

最大クリーク問題に対する頂点次数を考慮しない k -opt 局所探索法 A new k -opt local search for the maximum clique problem

渡邊 好幸[†]片山 謙吾[‡]南原 英生[‡]西原 典孝[‡]

Yoshiyuki Watanabe

Kengo Katayama

Hideo Minamihara

Noritaka Nishihara

1. まえがき

最大クリーク問題 (Maximum Clique Problem, MCP) とは, V を頂点の集合, E を頂点間の辺の集合とする無向グラフ $G = (V, E)$ が与えられた時, 頂点数が最大となるクリークを求める問題である. MCP に対する強力なメタ戦略アルゴリズムの多くは, クリークという部分グラフの特徴から, グラフ上の頂点の次数を考慮し解を改善する方法が一般的である. この考え方にもとづき, 我々は可変深度探索 (Variable Depth Search, VDS) のアイデアにもとづく k -opt 局所探索法 (k -opt local search, KLS) [1, 2, 3] を提案している. この解法を DIMACS ベンチマークグラフに適用した結果, 多くのグラフ例題に対して非常に有効であることを示している [2]. その一方で, KLS のように次数の大きい頂点を逐次選択する一般的な解法では, より大きなクリークの発見をしづらくなった例題も存在する. また, それらのグラフ例題に KLS を適用した結果, 他のグラフ例題に比べ良好な解を算出しにくいことも確認している. MCP における局所探索法は, 解 (クリーク) を改善 (拡大) する上で極めて重要な要素であるにもかかわらず, 頂点次数以外の情報を考慮し解を改善する方法については, あまり研究がなされていないのが現状である.

そこで本研究では, 従来の KLS のように, 頂点の次数を考慮し探索するのではなく, 頂点がクリークに含まれた回数 (頂点包含回数とよぶ) をもとに探索する, 新たな k -opt 局所探索法を提案する. 本論文では, この頂点包含回数にもとづく KLS を「Counting vertices Contained in the Clique based KLS」(C3KLS) とよぶ. この C3KLS を繰り返し行うマルチスタート法 [4] として適用し, 提案法の性能を検討する. 実験の結果, 従来の KLS に比べ, グラフ例題によっては高い探索性能が得られることを示す.

2. VDS にもとづく従来の k -opt 局所探索法

MCP に対する k -opt LS (KLS) [1] は, 可変深度探索 (VDS) のアイデアにもとづく局所探索法である. 以下, 基本アルゴリズムについて述べる. KLS の各反復における探索 (k -opt 近傍探索) は, 与えられた現在のクリーク (初期解) に対して, 連鎖的に複数個の頂点をクリークに追加 (Add) またはクリークから削除 (Drop) する操作によって構成する. まず, 現在のクリーク CC からそれらの操作によって生成可能な近傍解の集合を得る. その近傍解の集合から最良解 CC_{best} (すなわち, k 回の Add・Drop 移動操作により得られた最良解) を選び, その最良解 CC_{best} を次反復の初期解 CC とする. この一連の処理を各反復で対象となる k -opt 近傍内に良好なクリークが存在しなくなるまで反復する (図 1). 上述の KLS の k -opt 近傍探索では, 頂点を Add または Drop する際に, 頂点次数または頂点次数にもとづく頂点集合のサイズを考慮する. 以下, 詳細に述べる. Add フェーズ (Line 4-8) は, 現在のクリーク CC に追加可能な頂点集合 PA から, 頂点 v の部分グラフ $G(PA)$ 内

```

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

図 1 MCP に対する k -opt LS の疑似コード

```

C3-based-k-opt-Local-Search( $CC, PA, CV$ )
begin
  repeat
    1  $CC_{prev} := CC, D := CC_{prev}, P := \{1, \dots, n\}, g := 0, g_{max} := 0;$ 
    2  $repeat$ 
    3   if  $PA \cap P \neq \emptyset$  then //Add Phase
    4     find a vertex  $v$  with  $\min_{v \in PA} \{CV(v)\}$ 
    5     if multiple vertices with the same size are found
    6       then select one vertex  $v$  among them randomly;
    7      $CC := CC \cup \{v\}, g := g + 1, P := P \setminus \{v\};$ 
    8     if  $g > g_{max}$  then  $g_{max} := g, CC_{best} := CC, update CV(v);$ 
    9     else //Drop Phase( $|PA \cap P| = 0$ )
    10      find a vertex  $v$  with  $\min_{v \in CC} \{CV(v)\}$ 
    11      if multiple vertices with the same size are found
    12        then select one vertex  $v$  among them randomly;
    13       $CC := CC \setminus \{v\}, g := g - 1, P := P \setminus \{v\};$ 
    14      if  $v$  is contained in  $CC_{prev}$  then  $D := D \setminus \{v\};$ 
    15    endif
    16    update  $PA;$ 
    17    until  $D = \emptyset;$ 
    18    if  $g_{max} > 0$  then  $CC := CC_{best}, update CV(v)$  else  $CC :=$ 
    19     $CC_{prev};$ 
    20    until  $g_{max} \leq 0;$ 
    21    return  $CC;$ 
    22  end;
```

図 2 MCP に対する頂点包含回数にもとづく k -opt LS の疑似コードにおける次数 $deg_G(PA)(v)$ が最大となる頂点 v を選択する処理である. 同値の次数を有する頂点が複数存在した場合には, それらの頂点からランダムに選択する. この処理を PA が空集合になるまで繰り返す. 次いで, Drop フェーズ (Line 9-14) は, PA が空集合となっている (CC が拡大不可能な) 場合に実行する. Drop フェーズでは, CC から頂点 v を削除する際, 次の繰り返し時点で, PA のサイズをより大きくする頂点 v を削除する. Drop フェーズは CC から削除できる頂点がない, もしくは少なくとも 1 つ以上の頂点が追加可能になるまで繰り返される.

3. 頂点包含回数にもとづく新たな k -opt 局所探索法

頂点包含回数にもとづく k -opt 局所探索法 (C3KLS) について記述する. C3KLS は, 頂点次数を考慮しない KLS であり, その点で上述した従来の KLS と大きく異なる. 従来の KLS では, Add フェーズおよび Drop フェーズにおいて, 頂点次数の情報にもとづきクリークを探索したが, C3KLS では, Add および Drop の両フェーズにおいて, それまでの探索で保持されている頂点包含回数の情報にもとづき探索を行う. ここで頂点包含回数とは, 与えられたグラフ G における各頂点 $v \in V$ が, 探索中のクリークに含まれた回数のことである. 以降, 頂点包含回数の情報を $CV(v)$ とする.

[†]岡山理科大学大学院 工学研究科[‡]岡山理科大学 工学部 情報工学科

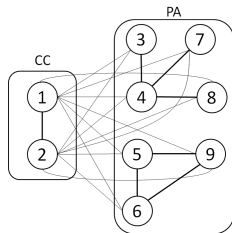


図 3 頂点選択をする際のグラフ例

表 1 グラフ $G(PA)$ の頂点の情報

$v \in PA$	$deg_{G(PA)}(v)$	$CV(v)$
3	1	2
4	3	4
5	2	1
6	2	2
7	1	2
8	1	4
9	2	3

図 2 に C3KLS の擬似コードを示す。各頂点の CV の情報は予め与えられるものとする（全ての頂点 $v \in V$ における $CV(v)$ の初期値は 0 とする）。従来の KLS との大きな違いは、図 2 の Add フェーズの Line 5 と Drop フェーズの Line 10 である。Add フェーズおよび Drop フェーズにおいて、各頂点 $v \in PA$ における $CV(v)$ が最小値となる頂点 v を見つける処理となっており、いづれも最小値を選択する。これらをそれぞれ MinAdd, MinDrop とよぶ。また、Add フェーズと Drop フェーズにおいて $CV(v)$ が最大値となる頂点 v を選択する方法をそれぞれ MaxAdd, MaxDrop とよぶ。

$CV(v)$ の増加処理は、図 2 の Line 8 において CC_{best} が更新された際に CC_{best} に含まれる頂点 v の $CV(v)$ を +1 する。また、探索の多様性を考慮し $CV(v)$ の減算処理を行う。これは Line 17 において、 $G \setminus CC$ となる全頂点 v の $CV(v)$ を -1 する。

表 1 および図 3 を用いて C3KLS と従来の KLS との Add フェーズにおける違いについて説明する。表 1 の左の欄から、図 3 における部分グラフ $G(PA)$ に含まれる頂点 $v \in PA$ 、部分グラフ $G(PA)$ における各頂点の次数 $deg_{G(PA)}(v)$ 、 $v \in PA$ それぞれの頂点包含回数 $CV(v)$ の一例を示し、近傍探索中に表 1 および図 3 のようになったと仮定する。このとき、従来の KLS では $deg_{G(PA)}(v)$ が最大となる頂点 v を選択するため、表 1 の $deg_{G(PA)}(v)$ 欄より v_4 を選択する。提案法において MinAdd を適用した場合は、 $CV(v)$ が最小となる頂点 v を選択するため、表 1 の $CV(v)$ 欄より v_5 を選択する。

4. 実験結果

C3KLS の性能を評価するために、従来の KLS との比較実験を行った。実験は、各 KLS をマルチスタート法の枠組みにそれぞれ導入した方法を、DIMACS ベンチマークグラフの代表的な問題例に適用し評価した。マルチスタート法は、検討するグラフ問題それぞれに対し、グラフ G の頂点群 V からランダムに 1 頂点を選択し、これを初期解（クリーク）として局所探索を実行する一連の処理を 100 回繰り返す。これを 1 試行とし、100 試行を行った。また、本実験において C3KLS では頂点包含回数を 1 試行ごとにリセットし、次試行に頂点包含回数は引き継がれないものとした。C3KLS および従来の KLS は C 言語を用いてコード化し、全ての実験は Lenovo D20 ThinkStation (Xeon E5645 2.40GHz, 12GB RAM) 上で実行した。

表 2 に C3KLS (MinAdd, MinDrop)、表 3 に C3KLS (MaxAdd, MinDrop) の実験結果をそれぞれ示し、従来の KLS の実験結果と比較する。表 2 および表 3 の左の欄から、問題例名 Name、既知の最良解値 BR を示し、提案法 C3KLS および従来の KLS でのマルチスタート法 100 試行中に得られた最良解値とその回数、解の平均値 Avg をそれぞれ示す。なお、C3KLS 欄の太字は従来の KLS と比べて良い結果であることを示し、印*は従来の KLS と同値の結果を示す。

表 2 C3KLS (MinAdd, MinDrop) と従来の KLS の実験結果

Name	BR	C3KLS		従来の KLS	
		Best	Avg	Best	Avg
C125.9	34	32 (52/100)	31.50	34 (100/100)	34.00
C250.9	44	40 (2/100)	38.42	44 (100/100)	44.00
C500.9	57	51 (1/100)	47.10	57 (100/100)	57.00
C1000.9	68	57 (10/100)	55.67	68 (42/100)	67.42
brock200.2	12	12 (100/100)	12.00	11 (100/100)	11.00
brock200.4	17	* 17 (66/100)	16.66	17 (3/100)	16.03
brock400.2	29	* 29 (6/100)	23.93	29 (100/100)	29.00
brock400.4	33	* 33 (66/100)	29.02	33 (6/100)	25.48
brock800.2	24	* 21 (16/100)	20.15	21 (100/100)	21.00
brock800.4	26	26 (2/100)	20.17	21 (100/100)	21.00
gen200.p0.9.44	44	39 (1/100)	36.62	44 (100/100)	44.00
gen200.p0.9.55	55	* 55 (83/100)	54.41	55 (100/100)	55.00
gen400.p0.9.55	55	46 (6/100)	43.96	51 (11/100)	53.19
gen400.p0.9.65	65	* 65 (1/100)	49.87	65 (100/100)	65.00
gen400.p0.9.75	75	* 75 (2/100)	58.67	75 (100/100)	75.00
p.hat1500-1.clq.b	12	* 12 (82/100)	11.82	12 (100/100)	12.00
p.hat1500-2.clq.b	65	59 (1/100)	54.85	65 (100/100)	65.00
p.hat1500-3.clq.b	94	82 (1/100)	75.07	94 (100/100)	94.00

表 3 C3KLS (MaxAdd, MinDrop) と従来の KLS の実験結果

Name	BR	C3KLS		従来の KLS	
		Best	Avg	Best	Avg
C125.9	34	* 34 (50/100)	32.28	34 (100/100)	34.00
C250.9	44	* 44 (2/100)	40.56	44 (100/100)	44.00
C500.9	57	55 (1/100)	50.71	57 (100/100)	57.00
C1000.9	68	66 (4/100)	59.88	68 (42/100)	67.42
brock200.2	12	12 (88/100)	11.80	11 (100/100)	11.00
brock200.4	17	* 17 (27/100)	16.24	17 (3/100)	16.03
brock400.2	29	* 29 (1/100)	23.76	29 (100/100)	29.00
brock400.4	33	* 33 (45/100)	28.08	33 (6/100)	25.48
brock800.2	24	* 21 (9/100)	20.09	21 (100/100)	21.00
brock800.4	26	* 21 (13/100)	20.12	21 (100/100)	21.00
gen200.p0.9.44	44	40 (20/100)	38.01	44 (100/100)	44.00
gen200.p0.9.55	55	* 55 (3/100)	47.54	55 (100/100)	55.00
gen400.p0.9.55	55	* 51 (26/100)	47.17	51 (11/100)	53.19
gen400.p0.9.65	65	55 (1/100)	48.32	65 (100/100)	65.00
gen400.p0.9.75	75	71 (1/100)	49.97	75 (100/100)	75.00
p.hat1500-1.clq.b	12	* 12 (94/100)	11.94	12 (100/100)	12.00
p.hat1500-2.clq.b	65	64 (51/100)	59.81	65 (100/100)	65.00
p.hat1500-3.clq.b	94	* 94 (3/100)	84.32	94 (100/100)	94.00

表 2 および表 3 の結果より、C3KLS は brock 以外のグラフ問題において、従来の KLS と比べて解が劣る結果となった。これらのグラフ問題に対しては、従来の KLS のような頂点の次数を考慮した解法が有効であると考えられる。しかしながら、brock のグラフ問題に対しては、従来の KLS よりも C3KLS の方がよりよい解を算出している場合が多い。特に、問題例 brock200.2 では、従来の KLS において、BR に一致する解が算出できなかったのに対して、表 2 に示す MinAdd 版の C3KLS では 100 試行すべてにおいて算出可能であった。また、問題例 brock400.4 においても、従来の KLS では 100 試行中 6 試行しか BR に一致する解を算出できなかったのに対して、提案法である C3KLS の MinAdd 版では 100 試行中 66 試行、表 3 に示す MaxAdd 版では 45 試行で算出できた。その結果、Avg の値もそれぞれ大幅に改善されており、C3KLS の有効性が確認できる。また、問題例 brock800.4 においても、従来の KLS では算出できなかったサイズ 26 のクリークが提案法では算出可能であった (表 2)。以上の結果より、提案する C3KLS は、従来の KLS に比べて brock のグラフ問題に対してより高い探索性能が得られ、有効であることを示した。

5. むすび

本論文では、MCP に対する新たな k -opt LS として頂点包含回数にもとづく C3KLS を提案した。C3KLS は従来の KLS のように頂点の次数を考慮するのではなく、頂点がクリーク（解）に含まれた回数をもとに探索する解法である。比較実験により、C3KLS は brock のグラフ問題に対して有効であることを示した。今後の課題として、より様々なグラフ問題に対して柔軟に対応できるアルゴリズムの設計を目的とし、提案した C3KLS と従来の KLS の 2 つの解法を探索の状況に応じて制御する戦略的制御 (Strategic Controller, SC) [3] を導入したメタ戦略アルゴリズムの研究が考えられる。

参考文献

- [1] K. Katayama, A. Hamamoto, H. Narihisa, "An Effective Local Search for the Maximum Clique Problem," Information Processing Letters, vol. 95, no. 5, pp. 503–511, 2005.
- [2] K. Katayama, M. Sadamatsu, and H. Narihisa, "Iterated k -opt Local Search for the Maximum Clique Problem," Proc. of 7th European Conference on Evolutionary Computation in Combinatorial Optimization, pp. 84–95, April 2007.
- [3] K. Katayama, A. Kohmura, K. Kohmoto, H. Minamihara, "Memetic Algorithm with Strategic Controller for the Maximum Clique Problem," Proc. of the 2011 ACM Symposium on Applied Computing (SAC-2011), Taichung, Taiwan, vol. 2, pp. 1062–1069, March 21–24, 2011.
- [4] 柳浦隆憲, 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.