

## 最大クリーク問題に対する Memetic Algorithm の選択法

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

(2007年10月1日受付、2007年11月2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の中で, 頂点数最大の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>.

MCP に対して, 分枝限定法等の厳密解法がいくつか提案されている<sup>6)7)</sup> が, それらの解法は, 比較的小規模もしくは疎なグラフに対してのみ有効である. そのため, 大規模なグラフに対して現実的な計算時間で良質な近似解を求めるための近似解法が数多く提案されてきた.

局所探索法 (Local Search, LS) にもとづく近似解法として, Battiti らによる Reactive Local Search (RLS)<sup>8)</sup>, Hansen らによる Variable Neighborhood Search (VNS)<sup>9)</sup>, 片山らによる  $k$ -opt Local Search (KLS)<sup>10)</sup>, Pullan らによる Dynamic Local Search (DLS)<sup>11)</sup> などが提案されている. 特に, RLS, DLS は MCP に対する最も強力な近似解法として非常に有名である. また, 進化的アルゴリズム (Evolutionary Algorithm, EA), 遺伝的アルゴリズム (Genetic Algorithm, GA) などの進化計算手法にもとづく近似解法として, Zhang らによる Evolutionary Algorithm with Guided Mutation (EA/G)<sup>12)</sup>, Singh らによる Heuristic based Steady-State Genetic Algorithm (HSSGA)<sup>13)</sup> などが提案されている.

最近我々は, MCP に対する地形解析によって, MCP の問題例の多くは探索空間が大域最適解に向かって構造化されていないことを確認している<sup>14)</sup>. したがって, 主要な進化オペレータとして, 一般的な交叉を用いる手法では, MCP に対して理想的な挙動を期待できないと考えられる. 本論文では, 進化計算手法の一つである Memetic Algorithm (MA) を MCP に対して適用することを試みる. 本 MA は, 文献<sup>14)</sup> の観測にもとづき, 交叉を用いず, 選択法, 突然変異および局所探索法によって構成される. 我々は既に, 局所探索法については KLS<sup>10)</sup>, 突然変異については LEC-Kick<sup>15)</sup> がそれぞれ有効な手法であることを示しているが, 選択法についての検討は十分になされていない. そこで, 本 MA 内部に導入可能な複数の次世代個体選択法を示し, その相違による探索性能および探索の多様性を検討する.

### 2. MCP に対する Memetic Algorithm

本節では, 本論文で提案する MCP に対する Memetic Algorithm (MA) について記述する. まず, MA の概要を示した後, MA の主要な構成要素について具体的に説明する.

```

procedure MA(psize)
begin
1   for i = 1 to psize do begin
2     Pp[i] := Initialize();
3     Pp[i] := KLS(Pp[i]);
4   end;
5   repeat
6     for i = 1 to psize do begin
7       Pc[i] := LEC-Kick(Pp[i]);
8       Pc[i] := KLS(Pc[i]);
9     end;
10    Pp := Select(Pp, Pc);
11  until terminate=true;
12  return best individual ∈ P;
end;

```

図1 MCP に対する Memetic Algorithm の流れ

## 2.1 MA の基本アルゴリズム

MA の流れを図1に示す。psize を親個体集団および子個体集団の個体数、Pp を親個体集団、Pc を子個体集団とする。まず、初期プロセスとして、psize 個の Pp それぞれにグラフ  $G$  の頂点から選択した1頂点を初期クリークとして与え、Local Search を適用することによって初期個体集団を得る (Line 1~4)。その後、Pp に対して Kick, Local Search を適用することで Pc を生成し (Line 6~9)、Pp と Pc を併せた個体集団から次世代の Pp を psize 個選択 (Selection) する (Line 10)。このプロセスを任意の世代数回繰り返すことによって探索を行う。以下で、本 MA の主要な構成要素である、Local Search, Kick, Selection について示す。

## 2.2 Local Search

MA 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)<sup>10)</sup> を用いる。以下で KLS の概要について示す。

### 2.2.1 KLS の基本アルゴリズム

KLS は、可変深度探索 (Variable Depth Search, VDS)<sup>16)17)</sup> 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

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

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。

```

procedure KLS(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

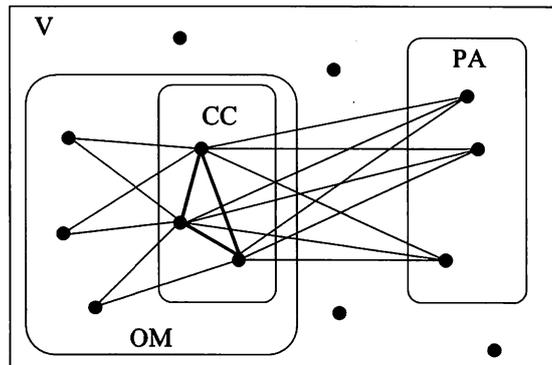


図3 CC, PA および OM の集合の一例

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。

```

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

### 2.2.2 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。

### 2.2.3 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。

## 2.3 Kick

本 MA で用いる突然変異 (Kick) 法として、Lowest-Edges-Connectivity-based Kick (LEC-Kick)<sup>15)</sup> を用いる。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、今回の 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。

## 2.4 Selection

一般的に、組合せ最適化問題に対する高性能なメタ戦略アルゴリズムは、探索の多様性および局所性をバランスよく備えたものである。本 MA で用いる次世代個体集団の選択法についても、評価値 (クリークサイズ) や個体間の距離などの指標を用いて、探索の多様性および局所性を考慮したものを設計する必要があると考えられる。本 MA で用いる選択法 (Selection) として、以下の 4 種類を考える。

- **Selection1**: 親個体集団  $P_p$  の各解に対して Kick および KLS を適用後の解群を次世代の  $P_p$  とする。すなわち、1 世代前の  $P_p$  の解群は次世代に陽には生存しない。なお、評価値 (クリークサイズ) および個体間の距離を全く考慮していない。よって選択圧力は無い、もしくは非常に低い。

- **Selection2**: 子個体集団  $P_c$  と親個体集団  $P_p$  の中から、クリークサイズの大きい個体集団を次世代の  $P_p$  とする。その際、解構造が同一の解を集団に含まないようにする。なお、個体間の距離を考慮していないため、探索の進行に伴って、距離の近い、評価値の高い個体で集団が構成されてしまう可能性が高い。よって選択圧力は非常に高い。
- **Selection3**: 子個体集団  $P_c$  と親個体集団  $P_p$  の中から、クリークサイズの大きい個体集団を次世代の  $P_p$  とする。その際、クリークサイズが同値の各解とのハミング距離が  $d$  以下の解を集団に含まないようにする。 $d$  の決定には、各世代毎の各個体に対する LEC-Kick によって  $C$  から削除された頂点数の平均値を用いる。個体間の距離をある程度考慮しているものの、 $d$  の値に依存するところがあり、評価値による選択圧力は高いと考えられる。
- **Selection4**: 子個体集団  $P_c$  と親個体集団  $P_p$  の中から、 $P_c$  および  $P_p$  の各個体との平均ハミング距離が大きい個体集団を次世代の  $P_p$  とする。評価値を全く考慮していないため、評価値による選択圧力は無いが、個体間の距離による選択圧力が非常に高い。

### 3. MA の性能評価実験

本節では、2.4 節で示した 4 種類の選択法の相違による MA の探索性能および探索の多様性について検討を行う。まず、実験 I として、DIMACS ベンチマークグラフを対象に各 MA の探索性能を比較検討する。次いで実験 II として、DIMACS ベンチマークグラフに各 MA を適用した際の各個体間の平均ハミング距離を調べることで、各 MA による探索の多様性を調査する。

#### 3.1 実験 I の詳細

選択法の相違による MA の探索性能を評価するために、4 種類の選択法 (Selection1~4) をそれぞれ有する 4 タイプの MA 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)<sup>18)</sup> から大規模もしくは厳密解の算出が困難な 37 グラフとする。各 MA の各個体に与える  $psize$  個の初期クリークは、グラフ  $G$  の頂点を度数に基づきあらかじめソートし、度数の高い頂点から非増加順に与える。各 MA の個体数  $psize$  を 20、計算打ち切り世代数を  $n \times 5$  回とする。なお、既知の最良解 (BR) 算出時にも計算を打ち切る。試行回数は各問題例に対して 10 回とする。全ての 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。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。

#### 3.2 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) 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。MA の欄には、Selection1~4 を有する MA の結果を示しており、各 MA の 10 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。

表 1 の結果より、Selection1 は、他の選択法と比べて全体的に良好な探索性能を示している。また、Selection2 は、他の選択法と比べて全体的に劣る結果を示している。Selection3 は、C2000.9 や keller6 のグラフで最も良好な結果を得ているが、brock や gen400\_p0.9.55 のグラフにおいて悪質な解を算出しており、ロバスト性に欠ける探索であることを示している。Selection4 についても、brock や gen400\_p0.9.55 のグラフにおいて比較的良好な結果を得ているが、C2000.9 や keller6 において悪質な解を算出しており、MANN\_a81 においては計算時間が膨大になっているため、やはりロバスト性に欠けている。以上より、選択法の相違によって、MA の探索性能に差が生じることを明らかにしたが、各選択法がどのように MA の探索に影響を与えるのかは不明である。そこで、選択法の相違が MA の探索に与える影響についてより深く調査するために、次の実験 II による結果とあわせてさらに考察する。

#### 3.3 実験 II の詳細

地形解析の結果<sup>14)</sup> より、MCP は大域最適解と局所最適解、および局所最適解同士に相関が見られないため、個体同士の距離を考慮し、個体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、上述 4 タイプの MA における個体集団内の多様性を調べるため、各 MA を DIMACS ベンチマークグラフに適用し

表 1 DIMACS グラフに対する 4 タイプの MA の実験結果

DIMACS benchmarks		MA											
Instance	BR	Selection1			Selection2			Selection3			Selection4		
		Best	Avg	Time(s)	Best	Avg	Time(s)	Best	Avg	Time(s)	Best	Avg	Time(s)
C125.9	34*	34	34.00	0.001	34	34.00	0.001	34	34.00	< ε	34	34.00	< ε
C250.9	44*	44	44.00	0.038	44	44.00	0.039	44	44.00	0.012	44	44.00	0.010
C500.9	57	57	57.00	1.703	57	56.80	1.985	57	57.00	1.464	57	57.00	9.296
C1000.9	68	68	68.00	53.136	68	67.20	47.032	68	67.80	54.568	68	67.20	27.221
C2000.9	78	77	75.90	262.946	77	76.40	228.040	78	77.20	235.741	77	75.50	290.772
DSJC000.5	13*	13	13.00	0.046	13	13.00	0.079	13	13.00	0.037	13	13.00	0.048
DSJC1000.5	15*	15	15.00	1.417	15	15.00	17.838	15	15.00	11.780	15	15.00	4.935
C2000.5	16	16	16.00	11.348	16	16.00	15.436	16	16.00	21.060	16	16.00	20.364
C4000.5	18	18	18.00	1549.171	18	17.10	226.911	18	17.10	232.880	18	17.10	424.882
MANN_a27	126*	126	126.00	0.019	126	126.00	0.019	126	126.00	0.018	126	126.00	0.013
MANN_a45	345*	345	344.60	285.193	344	344.00	4.637	344	344.00	4.924	345	344.00	187.076
MANN_a81	1100	1100	1100.00	825.889	1100	1099.40	503.088	1100	1100.00	574.522	1100	1099.80	47428.740
brock200_2	12*	12	12.00	0.069	12	11.60	1.135	12	11.80	1.367	12	12.00	0.072
brock200_4	17*	17	17.00	0.072	17	16.30	0.530	17	16.30	0.872	17	17.00	0.332
brock400_2	29*	29	27.80	5.042	25	25.00	0.139	25	25.00	0.099	25	25.00	0.230
brock400_4	33*	33	33.00	1.050	33	26.60	0.395	33	26.60	1.074	33	32.20	2.826
brock800_2	24	21	21.00	1.739	21	21.00	2.265	21	21.00	4.092	21	21.00	3.867
brock800_4	26	21	21.00	3.231	21	21.00	3.980	21	21.00	15.083	21	21.00	5.647
gen200_p0.9_44	44*	44	44.00	0.015	44	44.00	0.017	44	44.00	0.024	44	44.00	0.019
gen200_p0.9_55	55*	55	55.00	0.002	55	55.00	0.001	55	55.00	0.003	55	55.00	0.002
gen400_p0.9_55	55	55	55.00	1.954	55	53.60	6.839	55	54.20	17.488	55	55.00	1.519
gen400_p0.9_65	65	65	65.00	0.033	65	65.00	0.033	65	65.00	0.030	65	65.00	0.022
gen400_p0.9_75	75	75	75.00	0.014	75	75.00	0.014	75	75.00	0.029	75	75.00	0.022
hamming8-4	16*	16	16.00	< ε	16	16.00	< ε	16	16.00	< ε	16	16.00	< ε
hamming10-4	40	40	40.00	0.047	40	40.00	0.048	40	40.00	0.103	40	40.00	0.075
keller4	11	11	11.00	< ε	11	11.00	< ε	11	11.00	< ε	11	11.00	< ε
keller5	27	27	27.00	0.074	27	27.00	0.096	27	27.00	0.011	27	27.00	0.008
keller6	59	59	59.00	7.101	59	58.50	304.124	59	59.00	3.417	59	56.90	308.530
p_hat300-1	8*	8	8.00	0.003	8	8.00	0.003	8	8.00	0.001	8	8.00	0.001
p_hat300-2	25*	25	25.00	0.001	25	25.00	< ε	25	25.00	0.001	25	25.00	0.001
p_hat300-3	36*	36	36.00	0.005	36	36.00	0.005	36	36.00	0.009	36	36.00	0.006
p_hat700-1	11*	11	11.00	0.026	11	11.00	0.032	11	11.00	0.123	11	11.00	0.079
p_hat700-2	44*	44	44.00	0.006	44	44.00	0.006	44	44.00	0.008	44	44.00	0.007
p_hat700-3	62	62	62.00	0.016	62	62.00	0.016	62	62.00	0.017	62	62.00	0.013
p_hat1500-1	12*	12	12.00	17.253	12	12.00	13.237	12	12.00	34.519	12	12.00	16.041
p_hat1500-2	65	65	65.00	0.052	65	65.00	0.052	65	65.00	0.083	65	65.00	0.058
p_hat1500-3	94	94	94.00	0.225	94	94.00	0.250	94	94.00	0.301	94	94.00	0.205

た際の各世代における各個体間の平均ハミング距離を調査する。対象グラフは、実験 I において、各 MA の結果に差が見られたグラフから C1000.9, C2000.9, MANN\_a45, MANN\_a81, brock400\_2, brock400\_4, gen400\_p0.9\_55, keller6 の 8 グラフを対象とし、試行回数は各問題例に対して 1 回とする。その他実験設定および実験環境は実験 I と同様である。

### 3.4 実験 II の結果

結果を図 5 に示す。(a)~(h) の各グラフは横軸を世代数、縦軸を各個体間の平均ハミング距離としている。また、世代数については、グラフを見易くするため 100 世代までを示している。なお、keller6 等の、途中で折れ線が途切れてしまっているグラフに関しては、折れ線の途切れている世代数で既知の最良解を算出したことを示している。図 5 の結果より、各世代における各個体間の平均ハミング距離について、全体的に Selection1 および Selection4 が高く、Selection2 および Selection3 が低いことがわかる。ほぼ各選択法の選択圧力等の特徴に従った結果が得られているといえるが、個体間の距離のみを考慮した Selection4 ですら、探索の早期に平均ハミング距離が急激に減少している問題例がいくつかある。このことから、MA 内部で使用した Local Search である KLS の決定論的な探索が MA の探索に強く影響していることが推測される。図 5 の結果と実験 I による表 1 の結果より、brock400\_2, brock400\_4, gen400\_p0.9\_55 のグラフについては、平均ハミング距離の大きい Selection1 および Selection4 が良質な解を算出しており、C2000.9 のグラフについては平均ハミング距離の小さい Selection2 および Selection3 が良質な解を算出している。また、全体的に良好な探索性能を示した Selection1 の平均ハミング距離は、4 タイプの MA の中でも中間的な距離を保っている。以上の結果を踏まえると、本 MA に対して有効な選択法を設計するためには、各問題例に対する、探索の局所性と多様性のバランスについて十分に考慮する必要があるといえる。

## 4. むすび

本論文では、MCP に対する Memetic Algorithm(MA) を示し、その内部に導入可能な複数の選択法の相違による探索性能および探索の多様性を調査した。異なる 4 種類の選択法を有する 4 タイプの MA の探索性能の比較検討において、Selection1 を有する MA が全体的に比較的良好な結果を示した。また、各 MA による探索の多様性を調査するために、各世代における各個体間の平均ハミング距離を調査した。その結果、各問題例に対する探索の局所性および多様性のバランスが本 MA の探索性能に強く影響することを示した。

以下では、今後の課題・検討事項について記述する。

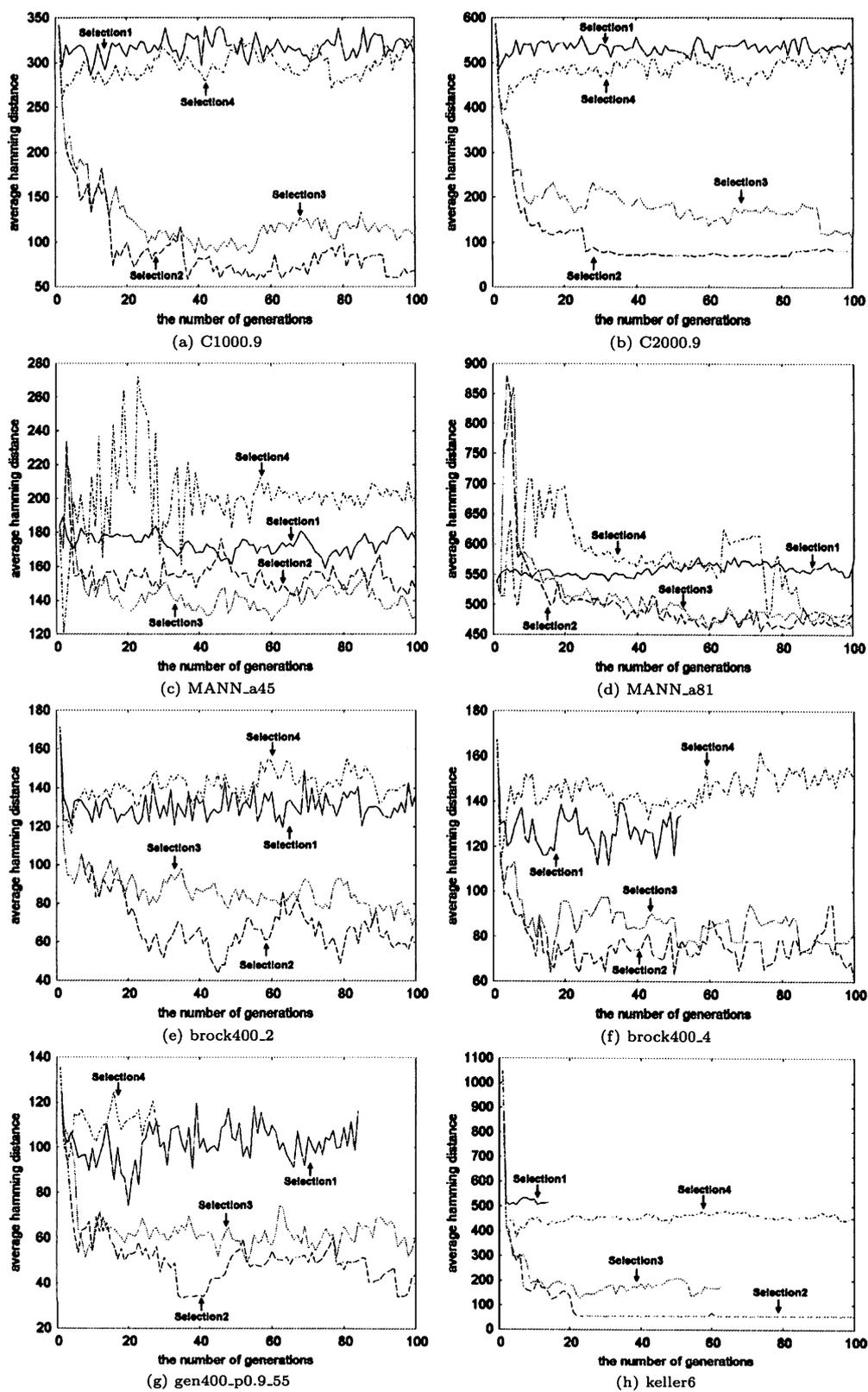


図 5 各世代における各個体間の平均ハミング距離

(1) 実験 I・II の結果より、探索の局所性と多様性のバランスが重要であると考えられるため、それを十分に考慮した選択法の設計について検討の余地がある。

(2) 実験 II の結果より、KLS の決定論的な探索が MA の探索に強く影響していることが考えられるため、MA の探索途中に得られる情報を KLS 内部で利用することによって、決定論的な探索を緩和しようとするアイデアの導入も興味深い。

#### 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) P.R.J. Östergård, "A fast algorithm for the maximum clique problem," Discrete Applied Mathematics, vol.120, no.1-3, pp.197-207, 2002.
- 7) E. Tomita and T. Kameda, "An Efficient Branch-and-bound Algorithm for Finding a Maximum Clique with Computational Experiments," Journal of Global Optimization, vol.37, no.1, pp.95-111, 2007.
- 8) R. Battiti and M. Protasi, "Reactive local search for the maximum clique problem," Algorithmica, vol.29, no.4, pp.610-637, 2001.
- 9) P. Hansen, N. Mladenović and D. Urošević, "Variable neighborhood search for the maximum clique," Discrete Applied Mathematics, vol. 145, no. 1, pp. 117-125, 2004.
- 10) 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.
- 11) 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.
- 12) Q. Zhang, J. Sun, and E. Tsang, "An Evolutionary Algorithm With Guided Mutation for the Maximum Clique Problem," IEEE Transactions on Evolutionary Computation, vol. 9, no. 2, April 2005.
- 13) A. Singh and A. K. Gupta "A hybrid heuristic for the maximum clique problem," Journal of Heuristics, vol. 12, pp. 5-22, January 2006.
- 14) 上野伸郎, 濱本宏宏, 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.
- 15) K. Katayama, M. Sadamatsu, and H. Narihisa, "Iterated  $k$ -opt Local Search for the Maximum Clique Problem," 7th European Conference on Evolutionary Computation in Combinatorial Optimization, pp. 84-95, April 2007.
- 16) B. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," Bell System Technical Journal, vol.49, pp.291-307, 1970.
- 17) S. Lin and B. Kernighan, "An effective heuristic algorithm for the traveling salesman problem," Operations Research, vol.21, pp.498-516, 1973.
- 18) 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.

## On Selection Methods of Memetic Algorithm 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 1, 2007; accepted November 2, 2007)

In the field of evolutionary computation, it is well known that the effectiveness of evolutionary algorithms can be enhanced by incorporating problem-dependant local search heuristics. These approaches are often called Memetic Algorithm (MA). In many cases, MA have been shown to be capable of finding (near-)optimum solutions for difficult combinatorial optimization problems.

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

**Keywords:** combinatorial optimization; maximum clique problem; memetic algorithm; local search.