A-030

2 次割当問題に対する遺伝的反復貪欲法

Genetic Iterated Greedy with k-swap Local Search for QAP

長谷部 晶久[1]　外山 史[1]　東海林 健二[1]　宮道 壽一[1]

Akihisa Hasebe　Fubito Toyama　Kenji Shoji　Juichi Miyamichi

1 まえがき

代表的な組み合わせ最適化の問題の一つとして 2 次割当問題 (Quadratic Assignment Problem，QAP) がある．QAP は LSI のレイアウト設計やノード配置問題などの多くの応用例がる問題として知られている．QAP は NP 困難な問題であるため，問題の規模が大きくなると厳密な最適解を求めることは極めて困難である．したがって，これまでに数多くのメタ戦略アルゴリズムが提案されている．中でも，遺伝的反復局所探索法 (Genetic Iterated Local Search，GILS)[1] が QAP に対して優れた探索性能を示している．GILS は，局所探索として探索能力に優れた k-swap 局所探索法を繰り返し行う反復局所探索法 (ILS) に交叉操作を導入した手法である．ILS では局所最適解からの脱出操作としてランダムベースで値を入れ替える Kick と呼ばれる処理が用いられている．

本手法では，k-swap 局所探索法を用いるが，局所最適解からの脱出操作として貪欲法による解の再構成 (Kick) を用いた反復貪欲法を遺伝的局所探索法の局所探索として組み入れた手法 (Genetic Iterated Greedy with k-swap Local Search，GIGLS) を提案する．提案手法で用いる反復貪欲法は，解の再構成に貪欲法を用いるため，ランダムベースに行うよりも比較的良好な解を探索することができるという特徴を持っている．実験では，QAP に対する強力なメタ戦略アルゴリズムである GILS および Robust Taboo Search(RTS)[2] と比較し，探索性能の評価を行った．

2 二次割当問題

QAP は，$n \times n$ の距離行列 $D = [d_{ij}]$ とフロー行列 $F = [f_{kl}]$ が与えられたとき，式 (1) の目的関数を最小化する n 個の要素 $\{1, 2, \cdots, n\}$ からなる順列 π を求める問題である．

$$C(\pi) = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} f_{\pi(i)\pi(j)} \tag{1}$$

3 QAP に対する遺伝的反復貪欲法

QAP に対する遺伝的反復貪欲法 (GIGLS) の処理の流れを図 1 に示す．GIGLS では，まず P_{num} 個の初期集団 π_i をランダムに生成し，反復貪欲法を適用する (Line2-5)．次に，集団の中から親個体としてランダムに 2 つの個体 π_{p1},π_{p2} を選ぶ (Line9)．ここで，選ばれた 2 個体の解の要素の共通部分が 80% 未満であれば交叉を行い，そうでなければそれぞれの解に対して突然変異 (解の再構成) を行う．交叉を行う場合には，2 つの親個体に対して巡回交叉 (CX)[3] を適用し，新たに生成された 2 つの子個体 π_{c1},π_{c2} に対して反復貪欲法を適用する．そして，親個体と子個体の 4 個体の中から適応度の高い 2 個体が次の世代の集団に選択され，その他の個体は淘汰される (Line11-15)．突然変異を行う場合には，突然変異により再構成されたそれぞれの個体を，親個体と入れ替えることにより次世代の集団を生成する (Line16-17)．交叉または突然変異が行われた後，次の世代の集団に選択された解の評価値が，現集団中の最良解 π_{best}，探索中全体で発見された最良解 π_{all_best} よりも良ければ，それぞれ解の更新を行う (Line21-30)．また，本手法では解の探索に停滞が生じた場合，現在探索中の解空間にはより良い評価値の解はないものとして再スタート処理を行う．再スタート処理は，新たに初期集団を生成することで解の再探索操作を行う処理である．再スタート処理は，10 世代の間に解の更新がなければ実行される (Line32-38)．以上の処理 (Line6-39) を終了条件が満たされるまで繰り返す．

```
      procedure GeneticIteratedGreedy
      begin
 1:     π_all_best := ∞,π_best := ∞;
 2:     for i := 1 to P_num do
 3:       generate a random solution π_i;
 4:       π_i := IteratedGreedy(π_i);
 5:     endfor
 6:     while (termination criterion not satisfied) do
 7:       P := {1, 2, ··· , P_num};
 8:       repeat
 9:         select parents p1, p2 from P;
10:         P := P\{p1, p2};
11:         if (crossover criterion satisfied)
12:           then π_c1, π_c2 := Crossover(π_p1, π_p2);
13:                π_c1 := IteratedGreedy(π_c1);
14:                π_c2 := IteratedGreedy(π_c2);
15:                select solutions π_np1, π_np2
                                 from π_p1, π_p2, π_c1, π_c2;
16:           else π_np1 := Reconstruction(π_p1);
17:                π_np2 := Reconstruction(π_p2);
18:         endif
19:         π_p1 := π_np1;
20:         π_p2 := π_np2;
21:         if C(π_p1) < C(π_p2)
22:           then π_temp := π_p1;
23:           else π_temp := π_p2;
24:         endif
25:         if C(π_temp) < C(π_best)
26:           then π_best := π_temp;
27:         endif
28:         if C(π_best) < C(π_all_best)
29:           then π_all_best := π_best;
30:         endif
31:       until P = φ
32:       if (restart criterion satisfied)
33:         for i := 1 to P_num do
34:           generate a random solution π_i;
35:           π_i := IteratedGreedy(π_i);
36:         endfor
37:         π_best := ∞;
38:       endif
39:     endwhile
40:     return π_all_best;
      end;
```

図 1: QAP に対する遺伝的反復貪欲法

3.1 反復貪欲法

反復貪欲法とは，貪欲法を用いた解の再構成処理を繰り返し行う手法である．一般的に反復貪欲法は局所探索法と組み合わせて用いられ，局所最適解からの脱出操作として解の再構成処理が使用される．解の再構成処理については 3.2 節で詳しく述べる．本手法では，反復貪欲法における局所探索として k-swap 局所探索法 (KLS) [1] を用いている.KLS とは，与えられた解に対して解の二つの要素を交換して近傍解を得

[1]宇都宮大学大学院工学研究科

る 2-swap 近傍操作を連鎖的に適用することで得られる解集合 $\{\pi^1, \pi^2, \cdots, \pi^{\frac{n}{2}}\}$ の中から最も評価値の高い近傍解 π^k に遷移していく局所探索法である．

反復貪欲法の擬似コードを図 2 に示す．反復貪欲法は，まず与えられた解 π に対して k-swap 局所探索を適用し，得られた局所最適解を π_{best} に保持する (Line1-2)．次に，解 π に対して解の再構成を行い，k-swap 局所探索を適用する (Line4-5)．得られた解 π が反復貪欲法による探索中に発見された最良解より評価値が良ければ解の更新を行う (Line6-8)．この解の再構成と局所探索の処理を終了条件が満たされるまで繰り返す (Line3-9)．実験での終了条件は，この解の再構成と局所探索を行う繰り返し回数が，初期集団の生成後に適用される場合は 20 回以内，交叉後に適用される場合は 5 回以内に解の更新が行われなかったときとした．

```
     procedure IteratedGreedy(π)
     begin
 1:    π := LocalSearch(π);
 2:    π_best := π;
 3:    while (termination criterion not satisfied) do
 4:      π := Reconstruction(π);
 5:      π := LocalSearch(π);
 6:      if C(π) < C(π_best)
 7:        then π_best := π;
 8:      endif
 9:    endwhile
10:    return π_best;
     end
```

図 2: 反復貪欲法

3.2 解の再構成

解の再構成は，局所探索などによって得られた解を破壊する処理 (Destruction) と，破壊された解から貪欲法を用いて再び解を生成する処理 (Construction) からなる．貪欲法は，解を生成する各計算ステップで最も評価値の高い部分解を選択し，これを繰り返し行うことで解を生成する手法である．本手法は，全体に対するフロー量の大きい解の要素を，全体に対する距離が小さい解の位置に配置していくという考え方で解の再構成を行っていく．

解の再構成の流れを図 3 に示す．まず，Destruction では解 π からランダムに選択された r 番目の解の要素 $\pi(r)$ を $kick\_size$ 個取り除く．取り除かれた解の要素 $\pi(r)$ は P_{item} に，解の要素の位置 r は P_{place} に保持する (Line1-6)．次に Construction では，取り除かれた解の要素の位置 $a \in P_{place}$ の中で全体に対する距離 $d_{sum}(a) = \sum_{k=1}^{n}(d_{ak} + d_{ka})$ の最も小さい解の要素の位置 $place$ に，取り除かれた解の要素 $b \in P_{item}$ の中で全体に対するフロー量 $f_{sum}(b) = \sum_{k=1}^{n}(f_{bk} + f_{kb})$ の最も大きい解の要素 $item$ を配置する (Line8-10)．選択された $item$, $place$ はそれぞれ P_{item}，P_{place} から取り除き，再度選択されないものとする．この処理を $kick\_size$ 回繰り返し，解を生成していく (Line7-12)．

4 実験

提案手法の有効性を示すため，提案手法 GIGLS と提案手法の解の再構成をランダムベースに行った GIRLS，および強力なメタ戦略アルゴリズムである GILS，RTS の比較実験を行った．GILS については，文献 [1] に基づいてコード化したものを用いた．また RTS についてはÉ.D.Taillard によって公開されているソースプログラム[2] をそのまま使用した．対象とした問題は QAP のベンチマーク問題例集である QAPLIB[3] より大規模な問題 9 例題を選択した．実験環境はすべて同じ計算機上 (CPU:Xeon E5420 2.50GHz，メモリ:48GB，コンパイラ:gcc Ver.4.1.2) で行った．実験に用いた提案手法 GIGLS と GIRLS のパラメータは，集団の数 P_{num} は 8，解の再構成のサイズ $kick\_size$ は問題サイズ n の 20%とした．また，すべての手法において終了条件は，既知の最適解が求まるか，20 分が経過するまでとした．

```
     Procedure Reconstruction(π)
     begin
 1:    P_item := φ, P_place := φ, P := {1, 2, ··· , n};
 2:    for i := 1 to kick_size do
 3:      r := select an element randomly from P;
 4:      P_item := P_item ∪ {π(r)}, P_place := P_place ∪ {r};
 5:      P := P\{r};
 6:    endfor
 7:    for i := 1 to kick_size do
 8:      find item ∈ P_item with max(f_sum(item));
 9:      find place ∈ P_place with min(d_sum(place));
10:      π(place) := item;
11:      P_item := P_item\{item}, P_place := P_place\{place};
12:    endfor
13:    return π;
     end
```

図 3: 解の再構成処理

各問題に対する試行回数を 20 回として比較実験を行った結果を表 1 に示す．表 1 には対象とした各問題名と各試行で得られた最良解の平均精度 (%) を示す．問題名中の数値はその問題サイズ n を表す．なお，表内の太字値は各問題において最も良好な結果を示している．最良解の精度は以下の式を用いて算出した．

$$\text{精度 (\%)} := \frac{\text{各試行で得られた最良解の平均値} - \text{既知の最適解値}}{\text{既知の最適解値}} \times 100$$

表 1 より，すべての問題に対して提案手法が従来手法よりも良い結果が得られた．特に解の再構成に貪欲法を用いた提案手法 GIGLS は tai80a や tai100a のようなランダムに生成された，構造化されていない問題に対して有効であることを確認した．

表 1: 実験結果

問題名	GIGLS	GIRLS	GILS	RTS
tai80a	**0.788**	0.813	1.619	0.891
tai100a	**0.820**	0.991	1.719	0.880
tai80b	**0.000**	**0.000**	0.021	0.076
tai100b	**0.000**	**0.000**	0.049	0.082
tai150b	0.196	**0.171**	0.315	0.383
tai256c	0.122	**0.103**	0.194	0.407
will100	0.090	**0.083**	0.210	0.100
sko100a	**0.005**	0.006	0.050	0.024
tho150	0.028	**0.022**	0.093	0.061
平均	**0.228**	0.243	0.474	0.323

5 おわりに

本論文では，QAP に対して遺伝的反復貪欲法を提案した．実験では，従来手法の中で優れた探索性能を持つ GILS と RTS と比較を行うことで，提案手法の有効性を確認した．

参考文献

[1] 北田雅亨，片山謙吾，南原英生， 成久洋之:“2 次割当問題に対する遺伝的反復局所探索法，” 第 3 回進化計算フロンティア研究会， pp.81-89， 2010.

[2] É.D.Taillard, “Robust taboo search for the quadratic assignment problem, ” Parallel Computing, vol.17, pp.443-455, 1991.

[3] P.Merz and B.Freisleben, “Fitness Landscape Analysis and Memetic Algorithms for the Quadratic Assignment Problem, ”, IEEE Transactions on Evolutionary Computation, Vol.4, No.4, pp337-352, 2000.

[2] http://mistic.heig-vd.ch/taillard/codes.dir/tabou_qap2.c

[3] http://www.opt.math.tu-graz.ac.at/qaplib/inst.html