

最大クリーク問題に対する反復 k -opt 局所探索法の Kick 法の性能比較

幸村 明典・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

(2009年9月30日受付、2009年11月5日受理)

1. まえがき

組合せ最適化問題は、その多くが NP-困難¹⁾ であるため、現実的な時間で良質の解を求める近似解法の研究が盛んである。近年、計算機性能の急速な向上により、多少時間はかかっても、より精度の高い解を求める解法の要求が高まっている。この目的を実現するための一般的な枠組みを提供しようとするのがメタ戦略である。メタ戦略の代表的なものに反復局所探索法がある。反復局所探索法は与えられた解を改善し、より良い解にする操作である局所探索と、局所解から脱出する操作である突然変異を繰り返し適応する手法である。

実用上重要な組合せ最適化問題の一つに最大クリーク問題 (Maximum Clique Problem, MCP) がある。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²⁾、NP-困難である。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³⁾、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⁴⁾⁵⁾。我々が以前行った地形解析の結果は MCP のほとんどの問題例で大谷構造を観測できなかった。このことは、MCP の探索において最適解への接近が容易ではないということを示している。

最近我々は、MCP に対する反復局所探索法として、反復 k -opt 局所探索法 (Iterated k -opt Local Search, IKLS) を提案した⁹⁾。IKLS の局所探索法には、VDS のアイデアにもとづく k -opt 局所探索法 (KLS) を使用している。また、IKLS の突然変異には LEC-Kick を使用しており、DIMACS ベンチマークグラフに対して良い結果を示している。本論文では LEC-Kick を含め、その他 5 タイプの Kick を示し、Kick の違いによる IKLS の探索性能の違いを検証する。

2. 最大クリーク問題

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)²⁾ 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。

3. 大谷構造

特定の組合せ最適化問題に対して局所探索法などを用いた地形解析 (landscape analysis) の研究が行われている。地形とは、探索空間における構造の複雑さの尺度である。その解析によって、対象とする最適化問題を発見的に解くことの困難さのある程度見積もることが可能になる。現在までに、巡回セールスマン問題、バイナリー 2 次計画問題¹⁴⁾ などに対する地形解析が知られており、その多くでは、「大谷構造」が観測されている。大谷構造とは、探索空間の地形が真の最適解に向かって一つの大きな谷となるような構造をしていることである。MCP に対する地形解析の結果は、そのほとんどの問題例で大谷構造が観測でき

```

procedure IKLS
input: graph  $G = (V, E)$ ;
output: best clique  $C_{best}$  in  $G$ ;
begin
1  generate  $C$ ; compute  $PA, OM$ , and  $deg_{G(PA)}$ ;
2   $C := \text{Local Search}(C, PA, OM, deg_{G(PA)})$ ;  $C_{best} := C$ ;
3  repeat
4     $C := \text{Kick}(C, PA, OM, deg_{G(PA)})$ ;
5     $C := \text{Local Search}(C, PA, OM, deg_{G(PA)})$ ;
6    if  $|C| > |C_{best}|$  then  $C_{best} := C$ ; endif
7    if restart=true then
8      generate  $C$ ; compute  $PA, OM$ , and  $deg_{G(PA)}$ ;
9       $C := \text{Local Search}(C, PA, OM, deg_{G(PA)})$ ;
10     if  $|C| > |C_{best}|$  then  $C_{best} := C$ ; endif
11    endif
12  until terminate=true;
13  return  $C_{best}$ ;
end;

```

図1 MCP に対する反復 k -opt 局所探索法

ないというものであった¹⁰⁾.

4. MCP に対する反復 k -opt 局所探索法 (IKLS)

本節では、MCP に対する Iterated k -opt Local Search (IKLS)⁹⁾ について記述する。IKLS は反復局所探索法を基にしたアルゴリズムである。反復局所探索法は局所探索 (Local Search:LS) と突然変異 (Kick) の2つの操作から構成されている。2つの操作を反復しながら探索を進めて行く。

4.1 IKLS の基本アルゴリズム

MCP に対する反復 k -opt 局所探索法 (Iterated k -opt Local Search, IKLS) の流れを図1に示す。IKLS の主要な構成要素は、Local Search (Line 2, Line 5), Kick (Line 4), Restart (Line 7~11) であり、これらの反復により探索を行う。まず、初期プロセスとして、グラフ G の頂点からランダムに選択した1頂点を初期クリーク C とし、 $PA, OM, deg_{G(PA)}$ を更新する (詳細は4.2節を参照) (Line 1)。そして、Line 2で初期クリーク C に対して KLS を適用することで IKLS の初期プロセスを完了する。その後、Line 3~12のメインループによって探索を行う。Line 4では、KLS を適用して得られた解に対して Kick を適用し、新たな初期解を生成する。Line 5では、Kick によって得られた解に対して KLS を適用することによりさらにクリークの拡大を図る。Line 7~11では、探索の多様性を持たせるための処理として、IKLS に対して Restart 処理を施す。その条件は、KLS の探索途中における最良解値 ($|C_{best}|$) と同じ回数 KLS を適用しても最良解の更新が無い場合である。Restart における初期クリークとして、グラフ G の中から $V \setminus C$ である1頂点をランダムに選択する。IKLS の終了条件 (Line 12) は IKLS の処理中における KLS の適用回数が $n \times 100$ 回に到達した時、もしくは、最適解もしくは既知の最良解と同じサイズのクリークが得られた時とする。以下で、IKLS の主要な構成要素である Local Search, Kick について示す。

4.2 Local Search

IKLS 内部で使用する Local Search として、我々が既に提案している MCP に対する k -opt 局所探索法 (KLS)¹²⁾ を用いる。KLS の概要について以下に示す。

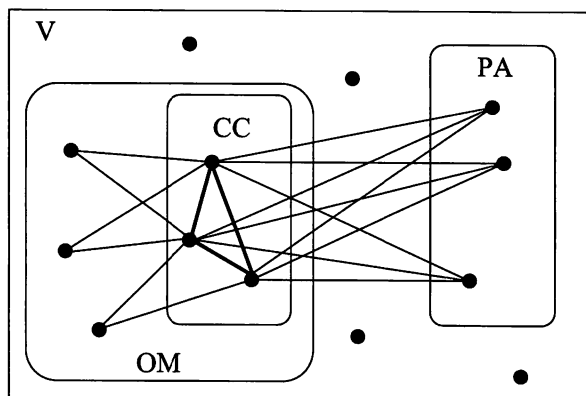
4.2.1 KLS の基本アルゴリズム

KLS は、可変深度探索 (Variable Depth Search, VDS)⁶⁾⁷⁾ 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

```

procedure KLS( $CC, PA, OM, deg_G(PA)$ )
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  $deg_G(PA)(i), \forall i \in PA \cap P;$ 
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2 MCP に対する k -opt 局所探索法の擬似コード図3 CC, PA および OM の集合の一例

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2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、図2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 $CC^{(l)}$ の全頂点に隣接する、 $CC^{(l)}$ に追加可能な頂点の集合

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

```

procedure LEC-Kick( $CC, PA, OM, deg_G(PA)$ )
begin
1  if all  $i \in CC$  are disconnected to all  $j \in V \setminus CC$  then
2    select a vertex  $v \in V \setminus CC$  randomly; compute  $PA, OM$ , and  $deg_G(PA)$ ;
3     $CC := \emptyset$ ;  $CC := CC \cup \{v\}$ ; return new clique  $CC$ ;
4  endif
5  find a vertex  $v \in V \setminus CC$  with the lowest edge number to vertices of  $CC$ .
6  if multiple vertices with the same lowest edge number are found
   then select one vertex  $v$  among them randomly;
7  drop vertices from  $CC$  that are not connected to  $v$ ;
   // the dropped vertices are removed from  $P$  in Fig. 2 (line 3) only for 1st iteration of the next KLS.
8  update  $PA, OM$ , and  $deg_G(PA)$ ;
9  return new clique  $CC$ ;
end;

```

図4 LEC-Kickの擬似コード

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3参照）。 $deg_G(PA^{(l)})$ 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、複数個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。

4.2.2 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。

4.2.3 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。

4.3 Kick

IKLS で使用する突然変異 (Kick) として、6種類の Kick (LEC-Kick, HEC-Kick, RAC-Kick, 適応的 1-Kick, 適応的 2-Kick, 適応的 3-Kick) を説明する。LEC-Kick, HEC-Kick, RAC-Kick¹¹⁾ は探索のどの世代においても、各々の Kick の条件 (次数の数) を満たす頂点を選び Kick をする。他方、適応的 1-Kick, 適応的 2-Kick, 適応的 3-Kick は LEC-Kick をベースにしながらも、探索の各世代毎に違う条件の頂点を

選べる工夫が施してある。

4.3.1 LEC-Kick

LEC-Kick の擬似コードを図 4 に示す。まず LEC-Kick の例外処理 (Line 1~4) について説明する。与えられたクリーク CC に対して、グラフ G における CC 以外の頂点群 $V \setminus CC$ が全く隣接していない場合、LEC-Kick を適用することができない。この場合、例外処理として、 $V \setminus CC$ から 1 頂点をランダムに選択し、その 1 頂点を新たなクリークとみなし、探索を継続する。以下、LEC-Kick を適用時の具体的な操作について示す。 CC が与えられた時、 $V \setminus CC$ で、 CC の頂点群と最低 1 頂点以上隣接している頂点集合から、隣接している数が最も少ない頂点 v を選択する (Line 5)。選択する頂点の候補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 1 頂点を選択する (Line 6)。そして、頂点 v と頂点 v に隣接する CC の頂点群とで新たなクリークを構成する (Line 7~9)。なお、Kick によって CC から削除された頂点群は、次の KLS の探索における 1 回目の反復で集合 P (図 2, Line 3) から除外する。

図 5 は CC が与えられたときに、隣接している頂点の中からどの頂点を選ばれるかを示したものである。 CC には頂点 $v_1 \sim v_4$ が接続している。これらの頂点は図 4 の line 5 で選択される頂点である。LEC-Kick では CC と最小次数で接続している v_4 が選ばれる。

また、HEC-Kick, RAC-Kick は Line5 の隣接している頂点の選択の仕方に違いがある。

4.3.2 HEC-Kick

グラフ G において、 CC が与えられた時、 $V \setminus CC$ で、 CC の頂点群と最低 1 頂点以上隣接している頂点集合から、隣接している数が最も多い頂点 v を選択する。そして、頂点 v と頂点 v に隣接する CC の頂点群とで新たなクリークを構成する。

HEC-Kick では図 5 において、 $v_1 \sim v_4$ の中から、 CC と最大次数で接続している頂点 v_3 が選ばれる。

4.3.3 RAC-Kick

グラフ G において CC 以外の頂点群 $V \setminus CC$ で、 CC の頂点群と最低 1 頂点以上隣接している頂点集合から、LEC-Kick で選ばれる頂点と HEC-Kick で選ばれる頂点を除いた頂点から、ランダムに 1 頂点 v を $V \setminus CC$ から選択する。そして、頂点 v と頂点 v に隣接する CC の頂点群とで新たなクリークを構成する。

RAC-Kick では図 5 において、 $v_1 \sim v_4$ の中から、LEC-Kick および HEC-Kick において選択される頂点を除外した頂点の $v_1 \sim v_2$ からランダムに 1 頂点を選択する。

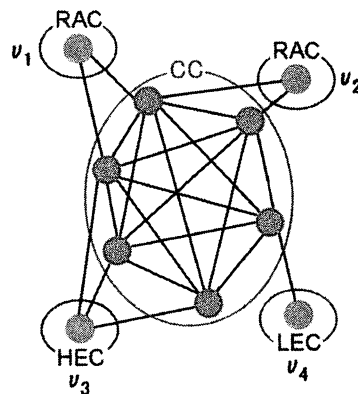


図 5 LEC-Kick, HEC-Kick, RAC-Kick の頂点選択

4.3.4 適応的 1-Kick

適応的 1-Kick, 適応的 2-Kick, 適応的 3-Kick は CC と隣接している頂点の中から、隣接している数が LEC-Kick, HEC-Kick, RAC-Kick とは違う基準で頂点を選択する。そして、ここで使用した適応的とは、LEC-Kick では選べない頂点を探索の状況に応じて選べるようにしているという意味である。適応的 1-Kick,

適応的 2-Kick, 適応的 3-Kick は Line 5 の隣接している頂点の選択の条件を, 探索の世代ごとに変化させる.

現世代 i の探索によって得られた解を $CC^{(i)}$ としその評価値を $|CC^{(i)}|$ と表す. また, 前世代を $i-1$ と表現する. 適応的 1-Kick は $|CC^{(i-1)}| < |CC^{(i)}|$ ならば, G において $CC^{(i)}$ に隣接している頂点群の中から, $CC^{(i-1)}$ と $CC^{(i)}$ 以外の頂点との間で接続した次数よりも相対的に 1 次数大きい次数で $CC^{(i)}$ と接続する頂点を選択する. そして, 選択した頂点とその頂点に隣接する $CC^{(i)}$ の頂点群とで新たなクリークを生成する. また, $|CC^{(i-1)}| > |CC^{(i)}|$ または $|CC^{(i-1)}| = |CC^{(i)}|$ ならば $CC^{(i)}$ 以外の頂点群の中から $CC^{(i)}$ と接続する次数が相対的に 1 次数小さい頂点を選択し, 新たなクリークを生成する.

図 6 は $i+1$ 世代において, CC が与えられたときに, 隣接している頂点の中からどの頂点が選ばれるかを示したものである. 適応的 1-Kick では v_6 が選ばれる.

4.3.5 適応的 2-Kick

適応的 2-Kick では $CC^{(i)}$ と接続している頂点の中から頂点を選択する場合, 選択される頂点の次数は探索の各世代において $|CC^{(i)}|$ を考慮しながら決定される. それに対して, 適応的 1-Kick は, $CC^{(i)}$ と接続している頂点の中から頂点を選択する場合, 探索のどの世代においても選択される頂点の次数は一定の尺度で決定する.

現世代 i の探索によって得られた解を $CC^{(i)}$ としその評価値を $|CC^{(i)}|$ と表す. また, 前世代を $i-1$ と表現する. 適応的 2-Kick は $|CC^{(i-1)}| < |CC^{(i)}|$ ならば G において, $CC^{(i)}$ に隣接している頂点群の中から, $CC^{(i-1)}$ と $V \setminus CC^{(i-1)}$ の頂点との間で接続していた次数よりも $|CC^{(i)}| - |CC^{(i-1)}|$ 次数相対的に大きい次数で $CC^{(i)}$ と接続する頂点を選択する. もし, $|CC^{(i-1)}| > |CC^{(i)}|$ ならば, $|CC^{(i-1)}| < |CC^{(i)}|$ のときは逆の操作をする. また, $|CC^{(i-1)}| = |CC^{(i)}|$ のときは, $i-1$ 世代と同様の処理をする.

適応的 2-Kick では図 6 において v_2 が選ばれる.

4.3.6 適応的 3-Kick

適応的 2-Kick での i 世代と $i-1$ 世代のクリークサイズの条件を逆にした Kick 法である.

適応的 3-Kick では図 6 において v_5 が選ばれる.

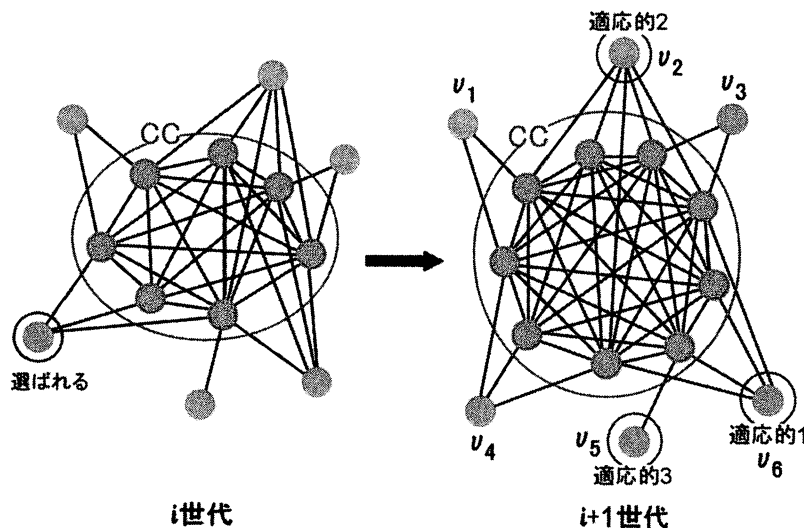


図 6 適応的 1-Kick, 適応的 2-Kick, 適応的 3-Kick の頂点選択

表1 異なる Kick を実装した 6 種類の IKLS の結果

Instance		IKLS								
name	BR	LEC-Kick			HEC-Kick			RAC-Kick		
		Best	Avg	Time	Best	Avg	Time	Best	Avg	Time
C125.9	34*	34 (25/25)	34.00	0.001	34 (25/25)	34.00	0.002	34 (25/25)	34.00	0.001
C250.9	44*	44 (25/25)	44.00	0.013	44 (25/25)	44.00	0.041	44 (25/25)	44.00	0.016
C500.9	57	57 (25/25)	57.00	2.212	57 (25/25)	57.00	1.334	57 (25/25)	57.00	0.751
C1000.9	68	68 (24/25)	67.96	69.600	68 (25/25)	67.32	13.766	68 (25/25)	68.00	14.134
C2000.9	78	77 (5/25)	76.20	290.075	77 (2/25)	76.08	59.357	78 (7/25)	77.24	252.791
DSJC500.5	13*	13 (25/25)	13.00	0.053	13 (25/25)	13.00	0.020	13 (25/25)	13.00	0.032
DSJC1000.5	15*	15 (25/25)	15.00	1.978	15 (25/25)	15.00	1.218	15 (25/25)	15.00	1.047
C2000.5	16	16 (25/25)	16.00	5.561	16 (25/25)	16.00	1.509	16 (25/25)	16.00	1.696
NANN.a27	126*	126 (25/25)	126.00	0.031	126 (25/25)	126.00	0.274	126 (25/25)	126.00	0.289
NANN.a45	345*	345 (24/25)	344.96	789.094	344 (21/25)	343.84	260.632	344 (9/25)	343.36	236.093
brock200.2	12*	12 (25/25)	12.00	0.084	12 (4/25)	11.16	0.064	12 (25/25)	12.00	0.273
brock200.4	17*	17 (25/25)	17.00	0.149	17 (3/25)	16.12	0.090	17 (25/25)	17.00	0.597
brock400.2	29*	29 (17/25)	27.72	5.122	25 (25/25)	25.00	0.124	29 (1/25)	25.16	0.227
brock400.4	33*	33 (25/25)	33.00	1.243	33 (1/25)	25.32	0.294	33 (17/25)	30.44	3.726
brock800.2	24	21 (25/25)	21.00	1.194	21 (25/25)	21.00	0.484	21 (25/25)	21.00	1.284
brock800.4	26	26 (5/25)	22.00	5.622	21 (25/25)	21.00	1.387	21 (25/25)	21.00	1.444
gen200.p0.9.44	44*	44 (25/25)	44.00	0.006	44 (25/25)	44.00	0.031	44 (25/25)	44.00	0.018
gen200.p0.9.55	55*	55 (25/25)	55.00	0.002	55 (25/25)	55.00	0.005	55 (25/25)	55.00	0.004
gen400.p0.9.55	55	55 (25/25)	55.00	3.046	55 (14/25)	54.12	3.081	55 (25/25)	55.00	0.571
gen400.p0.9.65	65	65 (25/25)	65.00	0.015	65 (25/25)	65.00	0.070	65 (25/25)	65.00	0.018
gen400.p0.9.75	75	75 (25/25)	75.00	0.009	75 (25/25)	75.00	0.037	75 (25/25)	75.00	0.010
hamming8-4	16*	16 (25/25)	16.00	< ε	16 (25/25)	16.00	< ε	16 (25/25)	16.00	< ε
hamming10-4	40	40 (25/25)	40.00	0.015	40 (25/25)	40.00	0.027	40 (25/25)	40.00	0.016
keller4	11*	11 (25/25)	11.00	< ε	11 (25/25)	11.00	< ε	11 (25/25)	11.00	< ε
keller5	27	27 (25/25)	27.00	0.028	27 (25/25)	27.00	0.039	27 (25/25)	27.00	0.044
keller6	59	59 (25/25)	59.00	3.961	59 (1/25)	56.32	273.072	59 (9/25)	57.76	500.342
p.hat300-1	8*	8 (25/25)	8.00	0.002	8 (25/25)	8.00	0.002	8 (25/25)	8.00	0.001
p.hat300-2	25*	25 (25/25)	25.00	0.001	25 (25/25)	25.00	0.001	25 (25/25)	25.00	0.001
p.hat300-3	36*	36 (25/25)	36.00	0.003	36 (25/25)	36.00	0.010	36 (25/25)	36.00	0.004
p.hat700-1	11*	11 (25/25)	11.00	0.070	11 (25/25)	11.00	0.045	11 (25/25)	11.00	0.025
p.hat700-2	44*	44 (25/25)	44.00	0.005	44 (25/25)	44.00	0.005	44 (25/25)	44.00	0.004
p.hat700-3	62	62 (25/25)	62.00	0.012	62 (25/25)	62.00	0.013	62 (25/25)	62.00	0.010
p.hat1500-1	12*	12 (25/25)	12.00	15.199	12 (25/25)	12.00	5.800	12 (25/25)	12.00	3.892
p.hat1500-2	65	65 (25/25)	65.00	0.049	65 (25/25)	65.00	0.060	65 (25/25)	65.00	0.031
p.hat1500-3	94	94 (25/25)	94.00	0.122	94 (25/25)	94.00	0.403	94 (25/25)	94.00	0.089
Total			1730.84	1194.58		1712.28	623.287		1722.96	1019.461

Instance		IKLS								
name	BR	適応的 1-Kick			適応的 2-Kick			適応的 3-Kick		
		Best	Avg	Time	Best	Avg	Time	Best	Avg	Time
C125.9	34*	34 (25/25)	34.00	0.001	34 (25/25)	34.00	0.001	34 (25/25)	34.00	0.001
C250.9	44*	44 (25/25)	44.00	0.012	44 (25/25)	44.00	0.014	44 (25/25)	44.00	0.017
C500.9	57	57 (25/25)	57.00	2.061	57 (25/25)	57.00	2.048	57 (25/25)	57.00	1.835
C1000.9	68	68 (22/25)	67.88	33.204	68 (25/25)	68.00	37.855	68 (23/25)	67.92	46.115
C2000.9	78	77 (4/25)	76.12	248.970	78 (2/25)	76.60	181.301	77 (3/25)	76.08	240.817
DSJC500.5	13*	13 (25/25)	13.00	0.050	13 (25/25)	13.00	0.041	13 (25/25)	13.00	0.057
DSJC1000.5	15*	15 (25/25)	15.00	2.315	15 (25/25)	15.00	1.612	15 (25/25)	15.00	1.906
C2000.5	16	16 (25/25)	16.00	7.486	16 (25/25)	16.00	7.225	16 (25/25)	16.00	7.009
NANN.a27	126*	126 (25/25)	126.00	0.027	126 (25/25)	126.00	0.029	126 (25/25)	126.00	0.030
NANN.a45	345*	345 (18/25)	344.72	505.839	345 (24/25)	344.96	858.877	345 (17/25)	344.68	514.873
brock200.2	12*	12 (25/25)	12.00	0.075	12 (25/25)	12.00	0.080	12 (25/25)	12.00	0.100
brock200.4	17*	17 (25/25)	17.00	0.148	17 (25/25)	17.00	0.188	17 (25/25)	17.00	0.186
brock400.2	29*	29 (18/25)	27.88	4.883	29 (17/25)	27.72	4.225	29 (18/25)	27.88	3.064
brock400.4	33*	33 (25/25)	33.00	1.696	33 (25/25)	33.00	1.507	33 (25/25)	33.00	1.727
brock800.2	24	24 (1/25)	21.12	2.625	24 (1/25)	21.12	2.555	24 (2/25)	21.24	3.163
brock800.4	26	26 (3/25)	21.60	5.500	26 (2/25)	21.40	3.979	26 (4/25)	21.80	5.116
gen200.p0.9.44	44*	44 (25/25)	44.00	0.012	44 (25/25)	44.00	0.006	44 (25/25)	44.00	0.011
gen200.p0.9.55	55*	55 (25/25)	55.00	0.002	55 (25/25)	55.00	0.002	55 (25/25)	55.00	0.003
gen400.p0.9.55	55	55 (25/25)	55.00	2.290	55 (25/25)	55.00	1.471	55 (25/25)	55.00	1.703
gen400.p0.9.65	65	65 (25/25)	65.00	0.015	65 (25/25)	65.00	0.012	65 (25/25)	65.00	0.013
gen400.p0.9.75	75	75 (25/25)	75.00	0.008	75 (25/25)	75.00	0.008	75 (25/25)	75.00	0.009
hamming8-4	16*	16 (25/25)	16.00	< ε	16 (25/25)	16.00	< ε	16 (25/25)	16.00	< ε
hamming10-4	40	40 (25/25)	40.00	0.013	40 (25/25)	40.00	0.013	40 (25/25)	40.00	0.017
keller4	11*	11 (25/25)	11.00	< ε	11 (25/25)	11.00	< ε	11 (25/25)	11.00	< ε
keller5	27	27 (25/25)	27.00	0.028	27 (25/25)	27.00	0.036	27 (25/25)	27.00	0.022
keller6	59	59 (25/25)	59.00	7.088	59 (25/25)	59.00	62.100	59 (25/25)	59.00	8.096
p.hat300-1	8*	8 (25/25)	8.00	0.002	8 (25/25)	8.00	0.002	8 (25/25)	8.00	0.002
p.hat300-2	25*	25 (25/25)	25.00	0.001	25 (25/25)	25.00	0.001	25 (25/25)	25.00	0.001
p.hat300-3	36*	36 (25/25)	36.00	0.004	36 (25/25)	36.00	0.004	36 (25/25)	36.00	0.003
p.hat700-1	11*	11 (25/25)	11.00	0.055	11 (25/25)	11.00	0.060	11 (25/25)	11.00	0.063
p.hat700-2	44*	44 (25/25)	44.00	0.006	44 (25/25)	44.00	0.005	44 (25/25)	44.00	0.005
p.hat700-3	62	62 (25/25)	62.00	0.009	62 (25/25)	62.00	0.012	62 (25/25)	62.00	0.009
p.hat1500-1	12*	12 (25/25)	12.00	10.165	12 (25/25)	12.00	11.377	12 (25/25)	12.00	10.006
p.hat1500-2	65	65 (25/25)	65.00	0.035	65 (25/25)	65.00	0.042	65 (25/25)	65.00	0.037
p.hat1500-3	94	94 (25/25)	94.00	0.145	94 (25/25)	94.00	0.143	94 (25/25)	94.00	0.129
Total			1730.34	834.77		1730.8	1176.831		1730.6	846.145

参考文献

- 1) M. Garey, and D. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness," Freeman, New York, 1979.
- 2) 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.
- 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 approx-

5. IKLS の性能評価実験

本節では、4.3 節で示した 6 種類の Kick の相違による IKLS の探索性能について検討を行う。

5.1 実験の詳細

Kick の相違による IKLS の探索性能を評価するために、6 種類の Kick をそれぞれ有する 6 タイプの IKLS 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)¹³⁾ から大規模もしくは厳密解の算出が困難な 35 グラフとする。IKLS の終了条件は、LS 回数 $n \times 100$ 回時もしくは既知の最良解 (BR) 算出時とし、試行回数は各問題例に対して 25 回とする。全ての IKLS 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.4.1.2) 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 7 上で行う。

5.2 実験結果

結果を表 1 に示す。DIMACS benchmarks の欄には、問題例名 (Instance) と既知の最良解 (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。IKLS の欄には、異なるタイプの Kick を有する 6 種類の IKLS の結果を示しており、各 IKLS の 25 回試行中に得られた最良解値の最大値 (Best), 平均値 (Avg), 最良解を得るまでの平均計算時間 (Time) をそれぞれ示している。また、表の下にある総合計 (Total) は Avg の値を総合計したものと Time の値を総合計したものである。

結果より、LEC-Kick, 適応的 1-Kick, 適応的 2-Kick, 適応的 3-Kick は他と比べて良好な結果を示している。また、HEC-Kick は他の Kick 法と比べて全体的に劣る結果を示した。RAC-Kick は C2000.9 や C1000.9 等のグラフで良い結果を示しているが、brock や keller6 の問題例等で劣る結果を示した。

LEC-Kick は keller6 の問題例において 25 回の探索の全てで最適解を発見している。また、LEC-Kick は keller6 の問題例において最適解発見までの Time が 3.961 秒となっており、他の手法と比べても非常に高速に最適解に到達している。他方、HEC-Kick は keller6 の問題例において、他の手法と比べて著しく悪い結果を示している。HEC-Kick がこのような結果を示した背景には次のことが考えられる。HEC-Kick は検証した 6 種類の Kick 法の中では多様性のある探索を行うことが難しい手法であり、keller6 の問題例は地形解析の結果、大谷構造とはならずより多様性のある探索が求められるためである。また、p_hat1500-1 の問題例においては、全ての Kick 法とも 25 回の試行中 25 回とも最適解を算出しているが、RAC-Kick は Time が 3.892 秒となっており、他の手法と比べてかなり高速に最適解に到達している。これらのことより問題例の違いによって、Kick の性能および最適解に到達するまでの時間に影響があることが明らかとなった。

以上のことを踏まえて次のことが推測できる。我々の調査によると、MCP に対する地形解析の結果¹⁰⁾ は、大谷構造とならない問題例が多数を占めた。これは、局所探索法で得られた局所最適解群が、探索空間内で大域最適解に向って分布していないため、局所探索を基にしたメタ戦略アルゴリズムにおいては、より多様性のある探索が必要であることを示唆している。検討した 6 つの Kick 法の中で LEC-Kick は最も多様性のある探索が可能のため、最も良好な結果を示したと考えられる。

6. むすび

本論文では、MCP に対する IKLS を示し、その内部に導入可能な複数の Kick の相違による探索性能を調査した。異なる 6 種類の Kick を有する 6 タイプの IKLS の探索性能の比較検討において、LEC-Kick, 適応的 1-Kick, 適応的 2-Kick, 適応的 3-Kick を有する IKLS が全体的に比較的良好な結果を示した。

以下では、今後の課題・検討事項について記述する。

実験の結果より、それぞれの Kick には種々の問題例に対して違う有効性があり、これらの Kick を効率良く使うことによって、探索性能のさらなる向上が期待できる。

Performance Comparison On Kick Methods of Iterated k -opt Local Search for Maximum Clique Problem

Akinori KOMURA, 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, Kita-ku, Okayama 700-0005, Japan.

(Received September 30, 2009; accepted November 5, 2009)

Many metaheuristics have been proposed for combinatorial optimization problems. One of the typical metaheuristics is well known to be iterated local search (ILS). In many cases, ILS has been shown to be capable of finding (near-) optimum solutions for difficult combinatorial optimization problems.

In this paper, we investigate six Kick methods in which can be used in ILS for the maximum clique problem (MCP). ILS with each of the Kick methods is evaluated on DIMACS benchmark graphs. The result shows that the performance of ILS strongly depends on a difference between the Kick methods.

Keywords: combinatorial optimization; maximum clique problem; iterated local search.

- imation 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) S. Lin and B. Kernighan, “An effective heuristic algorithm for the traveling salesman problem,” Operations Research, vol.21, pp.498–516, 1973.
 - 7) B. Kernighan and S. Lin, “An efficient heuristic procedure for partitioning graphs,” Bell System Technical Journal, vol.49, pp.291–307, 1970.
 - 8) P. Merz and K. Katayama, “Memetic algorithms for the unconstrained binary quadratic programming,” BioSystems, vol.78, no.1–3, pp.99–118, 2004.
 - 9) K. Katayama, M. Sadamatsu, and H. Narihisa, “Iterated k -opt Local Search for the Maximum Clique Problem,” Proc. of Seventh European Conference on Evolutionary Computation in Combinatorial Optimisation (EvoCOP-2007), LNCS 4446, Springer-Verlag, pp. 84–95, Valencia, Spain, April 11–13, 2007.
 - 10) 上野伸郎, 濱本明宏, 片山謙吾, 成久洋之, “最大クリーク問題に対する地形解析,” 平成 16 年度電気・情報関連学会中国支部第 55 回連合大会講演論文集, p. 376, Oct. 16, 2004.
 - 11) 幸村明典, 片山謙吾, 南原英生, 成久洋之, “最大クリーク問題に対する局所探索法の Kick 法の比較,” 平成 20 年度電気・情報関連学会中国支部第 59 回連合大会講演論文集, p. 392, 鳥取, Oct. 25, 2008.
 - 12) 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.
 - 13) 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.
 - 14) 片山謙吾, 谷 昌史, 成久洋之, “バイナリー 2 次計画問題の地形解析と遺伝的局所探索の性能,” 電子情報通信学会論文誌 (A), vol. J84-A, no. 10, pp. 1258–1271, 2001.
 - 15) 貞松政史, 片山謙吾, 南原英生, 成久洋之, “最大クリーク問題に対する反復局所探索法の検討,” 平成 18 年度電気・情報関連学会中国支部第 57 回連合大会講演論文集, p.173, 岡山, Oct. 21, 2006.
 - 16) 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.
 - 17) P.R.J. Östergård, “A fast algorithm for the maximum clique problem,” Discrete Applied Mathematics, vol.120, no.1–3, pp.197–207, 2002.
 - 18) J. Abello, P. Pardalos, and M. Resende, “On maximum clique problems in very large graphs,” External Memory Algorithms and Visualization, J. Abello and J. Vitter, (Eds.), DIMACS Series on Discrete Mathematics and Theoretical Computer Science, vol. 50, pp.119–130, American Mathematical Society, 1999.
 - 19) P. Hansen and N. Mladenović and D. Urošević, “Variable neighborhood search for the maximum clique,” Discrete Applied Mathematics, vol. 145, No. 1, pp. 117–125, 2004.
 - 20) R. Battiti and M. Protasi, “Reactive local search for the maximum clique problem,” Algorithmica, vol.29, no.4, pp.610–637, 2001.
 - 21) W. Pullan and H.H. Hoos, “Dynamic Local Search for the Maximum Clique Problem,” Journal of Artificial Intelligence Research, vol. 25, pp. 159–185, 2006.