

最大クリーク問題に対する局所探索法と頂点選択方式

濱本 明宏・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システム科学専攻

* 岡山理科大学工学部情報工学科

(2005年9月30日受付、2005年11月7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)³⁾ とは, 与えられたグラフ G に含まれるクリークの中で, 頂点数最大のクリークを求める問題である.

MCP は実用上重要な問題であり, 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の基本問題として頻繁にあらわれる³⁾. MCP は組合せ最適化問題の一つであり, NP-困難⁴⁾ である. 従って, 多項式時間で厳密解を求め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. さらに, MCP の (近似度の良い) 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, その他にも MCP の難しさを示すネガティブな報告がなされている.

組合せ最適化問題に対する高性能な近似解法の多くは, メタ戦略¹⁸⁾ にもとづくアルゴリズムであり, その基本構成は大域的および局所的な探索処理をバランス良く備えたものである. 特に, 局所的な探索処理を担う代表例は局所探索法 (Local Search, Local Improvement) であり, 多くの組合せ最適化問題に対して優れた局所探索法が提案されてきた. 局所探索法は近傍探索法とも呼ばれ, 現在の解の近傍の中から良好な近傍解への移動を繰り返すものである.

代表的な組合せ最適化問題である巡回セールスマン問題 (Traveling Salesman Problem, TSP)⁵⁾ に対する優れた局所探索法として, Lin と Kernighan による局所探索法¹⁴⁾ が良く知られている. 彼らの局所探索法は, 可変深度探索 (Variable Depth Search, VDS) のアイデアにもとづいており, TSP に対する他の古典的解法 (2-opt 局所探索法や 3-opt 局所探索法等) の探索性能を上回る高性能な局所探索法である. TSP に対する近年のメタ戦略アルゴリズムの多くは, 彼らの局所探索法を主力的な探索法として導入しており⁷⁾, 現在では非常に大規模な問題例へも適用されるに至っている¹⁾. Lin と Kernighan による VDS のアイデアは彼らによって同時期に適用されたグラフ分割問題¹³⁾ をはじめ, 最近では他の研究者により, 一般化割当問題¹⁹⁾, バイナリー 2 次計画問題^{8, 16)} や最大多様性問題^{9, 12)} 等へも波及しつつある. これらの局所探索法はそれぞれの最適化問題において代表的な近似解法として知られているだけでなく, TSP の場合と同様に, メタ戦略アルゴリズムを構成するための主力的な探索法^{9, 12, 17)} である.

最近我々は, MCP に対する局所探索法として, VDS のアイデア^{13, 14)} にもとづく k -opt 局所探索法 (k -opt Local Search, KLS)¹¹⁾ を提案し, 良好な結果を得た. KLS は, 各反復において, 現在のクリーク (解) から複数個の頂点を連鎖的に追加・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. その特長は, 比較的大きな近傍を効率良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 Workshop⁶⁾ で利用されたベンチマークグラフに適用した結果, そのワークショップで報告されていた最良解と多くのグラフにおいて同等以上の良質の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広く知られている, Marchiori による遺伝的局所探索法および反復局所探索法¹⁵⁾, Battiti らによるタブー

```

MCP-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)(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| is maximized;
12         if multiple vertices with the same size of the resulting |PA| 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;
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¹¹⁾

探索法にもとづく Reactive Local Search²⁾ との比較を通して、それらのメタ戦略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に算出可能であることを示した。

KLS はシンプルで高性能な局所探索法であるが、局所探索過程における頂点の追加・削除の手続き(頂点選択方式) に関しては改良の余地が残されていた。そこで本論文では、新しい頂点選択方式を採用した KLS(改良法) を示し、従来の KLS(従来法) との比較を通じて、改良法の探索性能を調査する。

2. MCP に対する k -opt 局所探索法と頂点選択方式

本章では、我々が既に提案している MCP に対する k -opt 局所探索法 (KLS) について記述する。KLS¹¹⁾ は、可変深度探索 (VDS)^{13, 14)} 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) から複数個の頂点を連鎖的に追加・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 (Line 1 - 18) と内ループ (Line 3 - 16) の処理を有する。以降、外ループに対しては「反復」、内ループについては「繰り返し」と呼び区別する。

まず、本論文で頻出する、図 1 中の重要な記号および用語を説明する。 $CC^{(l)}$ 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 $CC^{(l)}$ の全頂点に隣接する $CC^{(l)}$ に追加可能な頂点集合

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

である。

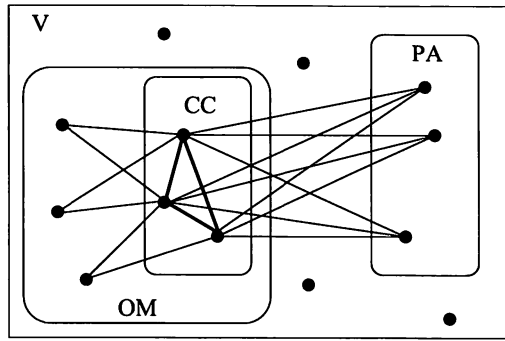


図2 CC, PA および OM の集合の一例

$OM^{(l)}$ は, $PA^{(l)}$ を若干緩和した 1 辺不足集合と呼ぶ辺集合

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

である. なお, $OM^{(l)}$ は $CC^{(l)}$ に含まれる 1 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ができる (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.

2.1 k -opt 近傍探索処理

以下では, KLS の各反復における基本アルゴリズム (k -opt 近傍探索処理) 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) $CC^{(0)}$ を対象に, Add 移動操作および Drop 移動操作を連鎖的に適用することで到達可能な近傍解の集合 $CC^{(0)}, \dots, CC^{(k)}, \dots, CC^{(r)}$ 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), 次の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 (上述の r に対応) が適応的に変動する.

上述の k -opt 近傍探索処理は, Add 移動操作を施す「Add フェーズ」 (Line 5 - 8) と Drop フェーズを施す「Drop フェーズ」 (Line 10 - 13) の二つのフェーズで構成される. 以下では, Add フェーズおよび Drop フェーズのそれぞれの頂点選択方式について記述する.

2.2 従来法の Add 移動頂点選択方式

反復 t において, 初期クリーク $CC^{(0)}$ および対応する $PA^{(0)}$ が与えられ, それぞれ $|CC^{(0)}| > 0$ および $PA^{(0)} \neq \emptyset$ であったと仮定する (繰り返しカウンタを $l = 0$ とする). $CC^{(0)}$ から $CC^{(1)}$ を得るための頂点の移動は, 誘導部分グラフ $G(PA^{(0)})$ 内の各頂点の次数 $deg_{G(PA^{(0)})}$ を考慮し, 最大次数の頂点 v を選ぶ. 最大次数の頂点が複数個ある場合には, それらの頂点群からランダムに一つの頂点を選択する. 選ばれた頂点 v を Add 移動操作により $CC^{(1)} := CC^{(0)} \cup \{v\}$ を得る. その後, $PA^{(0)}$ 等の情報を $PA^{(1)}$ 等に更新する. なお, k -opt 近傍探索処理中では, 一つの頂点が移動する度に (つまり, Add 移動または後に説明する Drop 移動操作が施される度に), 関連する PA 等の情報 (その他, $deg_{G(PA)}$ および後に説明する OM) を更新する. 更新の方法は文献²⁾ に示された方法と同様の更新法を採用する. 詳細は文献²⁾ を参照されたい.

Add 移動操作を続けると, CC はある繰り返し時点 l で Add 移動操作が不可能なクリーク $CC^{(l)}$ まで拡大される. Add 移動操作が不可能なクリークになると, Add フェーズを終了し, Drop フェーズに処理が移行する. 内ループの k -opt 近傍探索処理では, 拡大不可能なクリーク $CC^{(l)}$ から一つまたは複数の頂点を Drop フェーズにて削除することで, より良好な (大きな) クリークの探索を試みる.

2.3 従来法の Drop 移動頂点選択方式

Add フェーズが実施された後の内ループのある繰り返し時点 l で, 拡大不可能なクリーク $CC^{(l)}$ に到達したと仮定する. この時点で, Drop 移動操作により, 一つの頂点 v を $CC^{(l)}$ から削除する. 一つの頂点 v

```

MCP-k-opt-Local-Search( $CC, PA, OM, deg_G(PA)$ )
begin
1  repeat
2     $CC_{prev} := CC, D := CC_{prev}, P := \{1, \dots, n\}, g := 0, g_{max} := 0;$ 
3    repeat
4      if  $|PA \cap P| > 0$  then // Add Phase
5        find a vertex  $v$  with  $\max_{v \in \{PA \cap P\}} \{deg_G(PA \cap P)(v)\};$ 
6        if multiple vertices with the same maximum degree are found
7          then select one vertex  $v$  among them randomly;
8         $CC := CC \cup \{v\}, g := g + 1, P := P \setminus \{v\};$ 
9        if  $g > g_{max}$  then  $g_{max} := g, CC_{best} := CC;$ 
10       else // Drop Phase (if  $\{PA \cap P\} = \emptyset$ )
11         find a vertex  $v \in \{CC \cap P\}$  such that the resulting  $|PA \cap P|$  is maximized;
12         if multiple vertices with the same size of the resulting  $|PA \cap P|$  are found
13           then select one vertex  $v$  among them randomly;
14          $CC := CC \setminus \{v\}, g := g - 1, P := P \setminus \{v\};$ 
15         if  $v$  is contained in  $CC_{prev}$  then  $D := D \setminus \{v\};$ 
16       endif
17       update  $PA, OM,$  and  $deg_G(PA \cap P)(i), \forall i \in PA \cap P;$ 
18     until  $D = \emptyset;$ 
19     if  $g_{max} > 0$  then  $CC := CC_{best}$  else  $CC := CC_{prev};$ 
20 until  $g_{max} \leq 0;$ 
21 return  $CC;$ 
end;

```

図3 新しい頂点選択方式の導入した MCP に対する k -opt 局所探索法の疑似コード

を $CC^{(l)}$ から削除する際、 $CC^{(l)}$ から任意に削除するよりも、追加可能集合のサイズ $|PA^{(l+1)}|$ をできるだけ大きくする頂点を削除するのが望ましいことは明白である。この観点から、Drop フェーズでは、 $CC^{(l)}$ から削除する一つの頂点 v を選ぶ際、 $PA^{(l+1)}$ のサイズを最大にする頂点 v を選ぶ。 $PA^{(l+1)}$ のサイズが同じになる頂点が複数個存在する場合には、それらの頂点群からランダムに一つの頂点を選択する。

2.4 新たな頂点選択方式を採用した KLS

KLS を改良するために、KLS のアルゴリズム内部に導入可能な頂点選択方式について検討する。従来法では、Add 移動操作を適用する際、前述のように、 $PA^{(l)}$ からなる誘導部分グラフ $G(PA^{(l)})$ 内の各頂点の中で最大次数をもつ頂点を $CC^{(l)}$ に追加する。局所探索の各反復では、2 度以上同じ頂点が移動されることはないため、 k -opt 近傍処理後半では、 $G(PA^{(l)})$ に移動することが不可能な頂点が多く含まれる。 $PA^{(l)}$ に含まれる移動可能な頂点 ($v : v \in PA^{(l)}, v \in P$) が $G(PA^{(l)})$ 内で多くの移動不可能な頂点 ($v : v \in PA^{(l)}, v \notin P$) と隣接していると、後に拡大されるであろうクリークサイズに悪影響を与えると推測される。この問題を解決するために、改善法では、Add 移動操作を適用する際、移動可能集合 P を考慮して、 $PA^{(l)} \cap P$ からなる誘導部分グラフ $G(PA^{(l)} \cap P)$ 内の各頂点の中で最大次数をもつ頂点を $CC^{(l)}$ に追加する。

現在のクリーク $CC^{(l)}$ から一つの頂点 v を削除し、一つサイズの小さいクリークを得る際には、 $CC^{(l)}$ に一つの頂点を追加する場合と同様に P を考慮して、 $PA^{(l+1)} \cap P$ の個数が最大となる頂点を CC から削除する。

以上を踏まえ、新しい頂点選択方式の導入した KLS の疑似コードを図3に示す。

3. KLS の性能評価実験

3.1 実験方法

頂点選択方式の相違による KLS の探索性能を調査するために実験を行う。本論文では 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 問を対象とする。

一般に局所探索法は、一つの初期解が与えられた時、一つの局所最適解を算出する。従って一回だけの局所探索法の実行では良好な局所最適解を算出できない場合が多いため、通常、異なる初期解を与え、それぞれの初期解からスタートする局所探索法を複数回実行し、得られた最良解を出力する方法が一般に採用される。このような局所探索法の利用方法は、マルチスタート法と呼ばれ、局所探索法利用における最も単純かつ古典的な方法の一つである¹⁸⁾。本実験でも同様に、マルチスタート法を採用する。本実験で実施するマルチスタート法の詳細は以下の通りである。与えられたグラフ $G(V, E)$ の含まれる 1 頂点をクリークと見なすと、 $|V|(=n)$ 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して KLS を実行するマルチスタート法を採用する。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 の純粋な性能を検証するためには、マルチスタート法中、既に得られたクリークの情報を後の 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 n 個のクリーク群から最良解を出力する。なお、同じクリークサイズの最良解が複数得られる場合は、最初に得られる最良解を出力する。マルチスタート法の試行回数は 100 回である。

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。全ての実験は、HP ワークステーション xw6000(Xeon 2.8GHz) 上で実行した。

3.2 実験結果

結果を表 2 に示す。表 2 の Graph の欄には問題名 (Name) と第 2 回 DIMACS Implementation Challenge Workshop の参加者の最良の結果⁶⁾(BR) (* がついた BR は最適解値であることが証明されている) を示した。

改良法の欄には改良法を組み込んだ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最大値 (Best), 最良解値の平均値 (Avg), 最良解の平均値の標準偏差 (s.dev.) を示した。以降、最良解を得るまでの平均計算時間 (Time(s), 単位は秒), 平均計算時間の標準偏差 (s.dev.) を示した。

従来法の欄には文献¹⁰⁾ に示された従来法によるマルチスタート法の結果を引用した。ただし、従来法の平均計算時間は本実験環境に合わせて校正したものである。

従来法の結果と比較すると、改善法は Best の値で 37 問題例中 36 問に対して、同等もしくはより良い結果を示した。その内の 3 問題例に対しては、従来法よりも大きいクリークを算出することができた (brock200_4 はサイズが 1 大きい 17, brock400_2 はサイズが 8 大きい 33, gen400_p0.9_55 はサイズが 2 大きい 55)。Best の値で改善法が従来法よりも悪い結果を示したのは MANN_a81 の 1 問だけであり、改善法が算出したクリークサイズと従来法が算出したクリークサイズの差は 1 である。

従来法の Avg の値と改善法の Avg の値を比較すると、改善法は 37 問題例中 34 問に対して、同等もしくはより良い結果を示した (同じ結果を示したのが 23 問, より良い結果を示したのが 11 問)。C1000.9, MANN_a81, gen200_p0.09_44 の 3 問に対しては、従来法の結果の方がより結果であったが、改善法との差はわずかである。

次に、計算時間に関してふれる。改善法にもとづくマルチスタート法によって最良解を得るまで平均計算時間は、従来法よりも若干増加する傾向にあるようである。これは改善法が従来法に比べて平均的により良質な解を算出していることが原因であると推測される。従来法の Avg の値と改善法の Avg の値が同値であった問題例は 23 問であり、その 23 問に対する両解法の平均計算時間に着目すると、改善法は従来法よりも 23 問題中 15 問に対して良い結果を示した。その他の 8 問に対しては、従来法の平均計算時間の方が良い結果を示したが、改善法との差はわずかである。

表1 DIMACS マシンベンチマーク問題に対する User time (HP ワークステーション xw6000 Xeon 2.8GHz)

r100.5	r200.5	r300.5	r400.5	r500.5
<0.001	0.060	0.520	3.190	10.990

表2 DIMACS ベンチマークグラフに対する KLS の結果

Name	BR	改良法					従来法				
		Best	Avg	(s.dev)	Time(s)	(s.dev)	Best	Avg	(s.dev)	Time(s)	(s.dev)
C125.9	34*	34	34.00	(0)	< 0.001	(0.002)	34	34.00	(0)	<0.001	(0.003)
C250.9	44*	44	44.00	(0)	0.010	(0.011)	44	44.00	(0)	0.024	(0.023)
C500.9	57	57	56.18	(0.410)	0.178	(0.154)	57	56.15	(0.357)	0.237	(0.200)
C1000.9	67	68	66.22	(0.593)	1.292	(1.038)	68	66.38	(0.645)	1.759	(1.523)
C2000.9	76	77	75.07	(0.570)	11.720	(8.827)	77	74.90	(0.645)	12.908	(12.984)
DSJC500.5	13*	13	13.00	(0)	0.032	(0.026)	13	13.00	(0)	0.023	(0.012)
DSJC1000.5	15*	15	14.98	(0.140)	1.417	(1.023)	15	14.93	(0.255)	0.589	(0.487)
C2000.5	16	16	16.00	(0)	3.326	(2.783)	16	16.00	(0)	2.541	(2.162)
C4000.5	18	18	17.06	(0.237)	21.217	(29.395)	18	17.02	(0.140)	20.879	(15.752)
MANN_a27	126*	126	126.00	(0)	0.009	(0.009)	126	126.0	(0)	0.016	(0.013)
MANN_a45	345*	345	343.89	(0.343)	2.545	(2.345)	345	343.88	(0.354)	5.440	(4.848)
MANN_a81	1098	1099	1098.05	(0.260)	27.225	(54.418)	1100	1098.07	(0.292)	34.634	(73.109)
brock200.2	12*	11	11.00	(0)	0.010	(0.010)	11	11.00	(0)	0.003	(0.004)
brock200.4	17*	17	16.01	(0.099)	0.005	(0.006)	16	16.00	(0)	0.006	(0.006)
brock400.2	29*	25	24.88	(0.325)	0.118	(0.089)	25	24.42	(0.367)	0.117	(0.098)
brock400.4	33*	33	25.08	(0.796)	0.062	(0.058)	25	25.00	(0)	0.015	(0.011)
brock800.2	21	21	20.90	(0.300)	0.832	(0.654)	21	20.86	(0.347)	0.438	(0.293)
brock800.4	21	21	20.71	(0.453)	0.694	(0.679)	21	20.67	(0.470)	1.067	(0.830)
gen200_PO.9_44	44*	44	43.99	(0.099)	0.019	(0.017)	44	44.00	(0)	0.028	(0.020)
gen200_PO.9_55	55*	55	55.00	(0)	0.002	(0.004)	55	55.00	(0)	0.006	(0.006)
gen400_PO.9_55	55	55	52.24	(0.695)	0.100	(0.087)	53	52.21	(0)	0.183	(0.170)
gen400_PO.9_65	65	65	65.00	(0)	0.032	(0.0345)	65	65.00	(0)	0.006	(0.050)
gen400_PO.9_75	75	75	75.00	(0)	0.015	(0.013)	75	75.00	(0)	0.037	(0.028)
hamming8-4	16*	16	16.00	(0)	< 0.001	(0.002)	16	16.00	(0)	<0.001	(0.002)
hamming10-4	40	40	40.00	(0)	0.115	(0.109)	40	40.00	(0)	0.194	(0.1753)
keller4	11*	11	11.00	(0)	< 0.001	(0.001)	11	11.00	(0)	< 0.001	(0.001)
keller5	27	27	27.00	(0)	0.011	(0.011)	27	27.00	(0)	0.035	(0.029)
keller6	59	57	55.59	(0.633)	51.025	(36.126)	57	55.59	(0.708)	45.933	(54.800)
p_hat300-1	8*	8	8.00	(0)	0.002	(0.004)	8	8.00	(0)	0.001	(0.004)
p_hat300-2	25*	25	25.00	(0)	0.002	(0.004)	25	25.00	(0)	0.002	(0.003)
p_hat300-3	36*	36	36.00	(0)	0.008	(0.008)	36	36.00	(0)	0.010	(0.007)
p_hat700-1	11*	11	11.00	(0)	0.067	(0.063)	11	11.00	(0)	0.109	(0.056)
p_hat700-2	44*	44	44.00	(0)	0.007	(0.007)	44	44.00	(0)	0.007	(0.004)
p_hat700-3	62	62	62.00	(0)	0.013	(0.013)	62	62.00	(0)	0.014	(0.014)
p_hat1500-1	12*	12	12.00	(0)	3.210	(1.851)	12	12.00	(0)	2.285	(0.008)
p_hat1500-2	65	65	65.00	(0)	0.059	(0.046)	65	65.00	(0)	0.055	(0.038)
p_hat1500-3	94	94	94.00	(0)	0.411	(0.455)	94	94.00	(0)	0.376	(0.350)

4. むすび

我々が既に提案した最大クリーク問題 (MCP) に対する k -opt 局所探索法 (KLS) は可変深度探索法 (VDS) にもとづいており, MCP に対する他の近似解法と同等以上の探索性能を有する強力な局所探索法である. KLS の特徴は, 比較的大きな近傍を効率良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.

本論文では, この特徴を維持しながら, KLS の内部に新しい頂点選択方式の導入し, 頂点選択方式の相違による KLS の探索性能を調査した. DIMACS ベンチマークグラフを対象に新しい頂点選択方式の導入した KLS(改善法) を評価した結果, 改善法はより高品質な解を比較的短時間に算出可能であることを示し, その有効性を確認した.

参考文献

- 1) Applegate, D., Cook, W. and Rohe, A.: Chained Lin-Kernighan for Large Traveling Salesman Problems, *INFORMS Journal on Computing*, Vol. 15, No. 1, pp. 82-92 (2003).
- 2) Battiti, R. and Protasi, M.: Reactive Local Search for the Maximum Clique Problem, *Algorithmica*, Vol. 29, No. 4, pp. 610-637 (2001).
- 3) Bomze, I., Budinich, M., Pardalos, P. and Pelillo, M.: *The Maximum Clique Problem*, In D.-Z. Du, P.M. Pardalos (Eds.), *Handbook of Combinatorial Optimization* (suppl. Vol. A), Kluwer, pp. 1-74 (1999).
- 4) Garey, M. and Johnson, D.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York (1979).

- 5) Johnson, D. and McGeoch, L.: *The Traveling Salesman Problem: A Case Study*, Local Search in Combinatorial Optimization, John Wiley & Sons, pp. 215–310 (1997).
- 6) Johnson, D. and Trick, M.: *Cliques, Coloring, and Satisfiability*, Second DIMACS Implementation Challenge, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 26, American Mathematical Society (1996).
- 7) 片山謙吾, 成久洋之: 遺伝的反复局所探索法とその最適化性能, 電子情報通信学会論文誌 (A), Vol. J83-A, No. 2, pp. 179–187 (2000).
- 8) 片山謙吾, 成久洋之: バイナリー 2 次計画問題に対する変形 k -opt 局所探索法, 電子情報通信学会論文誌 (A), Vol. J84-A, No. 3, pp. 430–435 (2001).
- 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) Katayama, K., Hamamoto, A. and Narihisa, H.: An Effective Local Search for the Maximum Clique Problem, *Information Processing Letters*, Vol. 95, No. 5, pp. 503–511 (2005).
- 11) Katayama, K., Hamamoto, A. and Narihisa, H.: Solving the Maximum Clique Problem by k -opt Local Search, *Proceedings of the 2004 ACM Symposium on Applied Computing*, Vol. 2, pp. 1021–1025 (2004).
- 12) Katayama, K. and Narihisa, H.: *An Evolutionary Approach for the Maximum Diversity Problem*, Recent Advances in Memetic Algorithms, W.E. Hart, N. Krasnogor, and J.E. Smith (Eds.), Springer, pp. 31–47 (2004).
- 13) Kernighan, B. and Lin, S.: An Efficient Heuristic Procedure for Partitioning Graphs, *Bell System Technical Journal*, Vol. 49, pp. 291–307 (1970).
- 14) Lin, S. and Kernighan, B.: An Effective Heuristic Algorithm for the Traveling Salesman Problem, *Operations Research*, Vol. 21, pp. 498–516 (1973).
- 15) Marchiori, E.: *Genetic, Iterated and Multistart Local Search for the Maximum Clique Problem*, Applications of Evolutionary Computing, LNCS 2279, Springer, pp. 112–121 (2002).
- 16) Merz, P. and Freisleben, B.: Greedy and Local Search Heuristics for Unconstrained Binary Quadratic Programming, *Journal of Heuristics*, Vol. 8, No. 2, pp. 197–213 (2002).
- 17) Merz, P. and Katayama, K.: Memetic Algorithms for the Unconstrained Binary Quadratic Programming, *BioSystems*, Vol. 78, No. 1–3, pp. 99–118 (2004).
- 18) 柳浦睦憲, 茨木俊秀: 組合せ最適化—メタ戦略を中心として—, 朝倉書店 (2001).
- 19) Yagiura, M., Yamaguchi, T. and Ibaraki, T.: *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, pp. 459–471 (1999).

Vertex Selection Rules of Local Search for the Maximum Clique Problem

Akihiro HAMAMOTO, Kengo KATAYAMA*,
Hideo MINAMIHARA* and Hiroyuki NARIHISA*

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

(Received September 30, 2005; accepted November 7, 2005)

We have already proposed a variable depth search based algorithm, called k -opt local search (KLS), for the maximum clique problem and shown that KLS outperformed recent metaheuristic algorithms. 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. In this paper, we use efficient KLS algorithm which explores k -opt neighborhood in the search space deleted from already moved vertices. Computational results show that KLS is capable of finding high-quality cliques with reasonable running time for DIMACS benchmark graphs.