Engineering,  
Faculty of Engineering, Okayama University of Science.  
1-1 Ridai-cho, Okayama 700-0005, Japan.*

(Received October 2, 2006; accepted November 6, 2006)

We have already proposed an effective local search based on variable depth search, called  $k$ -opt local search (KLS), for solving the maximum clique problem. KLS efficiently explores the  $k$ -opt neighborhood defined as the set of neighbors that can be obtained by a sequence of several add and drop moves that are adaptively changed in the feasible search space. Although recent results showed that KLS is capable of finding high-quality cliques for DIMACS benchmark graphs, the analyses of KLS are not enough. In this paper, we analyze the stopping conditions and the search space of KLS.

**Keywords:** combinatorial optimization; maximum clique problem; variable depth search; local search.