

## 探索傾向に基づいた知識の導入による $k$ -opt 局所探索法の研究

河本 敬子・片山 謙吾\*・成久 洋之\*

岡山理科大学大学院工学研究科博士課程システム科学専攻

\*岡山理科大学工学部情報工学科

(2001年11月1日 受理)

### 1 序論

近年、最適化問題に対する効率的な解法として、遺伝的アルゴリズム (Genetic Algorithm, GA), アニーリング法, タブー探索法やそれらの変形版などのアルゴリズムであるメタ戦略が提案されてきた. メタ戦略は, 基本解法である局所探索法 (Local Search, LS) などと自然界に存在するアイデアを結合させることによって, 比較的短時間に高品質な近似解を算出可能である. この一例として, GA に LS を組み込んだ遺伝的局所探索法 (Genetic Local Search, GLS) などが挙げられるが, GLS で費やされる探索時間の多くは LS が占めている場合が多い. このことから, メタ戦略で多く用いられる LS の性能を維持しつつ, 探索時間の大幅な効率化を可能にする方法の提案が期待されている.

本論文では, バイナリー 2 次計画問題 (Binary Quadratic Programming Problem, BQP) を取り上げ, BQP に対して知られている LS の中でも, 最も有効な  $k$ -opt 局所探索法 ( $k$ -opt 法) に焦点をあてる. この  $k$ -opt 法は, 1960 年代後期から 70 年代前期において, 巡回セールスマン (traveling salesman problem) およびグラフ分割問題 (graph partitioning problem) に対して Lin と Kernighan により提案された局所探索法 [13, 16] のアイデア (可変深度探索, variable depth search) に基づき, 近年 BQP に対して Merz と Freisleben によって提案された [18]. BQP に対する  $k$ -opt 法では, 良好な近傍解を探索し, その近傍解への移動を繰り返すことで最終的に良質な近似解を算出する. しかしながら, 近傍解探索のプロセスは必ずしも効率的ではないため, 近傍解探索を打切るためのパラメータが準備されている. このパラメータ設定によって  $k$ -opt 法自体の探索時間がある程度短縮できることが知られているが, 十分な知識の導入による検討はなされていない.

本論文の目的は,  $k$ -opt 法の近傍解の探索傾向に基づいた知識の導入による大幅な探索時間の効率化手法の提案である. 本論文では, 2 でバイナリー 2 次計画問題, 3.1 で BQP に対する  $k$ -opt 局所探索法について記述する. 次に, 3.3 で  $k$ -opt 法における既存のパラメータ設定による近傍解の探索傾向を分析し, その観測結果を用いて Merz らが示したパラメータ設定 [18] をベースにした新たな方法を 4 に提示する. その結果, 既存のパラメータ設定との比較において, 同等の探索性能を維持しつつ, 探索時間を大幅に短縮可能であることを示す.

### 2 バイナリー 2 次計画問題

バイナリー 2 次計画問題 (BQP) とは,  $n \times n$  の対称行列  $Q = (q_{ij})$  が与えられたとき, 次の目的関数

$$f(x) = x^t Q x = \sum_{i=1}^n \sum_{j=1}^n q_{ij} x_i x_j, \\ x_i \in \{0, 1\} \forall i = 1, \dots, n \quad (1)$$

を最大化する解  $x$  を求める問題である [5]. BQP は, NP-困難であり, 数多くの応用例を有している. 例として, CAD 問題 [15], マシンスケジューリング問題 [1], capital budgeting and financial analysis 問題 [17], traffic message management 問題 [6], 分子構造問題 [21] などがある. 更に, BQP は様々な組合せ最適化問題と同等であることが知られている. そのような問題として, 最大カット問題, 最大クリーク問題, 最大頂点パッキング問題, 最小頂点被覆問題, 最大独立集合問題などがある [3, 5, 8, 19, 20].

### 3 BQP に対する $k$ -opt 局所探索法

#### 3.1 $k$ -opt 局所探索法

最初に、局所探索法で利用される解の表現について記述する。BQP における解  $x$  は、長さ  $n$  で構成される各ビット  $x_i (i = 1, \dots, n)$  に 0 もしくは 1 を持つ、0-1 ビット表現である。また、0 と 1 のどのような組合せであっても実行可能解となるため、BQP の解空間  $X$  のサイズは  $2^n$  となる。最も簡単な近傍は、解  $x$  に対して、 $j$  番目のビットを反転することによる  $1$ -opt 近傍である。このとき、一度に生成可能な  $1$ -opt 近傍解の数は  $n$  に等しく、 $1$ -opt 近傍を有する局所探索法は  $1$ -opt 局所探索法 ( $1$ -opt 法) と呼ばれる。 $1$ -opt 法は、各繰返しにおいて、現在解の目的関数  $f(x)$  と  $n$  個からなる  $1$ -opt 近傍解の  $f(x')$  とのゲイン  $g (= f(x') - f(x))$  が最大となる近傍解を生成しながら探索を進め、その最大となるゲインが 0 もしくは負となることを終了条件とするものである。また、 $k$ -opt 近傍  $N_{k\text{-opt}}(x) = \{x' \in X | d_H(x, x') \leq k\}$  ( $d_H$  は二つの解のハミング距離を指す) は、一つのビットから  $k$  個のビットを一度に反転することにより到達可能な解の集合と定義され、 $k$  の数に応じて指数的 ( $|N_{k\text{-opt}}| = n^k$ ) となる。

BQP に対する  $k$ -opt 近傍は、連続的な  $1$ -opt 近傍操作によって到達可能な解集合とされる。具体的には、各ビットに対して 1 回のみ行うことを保証したビットの反転を連続的に行う。これは、最大のゲインを持つビットを連続的に反転することにより  $n$  個の解集合を生成し、その  $n$  個の解集合の中の最良解を  $k$ -opt 近傍解とするものである。よって、BQP に対する  $k$ -opt 近傍は、各繰返しにおいて  $1 \sim k$  個の可変的なビットの反転によって構成され、一般的に  $k$  の数を常に固定した完全  $k$ -opt 近傍よりも効率的な探索を可能にする。

現在解  $x$  における  $j$  番目のビット ( $x_j \in \{0, 1\}$ ) を反転し得られる近傍解評価のためのゲイン値  $g_j$  は、次式で求められ、 $O(n)$  時間を要する。なお、 $\bar{x}_j = 1 - x_j$  である。

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

式 (2) を用いて  $n$  個の  $1$ -opt 近傍解のゲイン値を計算するには、 $O(n^2)$  時間を要する。Merz らは、 $n$  個の  $1$ -opt 近傍解のすべてのゲイン値を線形時間で可能とするゲイン更新法を採用しており [18]、我々の  $k$ -opt 法でもこの更新法を取り入れた。

ここで、図 1 に示す BQP に対する  $k$ -opt 法の擬似コードをもとにアルゴリズムを説明する。このアルゴリズムは、外ループと内ループの二つの繰返し処理から構成される。各変数は、探索中に算出された最良解  $x_{best}$  を保持するための  $x_{prev}$ 、最良解  $x_{best}$  のゲイン値  $G_{max}$ 、現在解  $x$  のゲイン値  $G$ 、近傍解へ移動するために現在解に対して反転操作を実施し得るビット番号を保持する  $C$  である。内ループ中の処理は、 $C$  のすべてのビットを反転させることにより、生成された  $n$  個の解の中から最大のゲイン  $G_{max}$  を有する  $x_{best}$  を  $k$ -opt 近傍解とする。次に、外ループでは、このような探索操作を良好な近傍解が生成できなくなるまで繰返す。この  $k$ -opt 法は、deterministic であるため、与えられた初期解に依存した局所最適解を算出する。

片山らはメタ戦略に局所探索法の導入を前提として、その  $k$ -opt 法にランダム性を与えた、ランダム  $k$ -opt 法を提案している [10, 11]。文献 [9, 12] では、GLS などのメタ戦略には、ランダム  $k$ -opt 法を導入する方が良好な解を算出しやすいことが示唆されている。

#### 3.2 $k$ -opt 局所探索法のパラメータ $m$ について

BQP に対する deterministic  $k$ -opt 局所探索法の基本アルゴリズムは、内ループにおいて  $n$  個の解の生成を巧妙に行い、その解集合の中から最良の解を選ぶ。しかしながら、探索時間の効率化を考えると、常に  $n$  個の解を調べる必要はなく、実際的にはそれよりも比較的少ない個数の解を調べることで、(保証は無いが) 最良解を選び出すことが可能になる。Merz らは、内ループの終了条件 ( $C = \emptyset$  になるまで) を内ループで保持される  $x_{best}$  が見つかった時点から数えて  $m$  回の内ループの繰返しにおいても最良解が更新されなければ、内ループを強制的に終了する方法に修正し、探索時間の短縮を試みた。これにより、パラメータ  $m$  を

$m \ll n$  のように設定する場合は、基本アルゴリズムよりも探索時間の効率化が可能であることを述べている [18]. Merz らは、この  $m$  を 100 とすることを推奨した. 一方、片山らのランダム  $k$ -opt 法でも同様のパラメータが存在し、 $m$  を 50 と推奨している [10].  $k$ -opt 法は、 $m$  の設定値によって、最終的に得られる解や探索に要する探索時間に影響を及し、無駄な探索時間を省くことができると考えられる.

```

procedure k-Opt-Local-Search( $x, g$ )
1  repeat
2     $x_{prev} := x, G_{max} := 0, G := 0, C := \{1, \dots, n\};$ 
3  repeat
4    find  $j$  with  $g_j = \max_{i \in C} g_i;$ 
5     $G := G + g_j;$ 
6    if  $G > G_{max}$  then  $G_{max} := G, x_{best} := x;$ 
7     $x_j := 1 - x_j, C := C \setminus \{j\};$ 
8    update gains  $g_i$  for all  $i;$ 
9  until  $C = \emptyset;$ 
10 if  $G_{max} > 0$  then  $x := x_{best}$  else  $x := x_{prev};$ 
11 until  $G_{max} \leq 0;$ 
12 return  $x;$  end;

```

図 1  $k$ -opt 局所探索法

### 3.3 $k$ -opt 法における探索傾向の分析

本論文では、greedy に生成された初期解による解からスタートする  $k$ -opt 法に関する探索傾向を分析する. それは、将来的にメタ戦略へ  $k$ -opt 法を導入する場合の影響について検討するためであり、より実的な効果を観測できるものと期待できる [14]. これは、多くの場合に局所探索法はメタ戦略の主力探索法になり得ることに起因している. 例えば、メタ戦略の一つと知られる GLS では、突然変異や交叉操作によって生成された解は、ランダム解よりも比較的良好な解となり、greedy な初期解と類似するものと考えられることができるからである. なお、greedy 解法 [18] は、ランダム性を有しており、計算機上で与えるランダムの種などに応じて異なる近似解を生成可能である. greedy 解法により平均的に得られる解質は、以下に記述する問題例に対して、多くの場合に、既知の最良解から約 1%~5%程度と比較的良好い.

表 1 は、greedy な初期解に対する基本アルゴリズムの各外ループにおいて  $G_{max}$  を得たときの内ループの繰返し回数を示す. ここで、内ループの繰返し回数を  $ItelnLoop$  と表記する. この結果は、100 回の実行による平均である. 本実験では、BQP の良く知られたテスト問題 [4] から 500~2500 変数までの大規模な 45 個の問題例を扱う. これらの問題例は、Glover らによる 500 変数の 5 個の問題例 (glov500) [7], Kochenberger らによる 1000 変数の 10 個の問題例 (kb-g) [2], Beasley による 500, 1000, 2500 変数の問題例 (beas500, beas1000, beas2500) [5] で、それぞれは 10 個の問題例を含む. 全問題例は、ORLIB [4] より取得可能であり、これらは、問題サイズ、各問題例の行列  $Q$  の density ( $dens(Q)$ :行列  $Q$  において 0 以外の数値が含まれる割合) などの違いがある.

Merz らのパラメータ  $m$  を 100 とした方法での各外ループにおける  $ItelnLoop$  は、 $G_{max}$  が得られてから最大 100 回を必要とする. しかしながら、表 1 によると、毎回  $G_{max}$  が得られてから 100 回も内ループを行う必要はないと考えられる. 例えば、glov500-1 において  $G_{max}$  が得られたときの  $ItelnLoop$  は、1 回目の外ループでは 26, 2 回目の外ループでは 12 となっており、外ループの回数が増すごとに  $G_{max}$  を得たときの  $ItelnLoop$  は、徐々に減少することが観測できる. また、他の問題例においても同様の傾向がみられる.

これらの観測から、各外ループにおいて  $G_{max}$  が得られたときの  $l_{\text{telnLoop}}$  付近で内ループを打切れば、解質をできるだけ落とさずに計算時間を大幅に短縮できると考えられる。

表 1 を更に分析すると、glov-500 の各問題例に対する 1 回目の外ループにおいて  $G_{max}$  を得たときの  $l_{\text{telnLoop}}$  の平均値は、26~30 となっており、いずれも、問題サイズ  $n$  の約  $1/20$  である。また、2 回目以降の各外ループにおいて  $G_{max}$  を得たときの  $l_{\text{telnLoop}}$  の平均値は、前回の外ループにおける  $l_{\text{telnLoop}}$  の約  $1/2$  であることが観測できる。更に、他の各問題例群においても同様の探索傾向がみられた。このように、外ループの回数が増すにしたがって、 $G_{max}$  が得られたときの  $l_{\text{telnLoop}}$  は問題例の特徴の違いに関係なく、平均的に減少する傾向が確認された。

#### 4 探索傾向に基づいた知識の導入による $k\text{-opt}$ 局所探索法

本論文での探索傾向に基づく知識を導入した  $k\text{-opt}$  法について記述する。提案する方法は、探索時間を大幅に短縮させるために、近傍解探索の分析結果に基づき基本となる内ループの繰返し回数を決定する。そして、探索状況から更に内ループを追加することによって、近傍解生成の打ち切りを効率的に行う。我々の提案する新たな方法は、表 1 による  $k\text{-opt}$  法の探索傾向の分析から得られた知識を導入することによって、内ループの繰返し回数を効率的に変動させることを可能としている。

以下に、探索傾向に基づく知識を導入した  $k\text{-opt}$  法のアルゴリズムを説明する。この方法は、図 1 の基本アルゴリズムで示した 1-10 行目を図 2 のように書き換えたものである。ここで、図 2 に示す擬似コードをもとにアルゴリズムを説明する。各変数は、現在の内ループの繰返し回数を示す  $L$ 、 $G_{max}$  が得られたときの内ループの回数  $L_{gmax}$ 、前回の内ループでのゲイン値  $G_{prev}$ 、内ループの基本繰返し回数  $l$ 、ある区間における  $G$  の増減を調べるための  $p$  である。 $l$  の値は、表 1 の各外ループにおいて  $G_{max}$  を得たときの  $l_{\text{telnLoop}}$  に基づいて決定する。1 回目の外ループでの  $l$  は問題サイズ  $n$  の  $1/20$ 、2 回目以降の外ループの  $l$  は前回の外ループで  $G_{max}$  が得られたときの  $l_{\text{telnLoop}}$  の  $1/2$  とした。

```

1   $L := 0, L_{gmax} := 0, G_{prev} := 0, l, p;$ 
2  update  $l$ ;
3  repeat
4     $L++$ ;
5    find  $j$  with  $g_j = \max_{i \in C} g_i$ ;
6     $G := G + g_j$ ;
7    if  $G > G_{max}$  then  $G_{max} := G, x_{best} := x, L_{gmax} := L$ ;
8     $x_j := 1 - x_j, C := C \setminus \{j\}$ ;
9    update gains  $g_i$  for all  $i$ ;
10   if  $((L \geq l) \text{ and } (G_{prev} \leq G) \text{ and } (L \geq p))$  then break;
11    $G_{prev} = G$ 
12 until  $C = \emptyset$ ;
13 if  $(L_{gmax} > L - p)$  then  $p := L + p$ , goto 2;
```

図 2 探索傾向に基づいた知識の導入による  $k\text{-opt}$  局所探索法

このアルゴリズムでは、まず内ループの基本繰返し回数として  $l$  回の内ループを繰返した後、更に 10 行目の現在の  $G$  と前回の内ループで得られた  $G_{prev}$  の比較において  $G$  の増大がみられなくなるまで内ループを繰返す。13 行目では、区間  $[L - p, L]$  における広範囲な  $G$  の増減を調べ、その区間内において  $G_{max}$  が得られていれば、今後、 $G_{max}$  が更新される可能性があると考え、現時点から更に  $p$  回の内ループを繰返す。ここでは、区間を決定するための変数  $p$  の初期値を 20 とした。

次に、図 3 を用いてその様子を詳細に示す。我々の提案する方法では、内ループの基本繰返し回数  $l$  によって、内ループの繰返し回数が  $l$  回または  $l$  回から  $G$  の増大がみられなくなった時点で内ループを打ち切り、無駄な繰返しを大きく省くことができる。しかしながら、この時点から何回かの内ループを繰返すことによって  $G_{max}$  が更新されるかもしれない。そこで、現時点までの内ループにおける  $G$  の増減によって、今後、内ループにおいて  $G_{max}$  が更新される可能性を考慮した予測を行う。これによって、我々は現時点と現時点から  $p$  回前の区間  $[L-p, L]$  の内ループにおいて  $G_{max}$  が得られていれば、現時点から更に  $p$  回の内ループを繰返す方法を提案した。

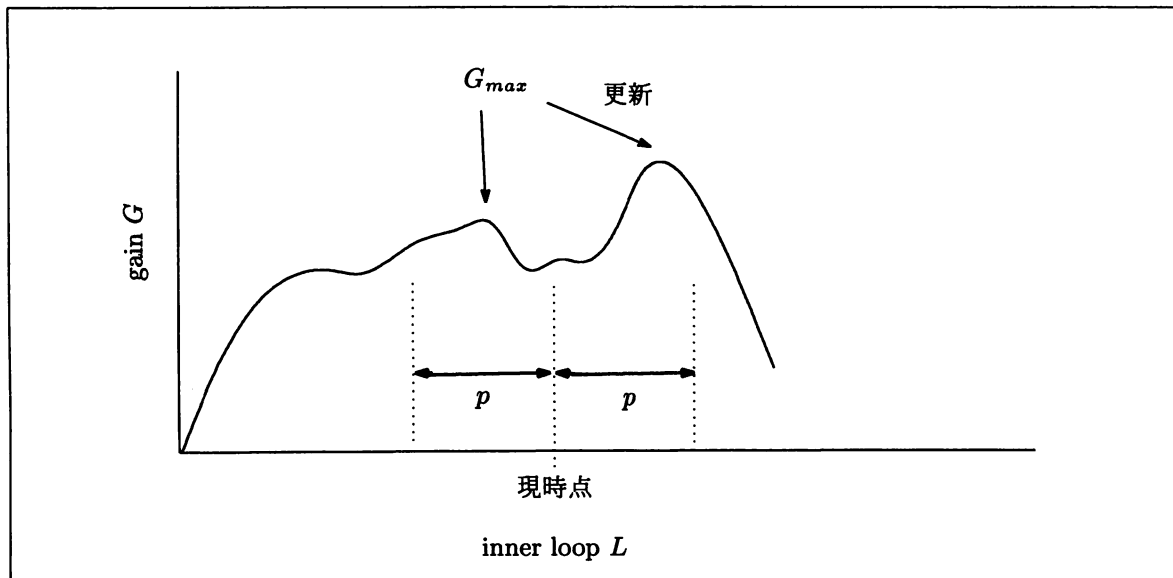


図 3  $G_{max}$  が更新される可能性を考慮した予測

## 5 数値実験

上で提案した探索傾向に基づく知識を導入した  $k$ -opt 法の有効性を検討するために、ベンチマーク問題を用いて、 $m$  を 100 に固定した Merz らの  $k$ -opt 法との比較を行う。実験は、表 1 で扱った問題例を使用する。すべての実験は、NEC PC98-NX Mate (Pentium III, 650MHz) 上で実行され、プログラム言語は C 言語である。

表 2 は、greedy な初期解に対する  $k$ -opt 法の実験結果である。各欄は、既知の最良解値 (best-known),  $m$  の値を従来通り 100 に固定、新しい方法に対して、1000 回の試行で得られた最良解 (best) と既知の最良解からのその解質 (%), 平均値 (avg) とその解質 (%), 全計算時間  $t$  (秒) を示す。更に、新しい方法の有効性を検討するために、従来法に対する平均の解質差 (loss), 全計算時間の短縮率 (r-t) を示す。

例えば、g1ov500-3 の問題例における従来の方法による avg の解質は 0.251%, 計算時間は 37 秒であるのに対し、新たな方法では、avg の解質が 0.287%, 計算時間は 3 秒となっており、0.036% の解質の低下で計算時間を 91.9% 短縮できている。また、全問題例における avg の解質の低下具合の差は従来の方法に対して、平均 0.064% の低下で、計算時間は平均 65.3% の短縮が可能である。

これらの結果から、我々の提示した方法は、最終的に得られる解質をできるだけ維持しつつ、探索時間を大幅に短縮可能であることを示した。よって、実用的な観点からメタ戦略に導入する場合にも、非常に有効であると考えられる。

## 6 結論

本論文では、 $k$ -opt 法の探索傾向の分析から得られた知識の導入による近傍解生成の打ち切りを行うことによって、大幅な探索時間の効率化手法を提案した。我々の提案する方法は、既存の効率化手法に比べ、平均的な解質をできるだけ維持しながら、探索時間を大幅に短縮できることを示した。

表 1 各外ループにおいて  $G_{max}$  を得たときの  $l_{\text{telnLoop}}$  の平均

instances	dens ( $Q$ )	outer-loop									
		1	2	3	4	5	6	7	8	9	10
glov500-1	0.1	26	12	5	2	1					
glov500-2	0.25	26	10	5	2	1					
glov500-3	0.5	30	11	5	2	1					
glov500-4	0.75	26	13	6	3	1					
glov500-5	1.0	28	15	7	4	2	1				
kb-g01	0.1	41	20	13	6	2	2				
kb-g02	0.2	60	29	15	9	4	2	1	1		
kb-g03	0.3	54	32	23	15	7	4	2	2	1	1
kb-g04	0.4	57	28	18	11	5	3	1	1		
kb-g05	0.5	47	26	15	7	3	1	1			
kb-g06	0.6	63	34	24	13	10	4	3	1	1	1
kb-g07	0.7	65	34	22	12	7	5	3	1	1	1
kb-g08	0.8	53	27	19	9	6	3	1			
kb-g09	0.9	65	36	18	9	6	4	2	1	1	
kb-g10	1.0	62	32	20	11	7	4	1			
beas500-1	0.1	27	13	6	2	1					
beas500-2	0.1	17	6	3	1						
beas500-3	0.1	22	8	4	2	1					
beas500-4	0.1	21	10	4	1						
beas500-5	0.1	24	10	7	3	1					
beas500-6	0.1	27	12	5	2	1	1				
beas500-7	0.1	25	10	6	3	1	1				
beas500-8	0.1	26	10	5	2	1					
beas500-9	0.1	26	10	6	2	1					
beas500-10	0.1	20	10	4	1						
beas1000-1	0.1	46	14	8	4	2	1				
beas1000-2	0.1	53	22	13	5	2	2	1			
beas1000-3	0.1	49	25	13	7	5	1	1	1		
beas1000-4	0.1	54	20	11	5	2	1	1			
beas1000-5	0.1	58	27	14	7	2	1				
beas1000-6	0.1	43	25	13	6	2	1	1			
beas1000-7	0.1	47	24	12	8	3	1				
beas1000-8	0.1	53	26	13	6	3	1	1			
beas1000-9	0.1	52	26	14	8	4	1	1			
beas1000-10	0.1	48	23	10	6	2					
beas2500-1	0.1	122	58	35	19	11	8	4	2	2	1
beas2500-2	0.1	106	53	29	15	9	5	2	1		
beas2500-3	0.1	123	56	30	19	10	6	3	4	1	
beas2500-4	0.1	106	47	26	16	10	4	2	1		
beas2500-5	0.1	118	56	34	18	13	6	4	2	1	1
beas2500-6	0.1	109	55	29	22	10	7	3	1	1	1
beas2500-7	0.1	114	60	38	20	14	7	3	1	1	
beas2500-8	0.1	94	43	23	12	9	6	2	2	1	1
beas2500-9	0.1	110	53	34	22	11	6	3	1	1	1
beas2500-10	0.1	110	49	28	19	10	6	3	2	1	1

表 2 greedy な初期解に対する実験結果

instances	dens ( $Q$ )	best- known	fixed $m = 100$					探索傾向に基づいた方法						
			best	(%)	avg	(%)	t(s)	best	(%)	avg	(%)	t(s)	loss(%)	r-t(%)
glov500-1	0.1	61194	61194	0.000	61059.066	0.221	6	61194	0.000	61036.039	0.258	1	0.037	83.3
glov500-2	0.25	100161	100161	0.000	99963.203	0.197	15	100161	0.000	99931.391	0.229	4	0.032	73.3
glov500-3	0.5	138035	138035	0.000	137688.484	0.251	37	138035	0.000	137638.656	0.287	3	0.036	91.9
glov500-4	0.75	172771	172771	0.000	172016.031	0.437	41	172771	0.000	171936.219	0.483	11	0.046	26.8
glov500-5	1.0	190507	190507	0.000	190017.625	0.257	60	190507	0.000	189910.859	0.313	19	0.056	68.3
kb-g01	0.1	131456	131441	0.011	130696.523	0.578	21	131441	0.011	130618.398	0.637	6	0.056	71.4
kb-g02	0.2	172788	172788	0.000	171795.234	0.575	59	172788	0.000	171621.250	0.675	18	0.100	69.5
kb-g03	0.3	192565	192534	0.016	189560.813	1.560	73	192565	0.000	189248.156	1.722	31	0.162	57.5
kb-g04	0.4	215679	215374	0.141	213336.891	1.086	108	215019	0.306	213056.625	1.216	31	0.130	71.3
kb-g05	0.5	242367	242367	0.000	240838.234	0.631	97	242367	0.000	240647.031	0.710	29	0.079	29.9
kb-g06	0.6	243293	243132	0.066	240444.313	1.171	145	243018	0.113	240016.125	1.347	32	0.176	77.9
kb-g07	0.7	253590	253561	0.011	249826.625	1.484	129	252618	0.383	249400.531	1.652	49	0.168	62.0
kb-g08	0.8	264268	264077	0.072	262086.281	0.826	150	264077	0.072	261805.281	0.932	63	0.106	58.0
kb-g09	0.9	262658	262624	0.013	259387.031	1.245	182	262624	0.013	258957.422	1.409	51	0.164	72.0
kb-g10	1.0	274375	274030	0.126	271032.438	1.218	215	274046	0.120	270615.375	1.370	75	0.152	65.1
beas500-1	0.1	116586	116586	0.000	115890.391	0.597	12	116586	0.000	115813.367	0.663	1	0.066	91.7
beas500-2	0.1	128339	128339	0.000	127963.680	0.292	9	128339	0.000	127934.039	0.316	3	0.024	66.6
beas500-3	0.1	130812	130812	0.000	130538.961	0.209	10	130812	0.000	130504.594	0.235	3	0.026	70.0
beas500-4	0.1	130097	130077	0.015	129672.227	0.327	10	130077	0.015	129619.117	0.367	4	0.040	60.0
beas500-5	0.1	125487	125487	0.000	124946.625	0.431	10	125487	0.000	124858.570	0.501	2	0.070	80.0
beas500-6	0.1	121772	121772	0.000	121175.992	0.489	15	121772	0.000	121092.656	0.558	5	0.069	66.7
beas500-7	0.1	122201	122201	0.000	121376.336	0.675	10	122201	0.000	121281.258	0.753	5	0.078	50.0
beas500-8	0.1	123559	123530	0.023	122926.336	0.512	10	123530	0.023	122853.891	0.571	4	0.059	60.0
beas500-9	0.1	120798	120798	0.000	120173.617	0.517	11	120798	0.000	120089.109	0.587	4	0.070	63.6
beas500-10	0.1	130619	130619	0.0	130259.305	0.275	14	130619	0.000	130197.305	0.323	3	0.048	78.6
beas1000-1	0.1	371438	371438	0.000	370718.688	0.194	25	371438	0.000	370635.906	0.216	5	0.022	80.0
beas1000-2	0.1	354932	354631	0.085	353504.250	0.402	33	354930	0.001	353341.531	0.448	7	0.046	78.8
beas1000-3	0.1	371236	371236	0.000	369663.281	0.424	31	371236	0.000	369465.875	0.477	10	0.053	67.7
beas1000-4	0.1	370675	370675	0.000	369705.844	0.261	21	370605	0.019	369565.313	0.299	9	0.038	57.1
beas1000-5	0.1	352760	352730	0.009	351584.000	0.333	29	352730	0.009	351434.281	0.376	10	0.043	65.5
beas1000-6	0.1	359629	359629	0.000	358467.375	0.323	37	359452	0.049	358301.969	0.369	11	0.046	70.3
beas1000-7	0.1	371193	371193	0.000	369812.781	0.372	35	371047	0.039	369642.031	0.418	15	0.046	57.1
beas1000-8	0.1	351994	351894	0.028	350711.656	0.364	32	351785	0.059	350555.531	0.409	13	0.045	59.4
beas1000-9	0.1	349337	349337	0.000	347868.656	0.420	34	349337	0.000	347679.500	0.474	14	0.054	58.8
beas1000-10	0.1	351415	351415	0.000	350170.125	0.354	26	351226	0.054	350002.469	0.402	6	0.048	76.9
beas2500-1	0.1	1515944	1515098	0.056	1510182.625	0.380	127	1515211	0.048	1509448.125	0.429	49	0.049	61.4
beas2500-2	0.1	1471392	1470386	0.068	1467646.000	0.255	110	1470386	0.068	1467242.875	0.282	45	0.027	59.1
beas2500-3	0.1	1414192	1413613	0.041	1410490.000	0.262	120	1413530	0.047	1409957.250	0.299	44	0.037	63.3
beas2500-4	0.1	1507701	1507701	0.000	1504721.375	0.198	107	1507701	0.000	1504263.500	0.228	46	0.03	57.0
beas2500-5	0.1	1491816	1491796	0.001	1488286.250	0.237	113	1491770	0.003	1487737.250	0.273	62	0.036	45.1
beas2500-6	0.1	1469162	1468319	0.057	1465333.875	0.261	117	1468293	0.059	1464696.625	0.304	48	0.043	59.0
beas2500-7	0.1	1479040	1478126	0.062	1473872.625	0.349	123	1477657	0.094	1473176.875	0.396	52	0.047	57.7
beas2500-8	0.1	1484199	1484199	0.000	1481679.250	0.170	91	1484199	0.000	1481283.625	0.196	44	0.026	51.6
beas2500-9	0.1	1482413	1482051	0.024	1479224.375	0.215	124	1481657	0.051	1478685.750	0.251	60	0.036	51.2
beas2500-10	0.1	1483355	1482544	0.055	1479378.375	0.268	113	1482462	0.060	1478833.000	0.305	48	0.037	57.5

## 参考文献

- [1] B. Alidaee, G.A. Kochenberger, A. Ahmadian, "0-1 quadratic programming approach for optimal solution of two scheduling approach for the optimal solution of two scheduling problems," *International Journal of Systems Science*, vol.25, no.2, pp.401-408, 1994.
- [2] M.M. Amini and B. Alidaee, and G.A. Kochenberger, "A scatter search approach to unconstrained quadratic binary programs," *New Ideas in Optimization* (eds, D. Corne, M. Dorigo and F. Glover), McGraw-Hill, pp.317-329, 1999.
- [3] F. Barahona, M. Jünger, G. Reinelt, "Experiments in quadratic 0-1 programming," *Mathematical Programming*, vol.44, pp.127-137, 1989.
- [4] J.E. Beasley, "OR-Library: distributing test problems by electronic mail," *Journal of the Operational Research Society*, vol.41, no.11, pp.1069-1072, 1990.
- [5] J.E. Beasley, "Heuristic algorithms for the unconstrained binary quadratic programming problem," Technical Report, Management School, Imperial College, UK, 1998.
- [6] G. Gallo, P.L. Hammer, B. Simeone, "Quadratic knapsack programs," *Mathematical Programming* vol.12, pp.132-149, 1980.
- [7] F. Glover, G.A. Kochenberger, and B. Alidaee, "Adaptive memory tabu search for binary quadratic programs," *Management Science*, vol.44, no.3, pp.336-345, 1998.
- [8] P.L. Ivănescu, "Some network flow problems solved with pseudo-Boolean programming," *Operations Research*, vol.13, pp.388-399, 1965.
- [9] K. Katayama, M. Tani, and H. Narihisa, "A comparison of genetic local search algorithms for binary quadratic programming," Technical Report of IEICE (COMP), COMP2000-12, vol.100, no.52, pp.49-56, May 2000.
- [10] 片山謙吾, 成久洋之, "バイナリー 2 次計画問題に対する変形  $k$ -opt 局所探索法," *信学誌 (A)*, vol.J84-A, no.3, pp.430-435, Mar. 2001.
- [11] K. Katayama, M. Tani, and H. Narihisa, "Solving large binary quadratic programming problems by effective genetic local search algorithm," *Proc. of the 2000 Genetic and Evolutionary Computation Conference*, Jul.8-12, Las Vegas, USA, pp.643-650, 2000.
- [12] K. Katayama and H. Narihisa, "On fundamental design of parthenogenetic algorithm for the binary quadratic programming problem,"
- [13] B.W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell System Technical Journal*, vol.49, pp.291-307, 1970.
- [14] K. Kohmoto, K. Katayama, and H. Narihisa, "Empirical Knowledge of a Parameter Setting in  $k$ -opt Local Search for the Binary Quadratic Programming Problem," *Proc. of the 2001 Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01)*, Aug 4-10, Seattle, USA, 2001.
- [15] J. Krarup, P.M. Pruzan, "Computer aided layout design," *Mathematical Programming Study*, vol.9, pp.75-94, 1978.
- [16] S. Lin and B.W. Kernighan, "An effective heuristic algorithm for the traveling salesman problem," *Operations Research*, vol.21, pp.498-516, 1973.
- [17] R.D. McBride, J.S. Yormark, "An implicit enumeration algorithm for quadratic integer programming," *Management Science*, vol.26, no.3, pp.282-296, 1980.
- [18] P. Merz and B. Freisleben, "Greedy and local search heuristics for unconstrained binary quadratic programming," Technical Report No.99-01, Informatik-Berichte, 1999. (to appear in *Journal of Heuristics*)
- [19] P.M. Pardalos, G.P. Rodgers, "A branch and bound algorithm for the maximum clique problem," *Computers and Operations Research*, vol.19, no.5, pp.363-375, 1992.
- [20] P.M. Pardalos, J. Xue, "The maximum clique problem," *Journal of Global Optimization*, vol.4, pp.301-328, 1994.
- [21] A.T. Phillips, J.B. Rosen, "A quadratic assignment formulation of the molecular conformation problem," *Journal of Global Optimization*, vol.4, pp.229-241, 1994.



$k$ -opt Local Search by Introduction of Knowledge based on Search Tendency

Keiko KOHMOTO, Kengo KATAYAMA\* &amp; Hiroyuki NARIHISA\*

*Graduate School of Engineering**\*Department of Information & Computer Engineering**Faculty of Engineering**Okayama University of Science**Ridaicho 1-1, Okayama 700-0005, Japan*

(Received November 1, 2001)

In recent years, the metaheuristics which are algorithm, such as Genetic Algorithm (GA), Simulated Annealing, Tabu Search, and so on, has been proposed as an efficient method for optimization problem. The metaheuristics can calculate near-optimal solutions of high-quality in a short search time relatively by combining Local Search (LS) with is the basic method, and the idea which exists in the natural world. As this example, although GLS which included LS in GA is mentioned, LS accounts for many of search time expended by GLS in many cases. Therefore, maintaining the performance of LS used by the metaheuristics, the proposal of the method to enable the efficiency enhancement of search time is expected.

In this paper, we focus on the most powerful the  $k$ -opt local search heuristic in LS known for the binary quadratic programming problem. The  $k$ -opt local search heuristic is known that it can reduce search time to some extent. However, the introduction of sufficient knowledge for the  $k$ -opt local search heuristic is not studied. We propose the efficiency enhancement technique to reduce all the more search time by introducing the knowledge based on the search tendency by analysis of the near solution search of the  $k$ -opt local search heuristic. In comparison with the existent the efficiency enhancement technique, our proposal method shows that search time can be more reduced, maintaining an equivalent search performance.