

最大クリーク問題に対する進化的アルゴリズム

片山 謙 吾^{†1} 貞松 政 史^{†1}
南原 英 生^{†1} 成久 洋 之^{†1}

組合せ最適化問題の代表例である最大クリーク問題 (maximum clique problem, MCP) に対して, これまでに遺伝的アルゴリズム等の進化的アプローチが多く提案されてきた. 本研究では, 高性能な進化的アルゴリズムの設計を目指し, まず, MCP における探索空間内の地形を解析する. その結果, DIMACS ベンチマークグラフ群の多くの例題において, MCP に対する k -opt 局所探索法によって算出される局所最適解群は, 真の最適解との間に相関が見られない場合が多いことを示す. 本論文では, このような悲観的な地形解析結果に対応する Memetic アルゴリズム (memetic algorithm, MA) を提案し, 最先端の進化的アプローチとの比較を通して, 本 MA の有効性を示す.

An Effective Evolutionary Algorithm for the Maximum Clique Problem

KENGO KATAYAMA,^{†1} MASASHI SADAMATSU,^{†1}
HIDEO MINAMIHARA^{†1} and HIROYUKI NARIHISA^{†1}

Many evolutionary approaches have been proposed for the maximum clique problem (MCP). To design a high-performance evolutionary algorithm for the MCP, we first analyze the landscape of local optima that can be obtained by the k -opt local search (KLS) based on the variable depth search. The results show that the search space is very pessimistic on many of the DIMACS benchmark graphs. This paper presents an effective memetic algorithm (MA) — an evolutionary algorithm incorporating KLS — for the MCP, and shows the effectiveness of the MA through comparisons of state-of-the-art evolutionary approaches to the problem.

^{†1} 岡山理科大学
Okayama University of Science

1. はじめに

多くの組合せ最適化問題を対象にして, 遺伝的アルゴリズム (Genetic Algorithm, GA)¹⁰⁾ や Ant Colony Optimization⁶⁾ に代表される進化計算法の研究がさかんである. 進化計算手法の最大の特徴は, 複数の解を用いた確率的な多点探索によって, 高い大域的探索能力を実現することである. 特に GA は, 交叉, 突然変異, 選択の遺伝的操作を備え, NP-困難⁸⁾ である組合せ最適化問題に広く適用できる枠組みを提供する. 一般的に GA 等の進化計算手法は, 大域的探索能力に優れている一方で, 局所的な探索能力の欠如が指摘されている. したがって, より良好な探索性能を実現するためには, これらの相反する探索をバランス良く保持することが重要である. そのような探索を実現する進化的アルゴリズムとして, Memetic アルゴリズム (Memetic Algorithm, MA)^{14),16),22)} が注目されている. MA は, GA と局所探索の融合化手法であり, 多くのメタ戦略^{9),26)} の結果を超越する優れた性能が報告されている^{11),16)}.

本論文では, 代表的な組合せ最適化問題の1つである最大クリーク問題 (Maximum Clique Problem, MCP)⁵⁾ を対象にして, MCP の探索空間内の地形⁴⁾ を解析したうえで, その結果に基づき設計した MA を提案する. 地形解析の結果, DIMACS ベンチマーク問題例群¹⁵⁾ の多くの MCP グラフにおいて, 局所探索法¹⁸⁾ によって導かれる局所最適解群は, 多くの場合, 真の最適解との間に相関がないことを示す. これは, 局所最適解群が真の最適解に向かって分布する, いわゆる大谷構造⁴⁾ とは異なる探索空間上の構造を示唆しており, このような探索空間内を最適解に向けて探索することは容易ではない (3 章参照).

提案する MA は, 地形解析の結果を受け, 交叉操作を利用せず, より広範囲の探索を実現する突然変異操作と解集団の多様性に配慮した選択操作を採用し, MCP に対する k -opt 局所探索法 (k -opt Local Search, KLS)¹⁸⁾ によって, 効果的な MA 探索を実現する.

提案法の探索性能を評価するために, DIMACS ベンチマークの MCP グラフ群を対象にして, MCP に対する最先端の進化計算手法である, Singh らによる Heuristic based Steady-State Genetic Algorithm (HSSGA)²⁴⁾, Zhang らの Evolutionary Algorithm with Guided Mutation (EA/G)³⁰⁾, Battiti らによる Reactive Evolutionary Algorithm (R-EVO)²⁾, Solmon らによる Ant Colony Optimization²⁵⁾ の報告結果と比較検討する. その結果, 本 MA は, それらの進化計算手法に比べ, 平均的に良好なクリークを効率良く算出できることを示し, 提案法の有効性を示す.

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 $E \subseteq V \times V$ が与えられたとき, $G = (V, E)$ を無向グラフという. 特に, すべての 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 $K \subseteq V$ による誘導部分グラフ $G(K) = (K, E \cap K \times K)$ が完全グラフのとき, すなわち, $\forall i, j \in K, i \neq j$ に対して $(i, j) \in E$ であるとき, K をクリークと呼ぶ. 最大クリーク問題 (MCP)⁵⁾ とは, 与えられたグラフ G に含まれるクリークの中で, 次の評価関数

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

を最大にするクリーク K 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現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⁵⁾, NP-困難⁸⁾ である. したがって, 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. 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⁷⁾, そのほかにも MCP の難しさを示す否定的な報告がなされている^{1), 12)}. したがって, MCP や MCP に関連する最大独立集合問題²³⁾ 等を多項式時間で近似的に解く解法の重要性が高まっている.

3. MCP に対する地形解析

4 章で述べるように, MA で保持される解群は局所最適解の集合である. また, MA は問題領域の知識を積極的に利用する²²⁾ ことから, 探索空間における局所最適解の分布の形状 (地形) を考慮することで, さらに高性能な MA を設計できる可能性がある. 高性能な MA や関連するメタ戦略の設計に際し, 探索空間における構造の複雑さの尺度である地形を解析したうえで, アルゴリズムの設計に役立てる研究^{4), 13), 17), 20), 21), 27)–29)} が行われている.

ある対象問題の探索空間内の形状 (地形) が真の最適解に向かって 1 つの大きな谷となるような構造は, 一般に, 大谷構造 (big valley structure)⁴⁾ と呼ばれている. 対象問題が大谷構造のような構造化された空間を持つならば, 探索の集中化と多様化の相反する探索戦略をバランス良く有するメタ戦略によって最適解への接近が可能であると考えられている. その観点から, 組合せ最適化問題に対するメタ戦略の解法の多くでは, 探索空間内の形状が大谷構造の仮定のもとで, アルゴリズムの設計がなされている場合が多い.

本章では, MCP に対する地形解析を通して, 大谷構造とは異なる探索空間上の構造が多

くの問題例で観測されることを示す.

3.1 地形解析の要領

文献 17), 21), 28) 等の研究と同様に, 地形を $L = (X, f, d)$ で表現する. なお, X は探索空間, f は目的関数, d は探索空間上で定義された距離である. これは, 地形の複雑さと最適化の難しさとの関連付けを可能にするが, 対象とする最適化問題に対する探索法で扱う評価関数, 解の表現, 距離の定義に依存する.

本解析では, 評価関数 f を 2 章で定義した f_{MCP} とする. 解は, 長さ $n (= |V|)$ の 0-1 ビットで表現される 1 次元ベクトルとする. すなわち, 得られたクリークの各頂点に対応するビットには 1, それ以外のビットは 0 とする ($X = \{0, 1\}^n$). 距離 d は空間 X における 2 つの解のハミング距離とする.

解析対象のグラフとして, MCP の代表的なベンチマークグラフである DIMACS の問題例群¹⁵⁾ を利用する. DIMACS ベンチマークの問題例は, ランダムグラフをはじめ, さまざまなタイプのグラフが準備されており, 多くの研究者によって研究対象とされている. DIMACS ベンチマークグラフにおける既知の最良解は, 多くの研究者によるメタ戦略に基づく解法によって算出されてきたが, 現在までに知られている最良解よりも良好な解は報告されていないことから, 真の最適解にきわめて近い解, もしくは実質的に (大域的) 最適解であると考えられる. そこで, 本解析では, 既知の最良解を (便宜上) 最適解として扱う. なお, 対象グラフにおける最適解 (既知の最良解) の評価値に一致する解は, k -opt 局所探索法 (KLS)¹⁸⁾ および KLS を拡張した解法によってあらかじめ算出した.

解析方法として, 対象グラフ G の頂点群 V からランダムに選択した 1 つの頂点を初期解 (クリーク) とする KLS の適用を 10 万回行い, 得られた複数の局所最適解を利用して, 探索空間内の地形を解析する. また, 対象のグラフに最適解が複数存在する場合は, それらすべてについて距離を求め, その平均値を利用した. その理由は, 一部の探索空間に偏った解析を避け, 対象グラフの平均的な構造を観測するためである.

3.2 解析結果

表 1 は, 顕著な特徴が観測された 13 例題に対する地形解析の結果である. 左から, 問題例名, 最適解値 Opt (本解析で確認された) 最適解の個数 $OptN$, 10 万回の KLS により算出された異なった局所最適解の数 $LOptN$, 最適解と得られた局所最適解の平均距離 \bar{d}_{opt} , 局所最適解どうしの平均距離 \bar{d}_{alo} および最適解と局所最適解の相関係数 ρ_{opt} , 局所最適解どうしの相関係数 ρ_{alo} を示した. なお, 表 1 の結果を解釈する際の注意事項として, \bar{d}_{opt} および \bar{d}_{alo} の値が Opt の 2 倍に近づくと, 最適解および局所最適解どうしは同じ頂点を共

3717 最大クリーク問題に対する進化的アルゴリズム

表 1 DIMACS ベンチマークグラフに対して 10 万回の k -opt 局所探索法により得られた局所最適解の距離解析結果と相関係数

Table 1 Results of landscape analysis on DIMACS benchmark graphs.

instance	Opt	$OptN$	$LOptN$	\bar{d}_{opt}	\bar{d}_{alo}	ρ_{opt}	ρ_{alo}
C2000.5	16	77	31358	29.28	26.88	0.98	0.99
C4000.5	18	11	45734	32.58	29.29	0.96	0.99
DSJC500.5	13	50	9754	23.24	21.36	0.85	0.96
DSJC1000.5	15	18	18843	26.84	24.27	0.86	0.98
MANN_a45	345	2	100000	458.82	457.03	0.01	0.94
MANN_a81	1100	7	100000	1468.13	1465.33	0.01	0.86
brock400_2	29	1	31057	48.82	38.87	0.35	0.45
brock800_2	24	1	31470	41.47	34.64	0.57	0.85
keller5	27	673	96271	48.94	46.42	0.99	0.99
keller6	59	14	100000	107.04	98.41	0.94	1.00
p_hat1500-1	12	1	17690	20.82	18.24	0.47	0.97
p_hat1500-2	65	448	63407	46.50	51.91	-0.64	-0.68
p_hat1500-3	94	536	94305	57.82	64.06	-0.71	-0.76

有することが少なくなる．また、 ρ_{opt} 、 ρ_{alo} がともに -1 に近いとき、最適解と局所最適解どうしに強い相関があることを示し、大谷構造の存在を示唆する結果となることに注意されたい²¹⁾．

視覚的に地形を確認するために、13 例題の中から特に特徴的であった (a) p_hat1500-3、(b) MANN_a81 および (c) keller6 の結果を図 1 に示す．各図 (a)(b)(c) は、局所最適解と大域最適解との距離（左側）、局所最適解どうしの平均距離（右側）の結果であり、それぞれを目的関数値との関係でプロットした．なお、横軸はプロット図が確認しやすいと思われる距離の範囲で表示した．縦軸は大域最適解値と局所最適解値の差を与えており、値が 0 に近づくほど良質な局所最適解であることを意味する．

表 1 の結果より、強い相関が観測できたのは、p_hat1500-2 と 3 のみであった．なお、異なる頂点サイズの p_hat 例題群でも同様の結果を観測している．この結果を視覚的に確認すると、図 1 (a) の結果となり、探索空間上で局所最適解群が 1 つのクラスタとなり、最適解に向けて分布している構造を確認できる．これは、大谷構造の存在を示唆する結果である．しかしながら、その他の多くの例題では、最適解と局所最適解どうしに強い相関は確認できなかった．それらの例題では、大谷構造とは異なる構造になっていることを示唆している（図 1 (b) および (c) の各図を参照）．特に、 ρ_{opt} および ρ_{alo} がともに 1 に近い結果は、表 1 に示すように DIMACS の多くの例題で観測された傾向である．図 1 (c)keller6 の結果

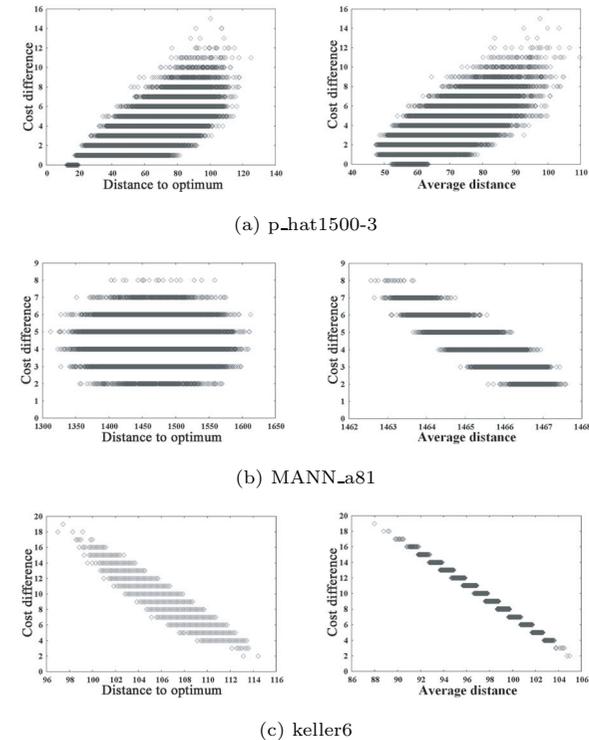


図 1 地形解析のプロット図

Fig. 1 Plot figures of landscape analysis.

は、その代表例であり、最適解との評価値の差が小さくなればなるほど、最適解と局所最適解の距離が遠く地形を確認できる．

以上より、少なくとも DIMACS ベンチマークのグラフ群においては、大谷構造の存在を示唆する例題群がきわめて少なく、大谷構造とは異なった探索空間上の構造が多くの場合に観測されたことを示した．

4. MCP に対する Memetic アルゴリズム

4.1 地形解析結果に基づく MA の設計方針

大谷構造は、非常に漠然とした探索空間の構造の 1 つである．無論、大谷構造が存在す

る問題（例）が解きやすいと短絡的にとらえることはできないが，大局的に探索空間を眺めた場合，大きなすり鉢状の形状となる．この形状の特徴として，質の良い解の周辺にはさらに上質の解が多く存在し，その傾向は最適解に向けて分布すると考えることができる．よって，大谷構造の存在は MA 等の探索アルゴリズムにとって，理想的な構造の 1 つとしてとらえることができるとともに，アルゴリズムの設計に際しても，比較的設計しやすい探索空間の構造と考えることができる．すなわち，解どうしの類似した部分情報をできるだけ継承する，新たな解の生成法（たとえば，交叉操作はその代表例である）によって，評価値の改善を可能にするのであれば，局所探索との併用等により，最適解への接近が比較的にスムーズに実行される構造と考えられる．

一方，本論文で対象とする MCP の場合，大谷構造とは異なる探索空間の構造を示唆しており，非常に悲観的なことに，最適解と局所最適解との評価値の差が小さくなればなるほど，最適解との距離が離れる傾向を多くのグラフで観測した．ここで，逆説的にとらえた場合を考える．本解析結果に反するように設計された進化計算法（大谷構造を前提とした進化計算法）の場合には，最適解が存在する領域とは異なる領域に探索をすすめる傾向が強くなり，最適解の算出は困難になると予測される（図 1(b), (c) 参照）．したがって，最適解の算出を MA によって目指す場合，大谷構造を前提としたアルゴリズム設計とは異なる方針をとらざるをえないと考えられる．その方針の 1 つは，探索の多様性を十分に高めることである．これとは相反して，たとえば一様交叉のような単純な交叉操作は，多様性を弱める方向に働く傾向があるため，MCP の多くの問題例に対しては有効な操作としてとらえることは難しい．局所最適化の能力に乏しいことから，局所探索法の代替にもなりにくい．加えて，表 1 に示したように， \bar{d}_{opt} および \bar{d}_{alo} の値が Opt の 2 倍に近い問題例が多いことを考慮すると，解どうしが同じ頂点を共有しない傾向が強いため，何らかの工夫を施さない限り，良好な評価値の実行可能解を生成することは期待できない．なお，文献 19) では，一様交叉に基づき，解の実行可能性を保つ方法が提案され，一定の成果が報告されているが，MCP に対する最近の進化計算研究の進展により，そのような単純な交叉操作によって十分な性能を得ることは困難であることが明らかにされつつある．以上をふまえ，提案する MA では，交叉操作を排除し，広範囲の領域へ探索を導く突然変異操作と解集団の多様性を高める選択操作，および，できるだけ良好な局所最適解（クリーク）の算出を目指し，KLS によって探索の集中化を具備する．よって提案法は，探索の多様化を突然変異と選択操作が行い，強い集中化は交叉ではなく，KLS によって実現する．

4.2 提案法 MA の流れ

Memetic アルゴリズム (MA) は，基本的に遺伝的アルゴリズム (GA) と局所探索法 (Local Search) のハイブリッド手法であり，遺伝的局所探索法 (Genetic Local Search) に類似した手法である^{14),16),22)}．

上述した設計方針に基づく提案法 MA の流れを図 2 に示す．なお，親集団 P_p および子集団 P_c のそれぞれの解集団のサイズを $PopSize$ とする．まず，解集団の初期化処理として， P_p のそれぞれにグラフ 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として与え，局所探索法を適用することで，局所最適解の集合を得る (Line 1~4)．その後，MA の主要処理として，現代の P_p の各解に突然変異操作および局所探索法を適用することで，新たな局所最適解群 P_c を生成し (Line 6~9)， P_p と P_c から $PopSize$ 個の解を選択することによって次世代の親集団 P_p を構成する (Line 10)．この主要処理を任意の世代数繰り返すことで MA の探索を実行する．以降，本 MA の構成要素である，局所探索法，突然変異操作および選択操作について記述する．なお，局所探索法および突然変異操作で施される処理は，実行可能領域の処理となり，各処理によって得られる解は実行可能解（クリーク）であることを保証する．

4.3 局所探索法

本 MA の局所探索の過程では，KLS¹⁸⁾ を用いる．

```

procedure MA( PopSize )
begin
1   for i = 1 to PopSize do begin
2     Pp[i] := Initialization( );
3     Pp[i] := LocalSearch( Pp[i] );
4   end;
5   repeat
6     for i = 1 to PopSize do begin
7       Pc[i] := Mutation( Pp[i] );
8       Pc[i] := LocalSearch( Pc[i] );
9     end;
10    Pp := Selection( Pp, Pc );
11  until terminate = true;
12  return best individual ∈ Pp;
end;

```

図 2 MCP に対する Memetic アルゴリズム
Fig. 2 The flow of our memetic algorithm for MCP.

```

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 max 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
12             the resulting  $|PA \cap P|$  is maximized;
13           if multiple vertices with the same size of
14             the resulting  $|PA \cap P|$  are found
15             then select one vertex  $v$  among them randomly;
16            $CC := CC \setminus \{v\}, g := g - 1, P := P \setminus \{v\};$ 
17           if  $v$  is contained in  $CC_{prev}$  then  $D := D \setminus \{v\};$ 
18         endif
19         update  $PA, OM,$  and  $deg_G(PA)(i), \forall i \in PA \cap P;$ 
20       until  $D = \emptyset;$ 
21       if  $g_{max} > 0$  then  $CC := CC_{best}$  else  $CC := CC_{prev};$ 
22     until  $g_{max} \leq 0;$ 
23     return  $CC;$ 
end;
```

図3 MCP に対する k -opt 局所探索法
Fig.3 The flow of k -opt local search for MCP.

KLS の擬似コードを図3に示す。ここで、図3で使用した記号を以下に定義する。 CC は与えられた現在のクリーク、 PA は CC に含まれる全頂点と隣接している頂点の集合、 OM は CC に含まれる全頂点のうち、ある1頂点以外の頂点とすべて隣接している頂点の集合、 $deg_{G(S)}(v)$ は V の部分集合 S からなる誘導部分グラフ $G(S)$ 内の各頂点 $v \in S$ の次数である。また、集合 P は移動候補頂点集合である。

KLSは、各反復において、 CC に対し、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（Line 4~8）、Drop移動（Line 9~13）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大きな近傍ととらえることで局所探索を行うアルゴリズムである。なお、 $P = \{1, \dots, n\}$ （Line 2）は、Add移動操作およびDrop移動操作の連鎖的な適用による近傍解の集合を生成する過程において、すでに追加

```

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

図4 突然変異操作
Fig.4 The flow of our mutation operator in MA.

または削除された頂点が再び追加・削除されることがないようにするために利用する。その他の詳細は文献18)を参照されたい。

4.4 突然変異

MCPに対する突然変異操作の最も簡単な方法の1つは、与えられたクリークからランダムにいくつかの頂点を削除する方法である。しかしながら、削除する頂点数をあらかじめ適切に決定することは一般に困難である。以下では、ユーザによる、そのようなパラメータ設定を必要としない突然変異操作を示す。

本突然変異は、対象グラフ G の頂点間の隣接関係に基づき、削除する頂点数を適応的に決定するパラメータレスの操作である。これは、2つのプロセス（例外処理と本処理）から構成される。図4にその擬似コードを示す。まず、例外処理（Line 1~5）について述べる。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 CC にまったく隣接していない場合、後に述べる本処理を適用することができない。この場合は、 $V \setminus CC$ 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する。

以下、突然変異の本処理について示す。グラフ G の $V \setminus CC$ において、 CC の頂点群と最低1頂点以上隣接する頂点集合から、隣接する数が最も少ない頂点 v を選択する（Line 6）。頂点 v となる候補が複数個存在する場合は、それらの候補群からランダムに1頂点を選択する（Line 7）。次いで、頂点 v とそれに隣接する CC の頂点群とで新たなクリークを構成する（Line 8）。以上の処理は、地形解析の結果を受けたものであり、突然変異後のクリーク

クは、与えられた CC に隣接した、比較的広範囲の領域に生成されるようになる。

なお、突然変異の本処理によって CC から削除された頂点群は、次回の KLS の探索における 1 回目の反復において、集合 P (図 3, Line 2) から除外する。これは、突然変異前の CC に戻ることを抑制するためであり、探索の多様化に配慮するものである。

4.5 選 択

地形解析の結果より、MCP は大域最適解と局所最適解、および局所最適解どうしに相関が見られない場合が多いため、解どうしの距離を考慮し、解集団内に多様性を持たせる必要があると考えられる。そこで、本 MA における次世代の解群の選択操作として次の方法を用いる。本選択の基本方針として、現世代の子集団 P_c と親集団 P_p の中から、クリークサイズ $|CC|$ の大きい解群を P_c の解優先で選び次世代の P_p とする。その際、 $|CC|$ が同値となる各解とのハミング距離が d 以下の解を含まないように次世代集団を構成することで、良質の解群を保持しつつ、集団の多様性を維持する。なお、 d はユーザによって設定可能であるが、MA 探索に最適な、または合理的な d の値をあらかじめ決定することは容易ではない。我々は、ユーザによるパラメータ設定の削減に配慮し、各世代の各解 $CC \in P_p$ に対する突然変異によって各 CC から削除された頂点数の平均値を d とする。したがって、 d は各世代において適応的に決定され、解集団内の多様性は、対象グラフ G の辺密度、MA で保持される解集団およびそのサイズの関係のもとで合理化される。

5. 実験結果

提案法 MA の探索性能を評価するために、MCP に対する最先端の進化計算手法である、Singh らによる Heuristic based Steady-State Genetic Algorithm (HSSGA)²⁴⁾、Zhang らの Evolutionary Algorithm with Guided Mutation (EA/G)³⁰⁾、Battiti らによる Reactive Evolutionary Algorithm (R-EVO)²⁾*1、Solnon らの Ant Colony Optimization (ACO)²⁵⁾ の結果との比較を行った。対象とする問題例は MCP の標準的なベンチマーク問題として知られている DIMACS ベンチマークグラフ¹⁵⁾ から、多くの研究者がアルゴリズム評価の対象とする 37 グラフとした。MA 実行のパラメータとして、解集団のサイズ $PopSize$ を 20、計算打ち切りの最大世代数を $|V| \times 5$ とした。これらのパラメータ値の設定は、対象問題例のすべてに対して共通とした。なお、既知の最良解の算出時に計算を打ち切

り、試行回数は各問題例に対して 10 回とした。本 MA 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。計算機は、hp workstation xw4300 (CPU: Pentium4 3.4 GHz, RAM: 4 GB) を用いた。文献 24) の実験環境は、CPU: Pentium3 1.0 GHz, RAM: 384 MB, 文献 30) では、CPU: AMD2400 2.0 GHz, 文献 2) では、CPU: Dual Xeon 3.4 GHz, RAM: 6 GB, 文献 25) では、CPU: Pentium 4 1.9 GHz である。

表 2 および表 3 は、本 MA の結果および比較手法 HSSGA²⁴⁾、EA/G³⁰⁾、R-EVO²⁾、ACO²⁵⁾ それぞれの報告結果である。Instance の欄には問題例名 Name と既知の最良解値 BR (*は最適解値であることが証明されている) を示した。各手法の欄には、複数回試行中に得られた最良解値 Best, 各試行で得られた最良解の平均値 Avg, それら最良解を得るまでの平均時間 time(s) を示した。表中の “ $< \epsilon$ ” は 0.001(s) 以下であったことを示す。なお、各手法の各問題例に対する試行回数は、ACO については 50 回、その他の手法はそれぞれ 10 回と報告されている。また、最下段の Average は、37 グラフに対する各手法の Best および Avg の結果の平均値を示した。なお、ACO については MANN_a81 の結果が報告されていないため省略した。

表 2 および 3 の Average (最下段) の結果から、本 MA は、他の手法に比べ、Best, Avg ともに最も良い結果を算出できることが分かる。特に、DIMACS ベンチマーク問題例の中で困難なグラフとされる MANN_a81 や keller6 に対する MA の結果は、他の進化計算手法の結果を大きく上回っていることが確認できる。これらの結果は、地形解析を通して設計した本 MA の有効性を示す結果と考えられる。さらに表 4 は、表 2 および 3 の Best および Avg の結果について、本 MA と他の 4 手法それぞれを比較して、より良い結果 (同値の結果を含む) を得た問題例の数を示した。この問題例数の結果からも、本 MA は、他の手法に比べ、平均的に良好な結果を算出した問題例数が明らかに多いことを確認できる。

探索時間については、実験環境や得られる解の質に差異があるため、厳密な比較を行うことができない。ここでは、探索時間の比較について、次の理由により R-EVO を比較対象とする。その理由は、我々の使用計算機と同等以上の性能を持つ計算機が使用されていることと、HSSGA, EA/G および ACO に比べて平均的に良好な結果が報告されており、高性能な進化的アルゴリズムの 1 つと考えられるためである。比較方法として、表 3 における本 MA と R-EVO の Avg の結果が同値である 25 グラフを対象にして、Time がより良い結果であった問題例の数 (表 5) によって比較する。表 5 の結果より、本 MA は R-EVO よりも効率的に同値の解を算出している問題例が多いことを確認できる。よって提案法は、

*1 R-EVO は、現在のところ MCP に対する最良のメタ戦略アルゴリズムの 1 つである Reactive Local Search³⁾ のアイデアを進化計算法に取り入れた解法である。

表 2 DIMACS ベンチマークグラフに対する MA, HSSGA および EA/G の比較結果
 Table 2 Comparison results of MA, HSSGA and EA/G on DIMACS benchmark graphs.

DIMACS benchmarks		MA			HSSGA ²⁴⁾			EA/G ³⁰⁾		
Instance	BR	Best	Avg	Time(s)	Best	Avg	Time(s)	Best	Avg	Time(s)
C125.9	34*	34	34.0	< ϵ	34	34.0	0.017	34	34.0	1.3
C250.9	44*	44	44.0	0.012	44	43.8	0.097	44	44.0	2.5
C500.9	57	57	57.0	1.464	56	54.2	1.190	56	55.2	4.8
C1000.9	68	68	67.8	54.568	66	64.1	4.073	67	64.4	18.0
C2000.9	78	78	77.2	235.741	74	71.0	33.587	72	70.9	38.4
DSJC500.5	13*	13	13.0	0.037	13	13.0	0.202	13	13.0	4.0
DSJC1000.5	15*	15	15.0	11.780	15	14.7	2.106	15	14.5	10.3
C2000.5	16	16	16.0	21.060	16	15.4	7.855	16	14.9	24.3
C4000.5	18	18	17.1	232.880	17	16.8	45.223	17	16.1	51.9
MANN_a27	126*	126	126.0	0.018	126	125.5	0.904	126	126.0	10.3
MANN_a45	345*	344	344.0	4.924	343	342.6	18.626	345	343.7	68.2
MANN_a81	1100	1100	1100.0	574.522	1095	1094.2	1140.894	1098	1097.2	705.1
brock200.2	12*	12	11.8	1.367	12	12.0	0.082	12	12.0	1.5
brock200.4	17*	17	16.3	0.872	17	16.7	0.325	17	16.5	1.7
brock400.2	29*	25	25.0	0.099	29	25.1	0.670	25	24.7	3.1
brock400.4	33*	33	26.6	1.074	33	27.0	0.787	33	25.1	3.3
brock800.2	24	21	21.0	4.092	21	20.7	3.060	21	20.1	7.6
brock800.4	26	21	21.0	15.083	21	20.1	0.867	21	19.9	7.6
gen200_p0.9_44	44*	44	44.0	0.024	44	43.1	0.305	44	44.0	1.8
gen200_p0.9_55	55*	55	55.0	0.003	55	55.0	0.082	55	55.0	3.3
gen400_p0.9_55	55	55	54.2	17.488	53	51.4	0.522	55	51.8	3.6
gen400_p0.9_65	65	65	65.0	0.030	65	63.8	0.488	65	65.0	3.6
gen400_p0.9_75	75	75	75.0	0.029	75	75.0	0.550	75	75.0	3.7
hamming8-4	16*	16	16.0	< ϵ	16	16.0	0.002	16	16.0	1.7
hamming10-4	40	40	40.0	0.103	40	39.0	2.914	40	39.8	14.2
keller4	11*	11	11.0	< ϵ	11	11.0	0.002	11	11.0	1.3
keller5	27	27	27.0	0.011	27	26.9	1.153	27	26.9	9.1
keller6	59	59	59.0	3.417	57	54.2	89.820	56	53.4	53.6
p_hat300-1	8*	8	8.0	0.001	8	8.0	0.005	8	8.0	2.0
p_hat300-2	25*	25	25.0	0.001	25	25.0	0.005	25	25.0	2.0
p_hat300-3	36*	36	36.0	0.009	36	35.9	0.051	36	36.0	2.3
p_hat700-1	11*	11	11.0	0.123	11	11.0	0.291	11	11.0	5.6
p_hat700-2	44*	44	44.0	0.008	44	44.0	0.054	44	44.0	7.6
p_hat700-3	62	62	62.0	0.017	62	61.7	0.573	62	62.0	11.1
p_hat1500-1	12*	12	12.0	34.519	12	11.5	4.173	12	11.1	16.8
p_hat1500-2	65	65	65.0	0.083	65	64.9	0.579	65	65.0	24.6
p_hat1500-3	94	94	94.0	0.301	94	93.1	0.830	94	93.7	29.2
Average	77.27	76.91	76.64		76.54	75.71		76.56	75.83	

表 3 DIMACS ベンチマークグラフに対する MA, R-EVO および ACO の比較結果
 Table 3 Comparison results of MA, R-EVO and ACO on DIMACS benchmark graphs.

DIMACS benchmarks		MA			R-EVO ²⁾			ACO ²⁵⁾		
Instance	BR	Best	Avg	Time(s)	Best	Avg	Time(s)	Best	Avg	Time(s)
C125.9	34*	34	34.0	< ϵ	34	34.0	1.215	34	34.0	0.0
C250.9	44*	44	44.0	0.012	44	44.0	3.237	44	44.0	1.0
C500.9	57	57	57.0	1.464	57	57.0	4.227	57	55.9	8.6
C1000.9	68	68	67.8	54.568	68	68.0	10.165	68	66.2	49.8
C2000.9	78	78	77.2	235.741	77	76.5	29.196	78	74.3	238.7
DSJC500.5	13*	13	13.0	0.037	13	13.0	1.573	13	13.0	1.4
DSJC1000.5	15*	15	15.0	11.780	15	15.0	3.732	15	14.3	7.8
C2000.5	16	16	16.0	21.060	16	16.0	6.833	16	15.3	40.6
C4000.5	18	18	17.1	232.880	18	17.1	15.729	18	16.8	257.6
MANN_a27	126*	126	126.0	0.018	126	125.8	7.991	126	126.0	44.8
MANN_a45	345*	344	344.0	4.924	343	342.5	46.859	344	342.9	749.4
MANN_a81	1100	1100	1100.0	574.522	1097	1096.7	438.570	—	—	—
brock200_2	12*	12	11.8	1.367	12	11.4	0.654	12	12.0	0.1
brock200_4	17*	17	16.3	0.872	17	16.1	1.249	17	16.8	1.7
brock400_2	29*	25	25.0	0.099	25	25.0	1.794	25	24.8	3.8
brock400_4	33*	33	26.6	1.074	25	25.0	2.965	33	27.1	5.7
brock800_2	24	21	21.0	4.092	21	21.0	3.725	24	20.1	11.6
brock800_4	26	21	21.0	15.083	21	21.0	2.977	26	20.0	11.1
gen200_p0.9_44	44*	44	44.0	0.024	44	44.0	1.789	44	44.0	0.5
gen200_p0.9_55	55*	55	55.0	0.003	55	55.0	2.026	55	55.0	0.3
gen400_p0.9_55	55	55	54.2	17.488	55	55.0	3.378	53	52.2	6.7
gen400_p0.9_65	65	65	65.0	0.030	65	65.0	3.744	65	65.0	2.3
gen400_p0.9_75	75	75	75.0	0.029	75	75.0	5.114	75	75.0	2.1
hamming8-4	16*	16	16.0	< ϵ	16	16.0	1.814	16	16.0	0.1
hamming10-4	40	40	40.0	0.103	40	40.0	8.110	40	39.3	29.3
keller4	11*	11	11.0	< ϵ	11	11.0	0.837	11	11.0	0.0
keller5	27	27	27.0	0.011	27	26.8	3.680	27	27.0	12.3
keller6	59	59	59.0	3.417	55	53.7	34.573	57	55.1	549.2
p_hat300-1	8*	8	8.0	0.001	8	8.0	1.149	8	8.0	0.0
p_hat300-2	25*	25	25.0	0.001	25	25.0	2.774	25	25.0	0.2
p_hat300-3	36*	36	36.0	0.009	36	36.0	2.194	36	36.0	0.7
p_hat700-1	11*	11	11.0	0.123	11	11.0	1.950	11	11.0	2.4
p_hat700-2	44*	44	44.0	0.008	44	44.0	6.166	44	44.0	4.5
p_hat700-3	62	62	62.0	0.017	62	62.0	12.579	62	62.0	7.8
p_hat1500-1	12*	12	12.0	34.519	12	11.8	5.181	12	11.1	9.7
p_hat1500-2	65	65	65.0	0.083	65	65.0	18.577	65	65.0	35.2
p_hat1500-3	94	94	94.0	0.301	94	94.0	24.735	94	94.0	54.9
Average	77.27	76.91	76.64		76.45	76.30				

表 4 本 MA と各手法の最良・平均結果の比較

Table 4 Comparison of the best and average results.

MA		HSSGA	
Best	Avg	Best	Avg
36/37	33/37	29/37	14/37
MA		EA/G	
Best	Avg	Best	Avg
36/37	35/37	31/37	19/37
MA		R-EVO	
Best	Avg	Best	Avg
37/37	35/37	32/37	27/37
MA		ACO	
Best	Avg	Best	Avg
34/36	33/36	34/36	22/36

表 5 本 MA と R-EVO の探索時間の比較

Table 5 Comparison of the running times.

MA	R-EVO
20/25	5/25

探索時間についても比較的効率の良い解法であると考えられる。

以上より、提案法 MA は、最先端の進化計算手法に比べて、平均的に高品質の解、もしくは少なくとも同質の解を効率良く算出可能であることを示した。

6. む す び

本論文では、代表的な組合せ最適化問題である最大クリーク問題 (MCP) に対して地形解析を実施し、その探索空間が大谷構造とは異なる構造を示唆する、きわめて悲観的な解析結果を示した。またその結果をもとに、交叉操作を利用せず、選択操作、突然変異操作および局所探索法によって構成される Memetic アルゴリズム (MA) を提案した。DIMACS ベンチマークグラフの代表的な例題を対象として、MCP に対する最先端な進化計算手法の結果との比較を行った。その結果、本 MA はそれらの進化的アプローチに比べ、平均的に良質な解を効率良く算出可能であることを示し、提案法の有効性を確認した。

謝辞 編集委員および査読者の先生方のご指摘、コメントにより、論文の内容が改善されました。感謝いたします。本研究の一部は、平成 18 年度岡山理科大学大学院システム科学専攻重点配分予算および平成 19 年度日本学術振興会科学研究費補助金 (基礎基盤 (C))

No.19500200) を受けて行われた。

参 考 文 献

- 1) Arora, S., Lund, C., Motwani, R., Sudan, M. and Szegedy, M.: Proof Verification and the Hardness of Approximation Problems, *Proc. 33rd Annual Symposium on Foundations of Computer Science, Pittsburgh, PA*, pp.14–23 (1992).
- 2) Battiti, R. and Brunato, M.: R-EVO: A Reactive Evolutionary Algorithm for the Maximum Clique Problem, Technical Report DIT-07-034, University of Trento, Italy (2007).
- 3) Battiti, R. and Protasi, M.: Reactive Local Search for the Maximum Clique Problem, *Algorithmica*, Vol.29, No.4, pp.610–637 (2001).
- 4) Boese, K.D., Kahng, A.B. and Muddu, S.: A new adaptive multi-start technique for combinatorial global optimizations, *Operations Research Letters*, Vol.16, No.2, pp.101–113 (1994).
- 5) Bomze, I., Budinich, M., Pardalos, P. and Pelillo, M.: The Maximum Clique Problem, *Handbook of Combinatorial Optimization, Suppl. Vol.A*, Du, D.-Z. and Pardalos, P. (Eds.), pp.1–74, Kluwer (1999).
- 6) Dorigo, M. and Stützle, T.: *Ant Colony Optimization*, MIT Press (2004).
- 7) Feige, U., Goldwasser, S., Lovász, L., Safra, S. and Szegedy, M.: Approximating Clique is Almost NP-complete, *Proc. 32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico*, pp.2–12 (1991).
- 8) Garey, M. and Johnson, D.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York (1979).
- 9) Glover, F. and Kochenberger, G.: *Handbook of Metaheuristics*, Kluwer (2003).
- 10) Goldberg, D.: *Genetic Algorithm in Search, Optimization and Machine Learning*, Addison-Wesley (1989).
- 11) Hart, W., Krasnogor, N. and Smith, J.: *Recent Advances in Memetic Algorithms*, Springer (2004).
- 12) Håstad, J.: Clique is Hard to Approximate within $n^{1-\epsilon}$, *Acta Mathematica*, Vol.182, pp.105–142 (1999).
- 13) 池田 心, 小林重信: 生得分離モデルを用いた GA と JSP への適用, *人工知能学会論文誌*, Vol.17, No.5, pp.530–538 (2002).
- 14) 石淵久生: Memetic アルゴリズム, メタヒューリスティクスと応用, pp.15–26, 電気学会 (2007).
- 15) Johnson, D. and Trick, M.: Cliques, Coloring, and Satisfiability, *2nd DIMACS Implementation Challenge, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, American Mathematical Society (1996).
- 16) 片山謙吾, 石淵久生: Memetic アルゴリズム, 計測と制御, Vol.47, No.6, pp.487–492

(2008).

- 17) 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).
- 18) 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).
- 19) Marchiori, E.: Genetic, iterated and multistart local search for the maximum clique problem, *Applications of Evolutionary Computing*, LNCS 2279, pp.112–121, Springer-Verlag (2002).
- 20) Merz, P. and Freisleben, B.: Fitness Landscapes, Memetic Algorithms and Greedy Operators for Graph Bi-Partitioning, *Evolutionary Computation*, Vol.8, No.1, pp.61–91 (2000).
- 21) Merz, P. and Katayama, K.: Memetic Algorithms for the Unconstrained Binary Quadratic Programming Problem, *BioSystems*, Vol.78, No.1–3, pp.99–118 (2004).
- 22) Moscato, P. and Cotta, C.: A Gentle Introduction to Memetic Algorithms, *Handbook of Metaheuristics*, pp.105–144, Kluwer (2003).
- 23) 島本 隆, 来山征士, 橋本 学, 坂本明雄: マルチデコード GA を用いた最大独立集合問題の解法, 電子情報通信学会論文誌 (A), Vol.J83-A, No.12, pp.1505–1511 (2000).
- 24) Singh, A. and Gupta, A.K.: A hybrid heuristic for the maximum clique problem, *Journal of Heuristics*, Vol.12, No.1-2, pp.5–22 (2006).
- 25) Solnon, C. and Fenet, S.: A study of ACO capabilities for solving the maximum clique problem, *Journal of Heuristics*, Vol.12, No.3, pp.155–180 (2006).
- 26) 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).
- 27) Yagiura, M. and Ibaraki, T.: Local Search, *Handbook of Applied Optimization*, Pardalos, P.M. and Resende, M.G.C. (Eds.), pp.104–123, Oxford University Press (2002).
- 28) 山田武士, Reeves, C.: フローショップスケジューリング問題の地形解析と遺伝的局所探索による解法, 情報処理学会論文誌, Vol.39, No.7, pp.2112–2123 (1998).
- 29) 山下浩司, 片山謙吾, 南原英生, 成久洋之: ノード配置問題に対する Memetic アルゴリズム, 進化計算シンポジウム 2007 講演論文集, pp.99–102, 進化計算研究会 (2007).
- 30) Zhang, Q., Sun, J. and Tsang, E.: An Evolutionary Algorithm With Guided Mutation for the Maximum Clique Problem, *IEEE trans. Evolutionary Computation*, Vol.9, No.2, pp.192–200 (2005).

(平成 20 年 2 月 29 日受付)

(平成 20 年 9 月 10 日採録)



片山 謙吾 (正会員)

平成 7 年岡山理科大学大学院修士課程修了。平成 10 年同大学院博士課程修了。博士 (工学)。同年同大学工学部情報工学科助手。平成 13 年同講師。平成 19 年同准教授。平成 17 年カリフォルニア大学サンタバーバラ校客員研究員。メタ戦略, 組合せ最適化, 機械学習, マルチエージェントシステムに関する研究に従事。IEEE, 電子情報通信学会, 人工知能学会

各会員。



貞松 政史

平成 18 年岡山理科大学工学部情報工学科卒業。平成 20 年同大学大学院修士課程修了。現在, 株式会社リゾームに勤務。組合せ最適化問題に対するメタ戦略およびデータマイニングに興味を持つ。



南原 英生 (正会員)

昭和 47 年立命館大学大学院修士課程修了。広島電機大学工学部助手, 同講師, 同助教授, 同教授を経て, 現在, 岡山理科大学工学部情報工学科教授。工学博士。信号解析, 最適化に関する研究に従事。電子情報通信学会, 応用統計学会, 計測自動制御学会, 日本音響学会等各会員。



成久 洋之 (正会員)

昭和 45 年京都大学大学院博士課程修了。工学博士。現在, 岡山理科大学工学部情報工学科特任教授。昭和 58 ~ 平成 6 年岡山理科大学情報処理センター所長を併任。オペレーションズリサーチおよびシステムの最適化に関する研究に従事。日本 OR 学会フェロー。電子情報通信学会会員。