大規模な最大多様性問題に対する遺伝的局所探索

片 山 謙 吾[†]　成 久 洋 之[†]

最大多様性問題（maximum diversity problem，MDP）とは，与えられた n 個の要素から m 個の要素を選ぶとき，できるだけ多様性を有するように要素を選定する問題である．本論文では，MDP に対する遺伝的局所探索法（genetic local search，GLS）を示す．本 GLS では，交叉および突然変異の各操作後に生成される解に対して実行可能解変換操作（リペア法）を施し，解の実行可能性を保証する．その後，Lin と Kernighan による可変深度探索のアイデアに基づく k-flip 局所探索法を適用することで，高品質な局所最適解の集団を構成しながら探索を進める．既存研究で対象とされた問題規模よりもはるかに大規模な 2500 変数までの問題例に対して本 GLS をテストし，2-flip 局所探索法をベースにした変形 GLS よりも平均的に良好な解を算出可能であることを示す．

A Genetic Local Search for Large Maximum Diversity Problem

Kengo Katayama[†] and Hiroyuki Narihisa[†]

This paper presents a genetic local search (GLS) — genetic algorithm incorporating local search — for the maximum diversity problem (MDP). To guarantee feasibility of solutions, a repair method is used in the GLS and is applied to a solution after crossover or mutation operator. In a process of local search of the GLS, we perform a sophisticated, k-flip local search based on an idea of variable depth search by Lin and Kernighan. Computational results show that the GLS with k-flip local search is capable of finding better solutions on average than a GLS with a simple 2-flip local search for large-sized problem instances of up to 2500 variables.

1. はじめに

遺伝的局所探索法（genetic local search，GLS）は，遺伝的アルゴリズム（genetic algorithm，GA）と局所探索（local search，LS）をあわせ持つハイブリッド手法である．これまでに，様々な組合せ最適化問題に適用され，優れた結果が多く報告されている[15]．

高性能な GLS に関する研究では，対象とする最適化問題における解の実行可能性を保証する操作の導入や有効に働く局所探索法の採用もしくは開発が不可欠である．特に，比較的新しい最適化問題を対象にする場合には，優れた局所探索法自体が存在しないこともあり，GLS 設計の際の大きな障害になる．さらに，局所探索はメタ戦略[18] に属する他のアルゴリズムでも多くの場合に採用されるため，優れた局所探索法を開発することは進化的計算の枠組みに対してだけでなく，メタ戦略全般および改良に基づく今後の研究に対して重要な指針を与える．

最大多様性問題（maximum diversity problem, MDP）[9),11] は比較的新しい最適化問題であり，我々の知る限りでは，現在までに，GA などの進化的計算法に類する手法は報告されていない．また，素朴な近傍操作による局所探索法は MDP に対して既知であるものの，優れた性能を有する局所探索法はない．さらに，MDP の既存研究において対象とされてきた問題例のサイズは比較的小規模なものがほとんどであり，大規模な問題例に対しての適用例は少ない．

本論文では，MDP に対するこうした研究状況をふまえ，高性能な局所探索法の設計を試みるとともに大規模な MDP に対して有効に働く GLS を示す．我々は，「k-flip 局所探索法」と呼ぶ局所探索法を設計する．これは可変深度探索（variable depth search，VDS）のアイデアに基づいている．VDS は，1970 年前後に Lin と Kernighan により，巡回セールスマン問題[12] およびグラフ分割問題[8] に対して示された巧妙な近傍探索のアイデアである．MDP に対する k-flip 局所探索法を有する GLS を 2500 変数までの大規模な問題例に対して適用し，素朴な近傍を有する 2-flip 局所探索法ベースの GLS よりも平均的に良好な近似解を算出可能であることを示すとともに，より高性能な局所探索を併用することの有効性を明らかにする．

† 岡山理科大学工学部情報工学科
Department of Information and Computer Engineering, Okayama University of Science

2. 最大多様性問題と既存研究

最大多様性問題（MDP）とは，n 個の要素および各要素が共通して有する属性の情報に基づいて，$n \times n$ の対称距離行列 d を算出し，距離に基づく多様性をできるだけ持つように n 個の要素から m 個の要素を選定する問題である．

本論文で対象とする MDP は，$d_{ij} = d_{ji}$ および $d_{ii} = 0$ の一般性を失うことなしに，サイズ m $(n > m > 1)$ が与えられたとき，次の目的関数を最大化する解 x を求める問題である．

$$
\begin{aligned}
\text{maximize} \quad & f(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_i x_j, \\
\text{subject to} \quad & \sum_{i=1}^{n} x_i = m, \\
& x_i \in \{0, 1\}, \ \forall i = 1, \ldots, n.
\end{aligned} \tag{1}
$$

一例として，架空の人選問題を考える．ここで考える人選問題とは，ある町内の将来像を話し合うため，その町内に住む n 人の集合 $S = \{s_i : i \in N\}$（ただし $N = \{1, 2, \ldots, n\}$）から，彼らの経験や学識などに従って，多種多様な意見を出せる多様性に富んだ m 人の部分集合を選出するものである．n 人の人たちは共通する r 個の属性 $s_{ik} (k \in R = \{1, 2, \ldots, r\})$（たとえば，性別，年齢，職業，出身都道府県，最終学歴など）を持つものとする．たとえば性別の場合，女性を 1，男性を 2 として，それらの性質に従う番号付けをすべての場合に対して行う．選ばれる部分集合の多様度を測るために，n 人の各ペアである s_i と s_j の各距離 d_{ij} を算出する必要がある．たとえば，ユークリッド距離（Euclidean distance）を用いて，各ペア（i 番目の人と j 番目の人）の多様度を測る場合，距離 d_{ij} は $[\sum_{k=1}^{r} (s_{ik} - s_{jk})^2]^{1/2}$ で求められる．このようにして対称距離行列 d を算出し，上記の目的関数のもとで，人選問題を解くことができる．

MDP には数多くの応用例が存在する．いくつかの例として，移民入国政策，委員構成問題，カリキュラム作成，マーケット計画やポートフォリオ選択，VLSI 設計，試験スケジューリング，環境保護バランス問題，医療処置問題，分子構造設計，宝石などのパネル配置問題などがあり，多岐にわたっている[5),9),11),17)]．

MDP の最初のモデルは，Kuo ら[11)] によって示された．彼らは，MDP が NP 困難であることを示し，整数計画法に基づくアプローチにより問題解決を試みた．MDP は NP 困難であるので，厳密解法に基づくアルゴリズムでは問題サイズの大規模化にともない膨大な計算時間を必要とする．したがって，いくつかの近似解法が提案されている．Ghosh[4)] は，greedy randomized adaptive search procedure（GRASP）を提案し，40 変数（$n = 40$）までの問題例に対して適用を行った．この GRASP の局所探索過程では 2-flip 近傍が採用されている．Glover ら[5)] は constructive および destructive ヒューリスティックアルゴリズムを示し，30 変数（$n = 30$）までの問題例に対してテストした．Kochenberger ら[9)] はランダムに生成した 100 変数から 1000 変数までの MDP に対して実行不可能領域空間も探索対象とするタブー探索法の適用を行った．このタブー探索法で利用された近傍は 1-flip 近傍である．

3. MDP に対する局所探索法の諸概要

3.1 局所探索の基本処理

局所探索は，様々な最適化問題に適用可能であり，一般的には近似解（局所最適解）の算出を目的とする．一般的な局所探索の基本処理は次のようになる．

(i) 何らかの処理によって与えられる解を初期解 x とする．

(ii) 解 x の近傍 $\mathcal{NB}$ をもとに，近傍解 x' を選ぶ．

(iii) 近傍解 x' の評価値 $f(x')$ が $f(x)$ よりも良好であるならば，x' を x とし，(ii) に戻る．近傍 $\mathcal{NB}(x)$ 内に良好な近傍解が存在しないのであれば，x を出力し，局所探索処理を終了する．

局所探索は，与えられた解を近傍および解の評価関数に基づいて改善する処理である．(i) における「何らかの処理によって与えられる解」とは，たとえば，ランダムにまたは貪欲に生成された解を指す（GLS の場合であれば交叉・突然変異後に生成された解などでもよい）．(ii) における「近傍 $\mathcal{NB}$」とは，解 x に少しの変化を与えることで到達可能な解の集合と定義される．さらに (ii) において「近傍解 x' を選ぶ」が，この選び方には大きくわけて 2 つの方法が知られている．1 つ目は「最良移動」による選び方であり，近傍内のすべての近傍解の中で最良の評価関数値を持つ近傍解を選ぶ方法である．2 つ目は「即時移動」によるもので，近傍内の近傍解をランダムまたはあらかじめ定められた順番に従い選ぶ方法である．(iii) では，選ばれた近傍解 x' の評価関数値に基づいて，もとの解 x よりも優れた解か否かを判定する．もし優れた解であるならば，x' を次の探索のための解 x として，(ii) の処理へ戻る．そうでない場合は，（特に，即時移動が採用されている場合には近傍内に良好な解が存在しないことを確認したうえで）x を出力し局所探索処理を終

了する．最終的に得られる解 x は，近傍 $\mathcal{NB}$ および評価関数 f のもとで「局所最適解」となる．局所最適解とは，すべての近傍解 $x' \in \mathcal{NB}(x)$ に対して（最大化問題の場合）$f(x') \leq f(x)$ の条件を満たす解 x のことをいう．したがって，局所探索法により得られる局所最適解の質はあらかじめ定義される近傍に大きく依存する．

局所探索法の設計では，近傍の定義だけでなく，解の評価値，解の表現，近傍解の評価値の計算などの決定も重要とされている．以下では，MDP に対する局所探索法および遺伝的局所探索法において基本となるそれらの諸概要について記述し，k-flip 局所探索法で採用する近傍解の評価値計算について述べる．

3.2 解の評価値および解の表現

MDP に対する解の評価値は，式 (1) に基づいて評価する．

MDP に対する解 x は長さ $n = |N|$（$N = \{1, 2, \ldots, n\}$）の 0-1 表現とする．この表現では，i 番 $(1 \leq i \leq n)$ の要素に要素値（ビット値）として 0 もしくは 1 を持ち，これを $x_i = 0$ または $x_i = 1$ と表記する．

本研究では，上の 0-1 表現とあわせて次の表現法も利用する．ビット値が 1 となっている $x_i = 1$ $(\forall i \in N)$ の要素番号の集合を S_1 とする．また，ビット値が 0 となっている $x_i = 0$ $(\forall i \in N)$ の要素番号の集合を S_0 とする．この表現においては $S_1 \cup S_0 = N$ や $S_1 \cap S_0 = \emptyset$ の関係が成り立つ．

MDP における解の実行可能性は，$\sum x_i = m$ $(\forall i \in N)$，つまり解 x における 1 の数の和が $m(= |S_1| = |N| - |S_0|)$ に等しいときに保証される．

3.3 近　　傍

近傍は局所探索法の性能を決定する最重要な事項である．MDP では最も素朴な近傍として 1-flip 近傍および 2-flip 近傍がある．以下，それぞれ定義する．

1-flip 近傍とは，解 x が与えられたとき，i 番の要素の値を反転（0 から 1，もしくは 1 から 0）した際に生成可能な解集合と定義される（S_1 の 1 要素を S_1 から除き S_0 に加えるか，またはその逆の操作によって生成可能な解の集合とも定義できる）．よって，実行可能解 x が与えられたとき，1-flip 近傍操作を 1 回施した場合に生成される解は実行不可能になる．

2-flip 近傍とは，解 x が与えられたとき，S_1 の 1 要素と S_0 の 1 要素との交換によって生成可能な解集合と定義される．よって，実行可能解 x が与えられたとき，2-flip 近傍操作によって生成される近傍解は実行可能解になる．

3.4 1-flip 近傍解の評価値高速計算

一般に，局所探索法における近傍解の評価値の計算は，近傍解を生成するたびに必要となるため，局所探索自体の効率に大きく影響する．したがって，近傍解の評価は効率的に行うことが重要になる．

特に，本研究で扱うような MDP は，解の評価関数が 2 次形式であるため，1 つの近傍解を生成・評価するたびに式 (1) を用いて素朴に計算したのでは，毎回 $O(n^2)$ の計算時間量が必要となる．MDP に対して 1-flip 近傍を考える場合，現在解 x と 1-flip 近傍解 x' のハミング距離 d_H はつねに 1（$d_H(x, x') = 1$）であるから，現在解から一度に生成可能な 1-flip 近傍解の候補はたかだか n 個となる．1 つの近傍解生成に必要な手間が定数時間 $O(1)$ とすると，n 個の近傍解の候補から最良の近傍解へ移動する 1 過程（最良移動）を考慮する際には，n 回の近傍解評価が必要となる．素朴に $O(n^2)$ 時間の評価関数を利用するとすれば，n 個の近傍解すべての評価に $O(n^3)$ 時間を要する．

これを効率化するために，現在解 x と近傍解 x' のそれぞれの評価値の差 $g(= f(x') - f(x))$（ゲインと呼ぶ）をとる方法が有効である．現在解 x の j 番の要素番号のビット値 x_j を反転する 1-flip 近傍によって生成可能な 1-flip 近傍解のゲイン g_j は次式により $O(n)$ 時間で計算可能である．

$$g_j = d_{jj}(\overline{x}_j - x_j) + 2 \sum_{i=1, i \neq j}^{n} d_{ij} x_i (\overline{x}_j - x_j), \quad (2)$$

ここで，$\overline{x}_j$ は $1 - x_j$ である．したがって，n 個の 1-flip 近傍解の全ゲインを計算するためには全体で $O(n^2)$ となり，上述の $O(n^3)$ 時間よりも短縮できる．

さらにゲイン計算の効率化のために，現在の n 個の 1-flip 近傍解のそれぞれ n 個のゲインと次の近傍解のためのゲインの差 Δg_i $(\forall i \in N)$ の計算に基づいて，新たな n 個の全ゲインを線形時間で更新することが可能である．仮に x_j の反転が行われたとすると（$\Delta g_i(j) = 2\, d_{ij}\, (\overline{x}_i - x_i)\, (x_j - \overline{x}_j)$ とするとき），

$$g'_i = \begin{cases} -g_i & \text{if} \quad i = j \\ g_i + \Delta g_i(j) & \text{otherwise} \end{cases} \quad (3)$$

を用いて更新することにより，新たな n 個の 1-flip 近傍解の全ゲインが $O(n)$ 時間で計算できる[13)]．

本論文で示す局所探索法では，1-flip 近傍解のゲイン値を利用することで，2-flip 近傍の近傍解および k-flip 近傍の近傍解を評価する．よって，局所探索過程において 1 回のビット反転を施すたびに式 (3) を用いて 1-flip 近傍解の全ゲインの更新を行う．

3.5 k-flip 近傍解の評価値計算

k 個 $(1 < k < n)$ のビット値を一度に反転するような，一般化された，より大きなサイズの近傍を考える場合には，上述の 1-flip 近傍解の評価値の計算だけでは不十分である．ここでは，上述の 1-flip 近傍に基づくゲイン計算を利用した，k-flip 近傍のためのゲイン計算について示す．1-flip 近傍解の全ゲインがすでに計算され，かついくつかの異なる k 個のビット値を反転する場合（配列 flip[ ] に k 個の要素番号が保持されているとする），その k-flip 近傍解のゲイン G は次式により算出可能である．

$$
\begin{aligned}
G \;=\; & g_\alpha && (\textit{1-flip})\\
+\; & g_\beta + 2d_{\alpha\beta}(1-2x_\alpha)(1-2x_\beta) && (\textit{2-flip})\\
+\; & g_\gamma + 2d_{\beta\gamma}(1-2x_\beta)(1-2x_\gamma)\\
+\; & 2d_{\gamma\alpha}(1-2x_\gamma)(1-2x_\alpha) && (\textit{3-flip})\\
+\; & g_\delta + 2d_{\gamma\delta}(1-2x_\gamma)(1-2x_\delta)\\
+\; & 2d_{\delta\alpha}(1-2x_\delta)(1-2x_\alpha)\\
+\; & 2d_{\delta\beta}(1-2x_\delta)(1-2x_\beta) && (\textit{4-flip})\\
& \qquad\qquad\vdots && \vdots\\
=\; & \sum_{i=1}^{k} g_{\mathrm{flip}[i]}\\
+\; & 2\sum_{i=1}^{k-1}\sum_{j=i+1}^{k} d_{\mathrm{flip}[i]\mathrm{flip}[j]}\\
& (1-2x_{\mathrm{flip}[i]})(1-2x_{\mathrm{flip}[j]}), && (\textit{k-flip})
\end{aligned}
$$

ここで，$1 < k \le n$，$\mathrm{flip}[\,] = \{\alpha, \beta, \gamma, \delta, \cdots\}$ である．たとえば，現在解 x における α 番と β 番のビット値が一度に反転される場合（2-flip 近傍を想定）のゲイン G は，g_α，g_β と $2d_{\alpha\beta}(1-2x_\alpha)(1-2x_\beta)$ の和により計算可能である．

以降，この k-flip 近傍解の評価値計算を「一般化ゲイン計算」と呼ぶ．

4. MDP に対する k-flip 局所探索法

一般に，大きなサイズの近傍を有する局所探索法は，小さなサイズの近傍を有するものよりも優れた局所最適解を算出できる場合が多い．しかしながら，大きなサイズの近傍を採用する場合にはその近傍解の数が多くなることが一般的である．したがって，大きなサイズの近傍を有する局所探索法の計算量は，小さなサイズの近傍を有する局所探索法よりも大きい．

近傍に関するこのような性質をふまえて，ここでは，可変深度探索 (variable depth search，VDS) と呼ばれる，Lin と Kernighan のアイデア[8),12)] を導入した局所探索法（k-flip 局所探索法）を MDP に対して示す．VDS とは，単純な近傍操作を連鎖的に複数回実行することで生成されうる解集合をあらためて近傍と定義する，巧妙な近傍探索のアイデアである．一般的には，大きなサイズの近傍内の有望な，限られた小部分の近傍解のみを探索対象とするので，単純に大きな近傍を用いる近傍探索よりも計算時間を抑えることが多くの場合に可能である．

4.1 k-flip 局所探索法の基本戦略

MDP に対する k-flip 局所探索法の基本戦略は次のようになる．

(I) 実行可能解を初期解 x とする．

(II) 解 x に対して基本近傍操作（2-flip move）を連鎖的に実行することで，m 個の候補解 x' を生成する．

(III) その候補解 x' の中から最良の評価値を有する解を k-flip 近傍解 x_{best} として選ぶ．

(IV) 近傍解 x_{best} の評価値 $f(x_{best})$ が $f(x)$ よりも良好であるならば，x_{best} を x とし，(II) に戻る．そうでなければ，解 x を出力し，局所探索処理を終了する．

3.1 節で示した，局所探索の基本処理と照らし合わせて考えると，(ii) の「解 x の近傍 $\mathcal{NB}$ をもとに，近傍解 x' を選ぶ」という処理は，ここでの (II) および (III) の処理に対応し，より巧妙な近傍探索が行われる．(II) に示すように，MDP に対する k-flip 局所探索法では，2-flip 近傍操作（2-flip move と呼ぶ）を初期解 x^0 に対して行うことで候補解 x^1 を生成し，その x^1 から同様に 2-flip move によって x^2 を生成することを x^m まで連鎖的に行うことによって，k-flip 近傍解となりうる異なる m 個の候補解 $x' = \{x^1, x^2, \ldots, x^m\}$ を生成する．その候補解の中から最良の評価値を有する解を k-flip 近傍解として採用し，その近傍解を次の k-flip 近傍探索のための初期解 x にするという処理が基本となる．

本局所探索法の各繰返しにおける k-flip 近傍探索では，初期解 x をもとにして連鎖的に異なる m 個の k-flip 近傍候補解を生成するために，その生成途中では初期解 x の各要素のビット値は 1 回以上反転させないようにする．これにより，同じ解を再び探索する「サイクリング」の現象を回避することが可能になり，m 個の候補解 x' はすべて異なる解となる．したがって，個々の解のハミング距離 d_H は初期解 x との関係から，$d_H(x, x') = k$ $(k = \{2, 4, \ldots, 2m-2, 2m\})$ となる．この関係から，各繰返しにおいて選ばれる最良の k-flip 近傍解は，k が 2 から $2m$ の範囲で変動する近傍解を扱うので，可変のサイズの近傍を結果的

```
   procedure k-Flip-Local-Search(x, g)
   begin
1      repeat
2          x_prev := x, G_max := 0, G := 0, C1 := S_1, C0 := S_0;
3          repeat
4              find j with g_j = max_{j∈C1} g_j;
5              find k with gain = max_{k∈C0} (g_k + g_j + 2d_jk(1 − 2x_k)(1 − 2x_j));
6              G := G + gain;
7              x_j := 1 − x_j, x_k := 1 − x_k, swap j and k in the sets S_1 and S_0, and update gains g for each flipping;
8              C1 := C1\{j}, C0 := C0\{k};
9              if G > G_max then G_max := G, x_best := x;
10         until C1 = ∅ (or new x_best is not found for several repeats t);
11         if G_max > 0 then x := x_best else x := x_prev;
12     until G_max ≤ 0;
13     return x;
   end;
```

図 1 MDP に対する k-flip 局所探索法
Fig. 1 k-flip local search for MDP.

に探索することになる．上述したように，本局所探索法の各繰返しにおける k-flip 近傍探索では，k-flip 近傍候補解の数を m 個までと限定する．その理由は，$m < n/2$ の際に，各要素のビット値を 1 回以上反転させない条件を含めた 2-flip move によって生成可能な解の数はたかだか m 個となるためである．

4.2 *k*-flip 局所探索アルゴリズム

本 GLS で利用する，上述の基本戦略をベースにした k-flip 局所探索アルゴリズムを**図 1** に示す．

図 1 では，局所探索の初期解として実行可能解 x および x に対応する 1-flip 近傍解の n 個の全ゲインを有する g があらかじめ与えられると想定する．全ゲイン g は式 (2) より算出可能である．加えて，局所探索の途中の全過程で，0-1 表現の解 x はつねに S_1 と S_0 による表現と一致するものとする．

k-flip 局所探索法は，主に内ループと外ループの 2 つのループにより構成される．内ループは，m 個の k-flip 近傍解の候補を生成する操作とその候補中にある最良解を k-flip 近傍解として選び出す処理である．外ループは，内ループで得られた k-flip 近傍解を評価し，k-flip 局所探索法の終了条件の判定を主に行う．

内ループの最初の処理として，2 つの集合 $C1$ と $C0$ を準備する．$C1$ には $x_i = 1$ となる要素番号を保持し，$C0$ には $x_i = 0$ となる要素番号を保持する．これにより，与えられた解 x の各要素のビット値を 1 回以上反転させないことを保証する（候補解生成の詳細については後述する）．内ループは，$C1 = \emptyset$ の条件を満足するまで反復されることになる（ステップ 10）．つまり，この条件は上述の基本戦略での条件である，m 個の候補解を生成するまで反復することと同等の条件となる．

しかしながら，多くの場合，m 個すべての候補解を生成する必要はないと考えられる．なぜなら，候補解の生成途中で，最良解として選ぶべき最良ゲイン値を持つ k-flip 近傍解には明らかになりえないような候補解をも生成する可能性が高いからである．本局所探索法では，内ループにおける暫定の最良解 x_{best} の更新が数回の内ループ反復後においてもなされなければ，内ループ処理を強制的に終了するという終了条件をステップ 10 に追加する（x_{best} の更新のチェックはステップ 9 の判定処理を利用することにより可能）．これにより，m 個すべての候補解を生成しなくても，上述の（m 個すべての候補解を生成する）基本戦略において見つかるであろう x_{best} を k-flip 近傍解として選出できることが多くの場合に可能となり，基本戦略における内ループの終了条件（$C1 = \emptyset$ の条件）だけの場合よりも処理時間の点で大幅な短縮が期待できる．むろん，ステップ 10 において追加した内ループ終了条件である several repeats の回数 t の設定によって，得られる解の質と探索処理時間のトレードオフを制御することが容易になるが，より高速な処理を要求する場合にのみ設定すればよく，必須のものではない．本研究の場合では，遺伝的局所探索法の局所探索プロセスにおいて k-flip 局所探索法を利用するため，より高速な処理が望ましい．本研究では，簡単な予備実験の結果に基づき，得られる解の質とその処理時間のトレードオフから，その回数 t を $n \geq 500$ の問題例に対しては $t = m/5$，$n < 500$ の問題例に対しては $t = m$ と設定する．なお，より巧妙なこの種の設定に関しての研究もあり，文献 10) に詳しい．

次に，候補解を生成するプロセス（ステップ 4〜8）について記述する．本局所探索法では 2-flip move を用いて各候補解の生成を行う．ここで，上述した 2 つの集合 $C1$ と $C0$ を用いる．ステップ 4 では $C1$ に含

まれる要素番号を対象にして，最大のゲイン値を持つ要素番号 j を見つける．ステップ 5 は，$C0$ に含まれる要素番号を対象にして，ステップ 4 で見つかった j をもとに，3.5 節で示した 2-flip 近傍までの一般化ゲイン計算によって要素番号 k を見つける処理である．見つかった 2 つの j と k はそれぞれ $C1$ と $C0$ に含まれていたので，解の j 番と k 番の要素値それぞれを反転したとしても実行可能解がつねに生成可能である．ステップ 6 では，j 番と k 番の要素値を反転した際の解のゲインを暫定的に算出し，ステップ 7 で実際に反転を行うとともに，次の候補解探索のために，式 (3) を用いてゲインの更新を行う．さらに，ステップ 8 では反転に使用した 2 つの要素番号のそれぞれを対応する集合 $C1$ と $C0$ から削除する．

以上の候補解生成プロセスを先ほど述べたステップ 10 の 2 つの条件のどちらかを満たすまで反復した後，内ループを終了する．その後，内ループ中で見つかった k-flip 近傍解 x_{best} をステップ 11 およびステップ 12 の条件式に従って評価し，次の k-flip 近傍探索を繰り返すか否かを決定する．なお，本 k-flip 局所探索法の内ループの（つまり，k-flip 近傍解の探索に必要な）時間量は，候補解の探索において $\mathrm{O}(m|S_1| + m|S_0|)$，ゲインの更新において $\mathrm{O}(mn)$ である．

5. MDP に対する遺伝的局所探索

遺伝的局所探索（GLS）は，遺伝的アルゴリズムと局所探索のハイブリット手法であり，ハイブリッド GA や Memetic Algorithm [15] などとも呼ばれる．

困難な最適化問題に対する GLS の基本戦略は，集団中の各個体（解）を局所探索法により局所最適解に到達させ，局所最適解の集団に対して遺伝的操作である，交叉，突然変異，選択などの各操作を適用するという一連の処理を繰り返す．このような戦略をとることで，局所探索法を併用しない単純な GA に比べ，対象として主に考慮するべき探索空間は格段に減少する[7]．さらに，膨大に存在するであろう局所最適解群の最良解を “真の最適解” と見なせば，局所探索法を併用する GLS はその探索の戦略において潜在的な優位性を持ちあわせている．

MDP に対する我々の GLS の基本処理を**図 2** に示す．まず貪欲に実行可能解を PS 個生成し，その後，実行可能領域に探索を絞った k-flip 局所探索法（図 1）を適用することで，局所的に最適化された解（局所最適解）の集合を初期の個体集団とする．その後，個体（解）集団に対して交叉操作により子孫を生成する．子孫の実行可能性を保証するために実行可能解変換操作（リペア法）を施し，変換された実行可能解からスタートする k-flip 局所探索法を実行する．生成された子孫群と親群から良好な PS 個の個体を次世代集団として世代を重ねていく．

```
procedure GLS
begin
    initialize a population P ∈ {I_1, ..., I_PS};
    foreach indiv. I ∈ P do
        I := Local-Search(I);
    endforeach
    repeat
        for i := 1 to #crossovers do
            choose two parents I_a, I_b ∈ P randomly;
            I_c := Crossover(I_a, I_b);
            I_c := Repair(I_c);
            I_c := Local-Search(I_c);
            add an individual I_c to P_c;
        endfor
        P := Selection(P, P_c);
        if diversification=true then
            foreach indiv. I ∈ P\{best indiv. } do
                I := Mutation(I);
                I := Repair(I);
                I := Local-Search(I);
            endforeach
        endif
    until terminate=true;
    return best individual ∈ P;
end;
```

図 2 MDP に対する遺伝的局所探索法の流れ
Fig. 2 Genetic local search for MDP.

上述の処理だけでは GLS の解集団が準最適探索空間に陥る可能性が高く，準最適探索空間からの脱出が困難となる．この対策として多様化/再スタート戦略を導入する．この戦略では，ビット値反転操作に基づく突然変異，リペア法，局所探索法の各操作を行い，陥った準最適探索空間から他の探索空間への脱出を試みる．

以下では，前述した局所探索のプロセスを除き，MDP に対する本 GLS の各操作について記述する．

5.1 初期集団の生成

MDP に対する GLS の初期集団に含まれる各解は，Merz らにより示されたランダム貪欲法[13] に簡単な修正を加えることによって解の実行可能性を考慮しつつ生成する．ランダム貪欲法はランダム性を有した局所的な評価基準に基づく 0-1 割当てを各変数に対して行うが，本 GLS における貪欲解法では，その割当ての際，1 をとる変数の数を MDP の制約条件を満たすまでカウントするように修正することにより実行可能解を生成する．

5.2 交　　叉

図 2 に示すように，交叉，リペア法，局所探索で構成される一連の処理中に交叉操作が含まれる．本 GLS

```
procedure Repair(x, g)
begin
    calculate a violation v := m − |S_1|;
    if v = 0 then return x;
    else if v < 0 then
        repeat
            find j with g_j = max_{j∈S_1} g_j;
            x_j := 1 − x_j, S_1 := S_1\{j};
            update gains g;
        until Σ_{i=1}^{n} x_i = m;
        return x;
    else
        repeat
            find j with g_j = max_{j∈S_0} g_j;
            x_j := 1 − x_j, S_0 := S_0\{j};
            update gains g;
        until Σ_{i=1}^{n} x_i = m;
        return x;
    endif
end;
```

図 3 MDP に対するリペア法
Fig. 3 Repair method for MDP.

では，一様交叉[16] に基づいて交叉を行い，1 回の交叉操作で 1 つの子孫を生成する．なお，集団サイズ PS 個の個体が重複しないようにランダムにペアリングした $PS/2$ 組のそれぞれを各交叉における 2 つの親解とする．したがって，各世代での交叉回数はつねに $PS/2$ 回と固定であり，$PS/2$ 個の子孫解が生成される．

5.3 リペア法

前述の交叉操作（および後述の突然変異操作）によって生成される解の実行可能性は保証されない．そこで本 GLS では，交叉および突然変異の操作後の解に対してリペア法を施し，実行可能解に変換する．

MDP に対するリペア法を**図 3** に示す．本リペア法では，まず実行不可能解か否かを判定し，実行不可能とする数分のビット値を反転することで実行可能解へ変換する．各反転の際に選ばれるビットは最大のゲインに対応するものとし，結果的に変換される解ができるだけ良い評価となるようにする．

5.4 選択・淘汰

本 GLS で利用する選択淘汰操作は，解の評価値に基づいて行われる．上記したように，交叉では親個体 PS 個から子孫個体 $PS/2$ を生成する．あらかじめ定めた集団サイズ PS 個を保つよう $PS + PS/2$ 個の解から良好な PS 個をその評価値に基づいて次世代に残す．ただし，次世代集団の多様性を考慮し，子孫個体と同一の評価値を有する重複の親個体は淘汰し，次世代集団内に同一の評価値を有する個体が存在しないようにする．

5.5 多様化/再スタート戦略

一般に，多くの最適化問題ではアルゴリズムの探索空間において準最適領域が数多く複雑に存在すると考えられる．よって，突然変異操作を含まない，上述の遺伝的探索処理だけでは GLS の解集団が準最適領域に陥る可能性があり，そこからの脱出が困難となる．この対策として多様化/再スタート戦略[2] を導入する．本 GLS では，集団中のエリート解の更新が 30 世代以上行われない場合にこの戦略を実施する．この戦略では，集団中のエリート解を除く $PS - 1$ 個の各個体に対して，ランダムに選んだ $n/2$ 個のビット値反転操作に基づく突然変異を行い，生成された子孫に対してリペア法を施し実行可能解を得る．その解を局所探索し，新たに多様化した集団から遺伝的局所探索を再びスタートする．これによって，準最適探索空間から真の最適解が存在する領域へと探索の移動が高確率で促され，結果的に GLS の探索性能の向上が期待できる．

6. 実 験 結 果

k-flip 局所探索法を有する GLS の有効性を示すために，後述する 2500 変数までの大規模な問題例に対して実験を行い，素朴な近傍を持つ 2-flip 局所探索法（付録参照）を有した GLS との比較を行う．

6.1 問 題 例

MDP に対する既存研究[4),5),9)] では，研究者自身が作成した問題例のみを利用しており，さらにそれらは公開されておらず入手困難である．よって，本実験では同じ問題例への適用が困難であるため，既存解法と本 GLS の厳密な比較が不可能である．そこで我々は，今後の追研究をも考慮し，ORLIB[1)] から入手可能なバイナリー 2 次計画問題（binary quadratic programming problem）のベンチマーク群から，MDP の定式化に沿うように $d_{ii} = 0$ と修正した問題（`beas500-1`（500 変数），`beas1000-1`（1000 変数），`beas2500-1`（2500 変数））を使用する．これらの各行列 d の density（行列 d に 0 以外の数値が含まれる割合）はどれも 10%である．なお，d_{ij} のいくつかの値は負値を含む．さらに我々は，問題サイズ n が 100，250，500，750，1000 および 2500 変数の問題群（それぞれ，`mdp00100`，`mdp00250`，`mdp00500`，`mdp00750`，`mdp01000`，`mdp02500`）をランダムに生成した．各行列の density は 100%で，d_{ij} の各値は非負である．これら 6 つの問題群は，`http://k2x.ice.ous.ac.jp/~katayama/bench/` から入手可能である．なお，我々が適用する問題群のサイズは既存研究の多くで適用されたものよりもはるかに大規模である．

上述の全 9 問題群は行列 d を与えるのみであるため，本研究では m を各問題群の n の 10%，20%，30%，

表 1　k-flip 局所探索を有した遺伝的局所探索法の結果
Table 1　Results for GLS with k-flip local search.

instance (off diag.)			GLS with k-flip Local Search								
name	n	m	best	(%)	avg.	(%)	b/run	t1	(gens)	t2	(gens)
mdp00100	100	10	3606	(0)	3606.0	(0)	30/30	0.1	(4)	—	(—)
	100	20	12956	(0)	12956.0	(0)	30/30	0.1	(1)	—	(—)
	100	30	27036	(0)	27036.0	(0)	30/30	0.1	(1)	—	(—)
	100	40	45872	(0)	45872.0	(0)	30/30	0.2	(2)	—	(—)
mdp00250	250	25	20834	(0)	20834.0	(0)	30/30	1.0	(6)	—	(—)
	250	50	75816	(0)	75816.0	(0)	30/30	1.8	(5)	—	(—)
	250	75	162252	(0)	162252.0	(0)	30/30	14.4	(40)	—	(—)
	250	100	279470	(0)	279470.0	(0)	30/30	2.5	(4)	—	(—)
mdp00500	500	50	78898	(0)	78898.0	(0)	30/30	5.1	(15)	—	(—)
	500	100	291916	(0)	291916.0	(0)	30/30	4.5	(5)	—	(—)
	500	150	631898	(0)	631898.0	(0)	30/30	12.9	(24)	—	(—)
	500	200	1096092	(0)	1096092.0	(0)	30/30	1.7	(2)	—	(—)
beas500-1	500	50	21750	(0)	21750.0	(0)	30/30	1.7	(25)	—	(—)
	500	100	48738	(0)	48738.0	(0)	30/30	1.0	(6)	—	(—)
	500	150	73544	(0)	73544.0	(0)	30/30	9.6	(62)	—	(—)
	500	200	95516	(0)	95516.0	(0)	30/30	0.5	(1)	—	(—)
mdp00750	750	75	171704	(0)	171704.0	(0)	30/30	74.5	(99)	—	(—)
	750	150	641140	(0)	641140.0	(0)	30/30	12.6	(6)	—	(—)
	750	225	1395672	(0)	1395672.0	(0)	30/30	106.3	(71)	—	(—)
	750	300	2430660	(0)	2430660.0	(0)	30/30	34.4	(16)	—	(—)
mdp01000	1000	100	299730	(0)	299721.7	(0.002758)	28/30	456.7	(249)	1000	(524)
	1000	200	1125264	(0)	1125264.0	(0)	30/30	165.4	(57)	—	(—)
	1000	300	2458316	(0)	2458316.0	(0)	30/30	375.0	(102)	—	(—)
	1000	400	4292438	(0)	4292438.0	(0)	30/30	81.3	(16)	—	(—)
beas1000-1	1000	100	62102	(0)	62102.0	(0)	30/30	22.1	(64)	—	(—)
	1000	200	141512	(0)	141512.0	(0)	30/30	41.6	(79)	—	(—)
	1000	300	218478	(0)	218478.0	(0)	30/30	177.8	(245)	—	(—)
	1000	400	284688	(0)	284688.0	(0)	30/30	57.0	(65)	—	(—)
mdp02500	2500	250	1775366	(0)	1774418.9	(0.053349)	3/30	2313.4	(163)	3000	(203)
	2500	500	6794586	(0)	6793782.4	(0.011827)	2/30	930.3	(30)	3000	(123)
	2500	750	14988200	(0.001575)	14987346.8	(0.007267)	0/30	—	(—)	3000	(88)
	2500	1000	26332276	(0)	26332274.3	(0.000007)	29/30	927.1	(19)	3000	(82)
beas2500-1	2500	250	246678	(0)	246678.0	(0)	30/30	581.9	(248)	—	(—)
	2500	500	566928	(0)	566473.1	(0.080245)	9/30	1323.0	(318)	3000	(709)
	2500	750	882276	(0)	882240.0	(0.004080)	28/30	720.2	(118)	3000	(506)
	2500	1000	1153068	(0)	1151871.1	(0.103798)	1/30	1245.2	(189)	3000	(397)
mdp02500	2500	750	14988436	(0)	14988307.6	(0.000857)	20/30	15000.3	(432)	30000	(807)

40%とした．たとえば，1000 変数の MDP の場合，$m = 100$，200，300，400 の計 4 つの問題例が構成できる．

6.2　結果と考察

全 9 問題群（36 問題例）は現在までに適用例のない新たな MDP の問題例であるので，各問題例の既知の最良解を示すことは今後の追研究に対して重要であると考えられる．よって，k-flip 局所探索法を有する GLS（k-flipGLS）を比較的長時間にわたり実行した．その計算時間（GLS の計算打切り時間）は Sun Ultra 5/10 ワークステーション（UltraSPARC-IIi 440MHz）使用のもとで，100 変数の問題例に対しては 10 秒，250 変数の問題例に対しては 30 秒，500 変数の問題例は 100 秒，750 変数の問題例は 300 秒，1000 変数の問題例は 1000 秒，2500 変数の問題例は 3000 秒とした．アルゴリズムは C によりコード化し，gcc（最適化オプション -O2）でコンパイルした．

k-flipGLS の結果を**表 1** に示す．左の 3 列は問題群の名前，問題サイズ n，サイズ m である．以降，30 回の試行中に得られた最良解値 “best” とその解質（解質 (%) = 100 × (既知の最良解値/ 得られた解の値) − 1.0），平均の解値 “avg” とその解質，30 回の試行で既知の最良解を得た回数 “b/run”，既知の最良解を得るまでに要した平均時間 “t1”（秒）と GLS で要した，“t1” に対応する平均世代数 “(gens)”，1 試行あたりの計算打切り時間 “t2”（秒）と対応する平均世代数 “(gens)” である．列 “t2” では，その打切り時間内に既知の最良解をすべての試行で算出した場合に限り “—” と記した．

パラメータの設定値として，GLS の集団サイズ PS は 40 である．その他，実行に必要なパラメータ設定値についてはすでに 4.2 節（k-flip 局所探索法の内ループ強制終了条件），5.2 節（交叉）および 5.5 節（突然変異に関する多様化/再スタート戦略）に記述している．

表 1 の結果より，100 変数から 1000 変数の問題例に対しては，mdp01000 の $m = 100$ の問題例を除き，最良解 “best” を算出した回数 “b/run” が 30 である

表 2 2-flip 局所探索を有した遺伝的局所探索法の結果
Table 2 Results for GLS with 2-flip local search.

instance (off diag.)			GLS with 2-flip Local Search								
name	n	m	best	(%)	avg.	(%)	b/run	t1	(gens)	t2	(gens)
mdp00100	100	10	3606	(0)	3606.0	(0)	30/30	0.1	(14)	—	(—)
	100	20	12956	(0)	12956.0	(0)	30/30	0.1	(4)	—	(—)
	100	30	27036	(0)	27036.0	(0)	30/30	0.1	(3)	—	(—)
	100	40	45872	(0)	45872.0	(0)	30/30	0.1	(7)	—	(—)
mdp00250	250	25	20834	(0)	20834.0	(0)	30/30	1.2	(50)	—	(—)
	250	50	75816	(0)	75816.0	(0)	30/30	0.9	(27)	—	(—)
	250	75	162252	(0)	162252.0	(0)	30/30	3.0	(109)	—	(—)
	250	100	279470	(0)	279470.0	(0)	30/30	1.4	(46)	—	(—)
mdp00500	500	50	78898	(0)	78898.0	(0)	30/30	6.8	(78)	—	(—)
	500	100	291916	(0)	291916.0	(0)	30/30	7.1	(84)	—	(—)
	500	150	631898	(0)	631898.0	(0)	30/30	8.5	(100)	—	(—)
	500	200	1096092	(0)	1096092.0	(0)	30/30	7.6	(77)	—	(—)
beas500-1	500	50	21750	(0)	21750.0	(0)	30/30	2.1	(112)	—	(—)
	500	100	48738	(0)	48738.0	(0)	30/30	0.9	(55)	—	(—)
	500	150	73544	(0)	73544.0	(0)	30/30	22.8	(955)	—	(—)
	500	200	95516	(0)	95516.0	(0)	30/30	0.7	(23)	—	(—)
mdp00750	750	75	171704	(0)	171703.9	(0.000078)	28/30	66.5	(265)	300	(1016)
	750	150	641140	(0)	641140.0	(0)	30/30	23.0	(84)	—	(—)
	750	225	1395672	(0)	1395666.0	(0.000406)	29/30	131.1	(491)	300	(1034)
	750	300	2430660	(0)	2430658.4	(0.000066)	29/30	73.8	(297)	300	(1113)
mdp01000	1000	100	299730	(0)	299642.5	(0.029204)	8/30	459.4	(717)	1000	(1610)
	1000	200	1125264	(0)	1125180.9	(0.007382)	18/30	516.3	(832)	1000	(1518)
	1000	300	2458316	(0)	2458290.6	(0.001033)	11/30	444.2	(756)	1000	(1557)
	1000	400	4292438	(0)	4292438.0	(0)	30/30	49.4	(100)	—	(—)
beas1000-1	1000	100	62102	(0)	62097.9	(0.006548)	29/30	188.1	(2013)	1000	(9906)
	1000	200	141512	(0)	141512.0	(0)	30/30	161.9	(1601)	—	(—)
	1000	300	218478	(0)	218406.5	(0.032742)	1/30	46.0	(529)	1000	(9344)
	1000	400	284688	(0)	284653.9	(0.011990)	14/30	179.8	(1796)	1000	(9546)
mdp02500	2500	250	1775088	(0.015659)	1773509.8	(0.104553)	0/30	—	(—)	3000	(728)
	2500	500	6794586	(0)	6792688.4	(0.027928)	1/30	2136.2	(581)	3000	(650)
	2500	750	14988018	(0.002789)	14987321.9	(0.007433)	0/30	—	(—)	3000	(695)
	2500	1000	26332276	(0)	26331889.6	(0.001467)	13/30	762.9	(246)	3000	(699)
beas2500-1	2500	250	246678	(0)	245780.3	(0.363929)	2/30	2239.9	(3791)	3000	(4865)
	2500	500	566928	(0)	565147.3	(0.314008)	1/30	730.5	(1333)	3000	(4132)
	2500	750	882276	(0)	881097.2	(0.133609)	4/30	1558.5	(2299)	3000	(4114)
	2500	1000	1151796	(0.110314)	1150995.9	(0.179706)	0/30	—	(—)	3000	(4141)

ことが分かる．よって，試行 30 回中 30 回とも同じ最良解を算出可能であるという傾向が確認できることから，実質的な最適解もしくは最適解にきわめて近い高品質な解が得られたものと考えられる．2500 変数の問題例に対しては，そのような傾向が若干薄れるが，問題例によっては試行 30 回の全回もしくは 30 回に近い頻度で高品質な解を得ている．なお，表 1 の最下の結果（mdp02500 の $m = 750$ の問題例）は，追加実験として k-flipGLS の計算打切り時間を 30000 秒にした場合のものである．この結果から，k-flipGLS は 3000 秒の場合では算出されなかった $f(x) = 14988436$ の解を 30 回試行中 20 回算出しており，より長い計算時間を許すことでより良好な解が平均的に得られることが観測された．

比較解法である，2-flip 局所探索法を有する GLS（2-flipGLS）の結果を**表 2** に示す．なお，表中の各項目，パラメータ設定値および計算打切り時間は，公平な比較を行うために，k-flipGLS の場合とまったく同じとした．表 2 の 100 変数から 500 変数までの問題例において，k-flipGLS で得た最良解と同じ解値を 30 回の試行全回で得ており，その最良解を得るまでの平均時間 “t1” も含め，両 GLS の探索性能はほぼ同等であることが分かる．

上述した GRASP [4] などの既存研究の多くでは，40 変数以下の小規模な問題例までを対象に数値実験が行われていたので，厳密な比較はできないものの，両 GLS は大規模な問題例への適用が可能および少なくとも 500 変数までの問題例に対してはきわめて高品質な解を頻繁にかつ比較的短時間に算出可能であることから，それらの既存解法と同程度もしくはそれ以上の性能を有するものと推測できる．

両 GLS の比較において，1000 変数以上の問題規模から k-flipGLS の優位性がうかがえる．mdp01000 および beas1000-1 の 8 問題例のうち，30 回試行で最良解をつねに算出したのは，2-flipGLS の場合，2 問題例のみであった．一方，k-flipGLS の場合では，同じ計算打切り時間にもかかわらず 8 問題例中 7 問でつねに既知の最良解を算出可能であった．2500 変数の 8 問題例に対しては，k-flipGLS は 2-flipGLS よりも “b/run” の回数がほとんどの問題例の場合において多

く，得られる平均の解質もすべて優れていることが観測できる．

以上より，k-flipGLS は 2500 変数のような大規模問題例に対しても平均的に良好な近似解を算出可能であり，小規模から中規模な問題例に対しては，本研究で示された既知の最良解を現実的な時間内にかつ頻繁に算出可能であることが確認された．また，2-flipGLS との比較を通して，高性能な局所探索の併用の有効性も明らかになった．

最後に，実験結果から観測される興味深い傾向について考える．表 1 および表 2 の結果を見ると，m の設定値である 30% の問題例（たとえば，`mdp00250` の $m = 75$，`beas500-1` の $m = 150$，`beas1000-1` の $m = 300$ など）では，他の m の設定値を与えた問題例の場合に比べ，GLS は既知の最良解に到達するまでに比較的多くの計算時間を必要とする（または既知の最良解に到達する頻度が比較的少なくなる）場合が多い．この興味深い傾向は，k-flipGLS の場合だけでなく 2-flipGLS の結果からも観測できることから，探索アルゴリズムの違いによる傾向よりもむしろ，問題例の特徴☆の 1 つである m が 30% の問題例の場合に，そのような傾向を引き起こす何らかの原因があるものと考えられる．その原因については，現在のところ明らかではないものの，考えられる事項の 1 つは探索空間内の局所最適解の分布に何らかの関係があるものと思われる．そのような局所最適解の分布に関する一研究として文献 7) があげられ，バイナリー 2 次計画問題の場合では局所最適解の分布の傾向によって最適解への探索を困難にする問題例の存在が明らかにされている．また，他の組合せ最適化問題（グラフ彩色問題など）においても，極端に解くことが難しくなる問題例の存在が知られている[3),14)]．さらに，最適解の算出を困難とする現象とその現象を引き起こす理由に関して興味深い仮説を提唱した研究[6)] もある．上述の局所最適解の分布に関する研究やその他関連する研究を通して，対象とする最適化問題（またはその問題例）をより詳細に調査することは，より強力なアルゴリズムの開発の助長につながると期待されるとともに，重要な研究課題であると考える．

7. おわりに

本論文では，最大多様性問題（MDP）に対して可変深度探索に基づく k-flip 局所探索法を有した遺伝的局所探索法（GLS）を示した．本 GLS の有効性を示すために，既存研究の多くで適用された問題サイズをはるかに上回る大規模な MDP の問題例を準備し，それらの問題例を対象として数値実験を行った．100 変数から 2500 変数までの問題例に対して，k-flip 局所探索法ベースの GLS および 2-flip 局所探索法ベースの GLS を公平な比較のもとで適用した結果，500 変数規模までの問題例に対する k-flip 局所探索法ベースの GLS の性能は，2-flip 局所探索法ベースの GLS と同程度であることが判明した．しかしながら，それ以上の規模の問題例に対しては，k-flip 局所探索法を有する本 GLS の有効性が示された．特に 1000 変数程度までの問題例に対しても，本 GLS の場合には，きわめて高品質な解を頻繁に算出可能であり，さらに 2500 変数の問題例においても，2-flip 局所探索法を有する GLS との比較のもとで，より良好な解を平均的に算出可能であることを示した．これらのことから，MDP に対する GLS の枠組みにおいては，より高性能な局所探索を併用することの有効性も明らかとなった．

一般に，最適化問題に対するメタ戦略アルゴリズムの多くでは，既存の局所探索法を大きな変更なしにメタ戦略の枠組みに導入することが可能である．よって，本 GLS で利用した k-flip 局所探索法を GLS 以外のメタ戦略アルゴリズムへ導入することも可能であり，その利用範囲は比較的広いと考えられる．

今後の課題としては，本研究で準備した問題例を対象として他のメタ戦略アルゴリズムとの比較を行うこと，さらに大規模な問題例への適用や問題例の設定値 m に関する検討などが考えられる．

☆ MDP の問題例を特徴づける要素は，m だけでなく，行列 d の density，行列 d に含まれる 0 以外の数値の分布状況やその範囲なども含まれることに注意しなければならない．

参 考 文 献

1) Beasley, J.E.: OR-Library: Distributing Test Problems by Electronic Mail, *Journal of the Operational Research Society*, Vol.41, pp.1069–1072 (1990).
2) Eshelman, L.: The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination, *Foundations of Genetic Algorithms*, Rawlings, G.J.E. (Ed.), pp.265–283 (1991).
3) Frank, J., Cheeseman, P. and Stutz, J.: When Gravity Fails: Local Search Topology, *Journal of AI Research*, Vol.7, pp.249–281 (1997).
4) Ghosh, J.B.: Computational Aspects of the Maximum Diversity Problem, Operations Research Letters, Vol.19, pp.175–181 (1996).
5) Glover, F., Kuo, C.-C. and Dhir, K.S.: Heuristic Algorithms for the Maximum Diversity

Problem. *Journal of Information and Optimization Sciences*, Vol.19, pp.109–132 (1998).

6) 池田　心, 小林重信：GA の探索における UV 現象と UV 構造仮説, 人工知能学会論文誌, Vol.17, pp.239–246 (2002).

7) 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, pp.1258–1271 (2001).

8) Kernighan, B.W. and Lin, S.: An Efficient Heuristic Procedure for Partitioning Graphs, *Bell System Technical Journal*, Vol.49, pp.291–307 (1970).

9) Kochenberger, G. and Glover, F.: Diversity Data Mining, Technical Report HCES-03-99, Hearin Center for Enterprise Science College of Business Administration (1999).

10) 河本敬子, 片山謙吾, 成久洋之：バイナリー２次計画問題に対する k-opt 局所探索法の効率化, 電子情報通信学会論文誌（D-I）, Vol.J85-D-I, pp.322–328 (2002).

11) Kuo, C.-C., Glover, F. and Dhir, K.S.: Analyzing and Modeling the Maximum Diversity Problem by Zero-One Programming, *Decision Sciences*, Vol.24, pp.1171–1185 (1993).

12) Lin, S. and Kernighan, B.W.: An Effective Heuristic Algorithm for the Traveling Salesman Problem, *Operations Research*, Vol.21, pp.498–516 (1973).

13) Merz, P. and Freisleben, B.: Greedy and Local Search Heuristics for Unconstrained Binary Quadratic Programming, *Journal of Heuristics*, Vol.8, pp.197–213 (2002).

14) 水野一徳, 西原清一：確率的制約充足アルゴリズムにおける局所最適構造, 人工知能学会論文誌, Vol.16, pp.38–45 (2001).

15) P. Moscato: Memetic Algorithm. http://www.densis.fee.unicamp.br/˜moscato/memetic_home.html

16) Syswerda, G.: Uniform Crossover in Genetic Algorithms, *Proc. 3rd International Conference on Genetic Algorithms*, pp.2–9 (1989).

17) Weitz, R.R. and Lakshminarayanan, S.: An Empirical Comparison of Heuristic and Graph Theoretic Methods for Creating Maximally Diverse Groups, VLSI Design, and Exam Scheduling, *Omega*, Vol.25, pp.473–482 (1997).

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).

```
procedure 2-Flip-Local-Search(x, g)
begin
  repeat
    find j with g_j = max_{j∈S_1} g_j;
    find k with G = max_{k∈S_0} (g_k + g_j + 2d_{jk}(1 − 2x_k)(1 − 2x_j));
    if G > 0 then
      x_j := 1 − x_j and x_k := 1 − x_k (swap j and k in the sets S_1 and S_0);
      update gains g for each flipping;
    endif
  until G ≤ 0;
  return x;
end;
```

図 4　MDP に対する 2-flip 局所探索法
Fig. 4　2-flip local search fo MDP.

付　　録

最大多様性問題に対する 2-flip 局所探索アルゴリズムを**図 4** に示す．4.2 節で示した k-flip 局所探索アルゴリズムと同様に，初期解として実行可能解 x および x に対応する 1-flip 近傍解の n 個の全ゲイン g があらかじめ与えられると想定するとともに，0-1 表現の解 x はつねに S_1 と S_0 による表現と一致するものとする．

(平成 15 年 4 月 10 日受付)
(平成 15 年 6 月　6　日再受付)
(平成 15 年 8 月 22 日採録)

片山　謙吾（正会員）

平成 4 年岡山理科大学工学部電子工学科卒業．平成 7 年同大学大学院修士課程修了．平成 10 年同大学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 年同大学工学部情報工学科講師．メタヒューリスティックス，組合せ最適化，強化学習に関する研究に従事．電子情報通信学会，人工知能学会各会員．

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教授．昭和 58 年〜平成 6 年岡山理科大学情報処理センター所長．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．日本オペレーションズリサーチ学会フェロー．電子情報通信学会会員．