

## 最短道順の実時間探索ための 複数方策メタ戦略融合 GA 方式

櫻井義尚<sup>†</sup> 塚本奈津貴<sup>†</sup> 高田考平<sup>†</sup> 鶴田節夫<sup>†</sup>

物流ネットワークや配送ルートの最適化システムでは、30~数百拠点（都市）の最短道順探索問題（巡回セールスマン問題：TSP）を対話的応答時間（3秒以下）かつ誤差3%以下で解かなければならない。これらの要求を満たすために、複数の方策を遺伝的アルゴリズムGAに組込む方法を研究してきた。まず、各々異なる方策（発見的知識）を1つ組込んだ複数のGAを、ランダムに初期化し直すメタ戦略（ランダムリスタート法）により順次実行することを指定時間だけ反復し、各GAの解のうち最適解を実行可能解あるいは近似最適解として選んだ。次に、各世代内に複数方策をまとめて組込んだGAを指定時間まで実行し、実行可能解（近似最適解）を得る方式を提案した。これらは同種目的の他方式よりもシンプルで高性能だが、規模が250都市以上では最悪誤差が4%を越えた。この解決のため、複数方策やメタ戦略をGAへ融合する改良方式を提案する。これは、1つのGAの指定世代において、メタ戦略としてのランダムリスタートを評価の悪いまたは同種の（ここでは評価が同じ）子個体群に適用し上記の発見的知識を用い局所最適化した後に世代の最終評価をして集団の多様性を保つ方式である。世界一高速な厳密解法の1つであるConcordeでも10秒から100秒かかる200~600都市前後のTSPLIBの問題を本方式で1000回試行した結果、最悪ケースでも3%以下の誤差の解が3秒以内で探索可能なことを実験で確認した。

### Inner Random Restart Genetic Algorithm to Optimize Delivery Schedule

YOSHITAKA SAKURAI<sup>†</sup> NATSUKI TSUKAMOTO<sup>†</sup>  
KOUHEI TAKADA<sup>†</sup> SETSUO TSURUTA<sup>†</sup>

A delivery route optimization that improves the efficiency of real time delivery or a distribution network requires to solve several tens to hundreds cities Traveling Salesman Problems (TSP) within interactive response time, with expert-level accuracy (less than about 3% of error rate). To meet these requirements, an Inner Random Restart Genetic Algorithm (Irr-GA) method is developed. This method combines random restart and GA that have several types of heuristics. This method uses a different type of heuristics such as a 2-opt type mutation and a block (Nearest Insertion) type mutation. Comparison based on the results of experiments proved that the method is superior to others and our previously proposed method.

## 1. 序論

小包や手紙の配送から複数の企業がからむ製品供給/分配のための配送まで、最適化を要する多くの物流ネットワークがある。これらの効率を高めるためには、配送ルートつまり多数の配送先への配送順序を最適化することが必要である。1回の配送では、数十箇所~数百箇所以上の異なった場所を回る。したがって、このような配達ルートの最適化は、中大規模巡回セールスマン問題（Traveling Salesman Problems, TSP）[1][2]としてモデル化できる。TSPは計算複雑性理論において、NP困難と呼ばれる問題のクラスに属する組み合わせ最適化問題である。n-都市のTSPを解くためにはn!オーダーの組み合わせを処理する必要がある、計算量の爆発を引き起こす。一方、実際は、定式化が困難な社会的な条件がからむため、現場での実用には、求解結果を人間が確認する必要がある。すなわち、近似解が実用できるか人間のユーザがチェックし、時には手動修正したり、代替案を選択したりしなくてはならない。つまり実用上、TSPの解法には人間の介入をスムーズに行うための対話（リアルタイム）応答性が要求される。

さて解の精度については、領域の専門家によって生成された近似解は数学上の最適解と比較すると3%程度の誤差はあるかもしれないが、これ以上に精度の悪い解を出して、実用的な問題を引き起こすようなことはない。つまり、厳密解が必要なわけではないが従来の近似TSP解法[1][2]のように、数学的に最適解を出すことがあっても、確実に3%程度以下の誤差の範囲内での解を保証しないのは問題となる。実際の利用現場では、一度でも誤差の大きい近似解を出力するとユーザからの信頼を失い、物流ネットワークの現場では実用され難い。そこで、インタラクティブつまり対話リアルタイム時間内にTSPを最適化するだけでなく、様々なパターンの問題に対して大きな誤差を出さないことを目的に、複数のヒューリスティクスを組み込んだ幾つかのタイプの遺伝的アルゴリズム（Genetic Algorithm, GA）を提案してきた。例えば、2-opt型GAは一部の問題パターンでは解の精度が良いが、別のパターンでは局所最適解に陥る傾向があるので、最近挿入法（Nearest Insertion, NI）を使ったブロック型GAをカスケードさせることにより2-opt型GAの短所を補う外部多世界GA（Mow-GA, Multi-outer-world GA）方式[3]を提案した。しかしながら、Mow-GAは1つのGAが終わってから次のGAを実行する。このため、前のGAで生成された優秀な個体やその良い形質がリセットされてしまい子孫に遺伝しない。つまり、ランダムな値をもつ初期個体から再スタート（つまりランダムリスタート）する。また、実行時間が短い場合、各GAにおいて十分な世代数を確保出来ない場合がある。さらに、各ヒューリスティクスはあくまで独立に探索を進めるため、探索過程における各ヒューリスティクスの相補作用は望めない[4]。そこで、次に、相互に弱点

<sup>†</sup> 東京電機大学 情報環境学部 情報環境学科  
Department of Information Environment, School of Information Environment, Tokyo Denki University

を補う関係にある複数のヒューリスティクス「世界」を一つの GA の「内部」に持つことにより、ヒューリスティクスの相補作用による探索効率を高めた内部多世界 GA (Multi-inner-world Genetic Algorithm, Miw-GA) [5]を提案した。これは、1 世代内に 2 つのヒューリスティクス、1 つは 2-opt 型の突然変異、もう一方はブロック (最近挿入つまり NI) 型突然変異を実行するものであった。この方法はさまざまな問題パターンに対し、対話時間内に高精度な解が出せたが、約 200 箇所(都市)を超える規模の問題 (TSP) になると、解の精度が 4% を超える場合が多くなった。

本論文ではこれを解決するために、ランダムリスタートを GA の指定世代において特定の子個体群に適用した後、上記 2 つの発見的知識・方策を用いた局所最適化や淘汰条件の緩和などの保護育成により集団の多様性を保つ方式を提案する。これにより、世界最高速レベルの TSP 解法といわれる Concorde (ブランチ&カットを用いた厳密解法[19])でさえ数十秒から数百秒かかる 200-500 都市前後の TSP でも、最悪 3% の誤差で 3 秒以内に求解可能となる。

以下、第 2 節では、配送ルート最適化問題とその技術課題を説明する。第 3 節で、問題を解決するための方法を提案する。次に、第 4 節で、その効果を検証するため他の方法と比較する。第 5 節では、実験と結果を示し、実験結果に基づいて解法の有効性を実証する。最後に第 6 節では、結果を結論づける。

## 2. 配送経路最適化における問題

本節では、まず、実際の物流ネットワークについて説明する。次に、これらの物流ネットワークの最適化問題を定式化する。

### 2.1 配送ルート最適化問題

複数企業が絡む製品供給/分配用の大規模物流ネットワークの最適化には、デポと呼ぶ配送基地の場所・コストも含む各種条件下で、配送ルートのシミュレーションを繰返し、総配送費用・時間を把握する必要がある。これらの結果を全体的に評価するのに、人間の判断が不可欠なため、システムは個々の配送ルートに対話リアルタイム時間内に求解する必要がある。

また、小包や手紙の配達/配送のためルート最適化問題では、オートバイや軽トラックなどの小型車両で 1 日に 1-3 回、配送が行われる。1 台の車両で配達される区域は、地域によって異なる。配送先は、都市部では比較的過密だが、農村部では散らばっている。したがって、配送先の数は、地域と時間帯によって、数十から数百以上 (最大 2000 未満) と異なる。配送先が毎回変化するため、配送毎にそのルートを最適化する必要があるものも多い。

これらは、人間的・社会的要因が絡むが、基本的には時間あるいは距離に関して最短

の配送ルートを探す問題で、有名な「中国人郵便配達問題」や「巡回セールスマン問題(TSP)」としてモデル化される。厳密解でなくても専門家レベルの精度を確保出来れば、近似最適解法によるコンピュータ支援は、これら小包/手紙の配送をはじめとする物流ネットワークにおける自動車排気ガスや労働者の負担を軽減するのにかなり有望である。

### 2.2 問題の定式化

これらの物流ネットワークの配送ルート最適化問題を定式化すると以下ようになる: 配送網は重み付き完全グラフ  $G=(V,E,w)$  で表現する。ここで  $V$  はノード(頂点)集合であり、ノード(頂点)  $v_i$  ( $i=1,\dots,N$ ) は巡回する配送元や配達先つまり巡回拠点あるいは都市を表す。  $N$  はノード(頂点)数である。  $E$  は枝集合であり、枝  $e_{ij}$  はノード  $v_i$  とノード  $v_j$  を結ぶルートを表す。  $w$  は枝の重み集合であり、枝の重み  $d_{ij}$  はノード  $v_i$  とノード  $v_j$  の 2 地点間距離である。このようなグラフ  $G=(V,E,w)$  上で最も短い巡回路長をもつハミルトン閉路を求める問題を Traveling Salesman Problem(TSP), このような閉路を最適解と呼ぶ。

このように物流ネットワークを TSP として定式化する。この配送効率を高めるのに、対話リアルタイム時間 (数ミリ秒, 大規模物流ネットワークでは数十から約数百ミリ秒) 内に TSP の近似解を、専門家レベルの精度 (最適解からの誤差が 3% 程度以下) で常に得ることが要求される。この分野の専門家は問題パターンにかかわらず、実用的でない、すなわち 3% 程度以上の誤差を持つ解を出さないためである。次節で、この問題を解決する知的な近似解法を提案する。

## 3. 複数方策活用内部ランダムリスタート GA (Irr-GA)

上述のように配送ルート最適化問題は、TSP として定型化し得る。本論文では、他手法との比較のため、TSP ベンチマークとして有名な TSPLIB の対称 (無向性) ユークリッド TSP (Symmetric traveling salesman problem) [1][2]を用いて評価を行う。但し、本提案手法の利用が対称 TSP に限定されるわけではない。実際の配送路では一方通行などにより非対称 TSP となるが、もちろんこれにも適用可能である。

### 3.1 提案手法のコンセプト

本論文では、様々な種類の配送先パターンに対して、対話リアルタイム時間 (3 秒) 以内に専門家レベルの精度 (最悪誤差 3%) の解を保証するために、複数方策を用いる GA とメタ戦略の一つであるランダムリスタート戦略を融合した複数方策活用内部ランダムリスタート GA (Inner Random Restart Genetic Algorithm, Irr-GA) を提案する。

Irr-GA は、1 つの GA の中に、メタ戦略としてのランダムリスタート方式と複数種類の発見的知識 (ヒューリスティクス: 本論文では、後述の最近挿入法: NI と 2-opt)

を持つ。GA による解探索は、個体と呼ばれる解候補の複数生成とそれらの評価・選択（淘汰）による生き残り解候補選択を1世代の処理として、これを数世代繰り返すことにより行われる。GA による解探索を効率的に進めるためには、比較的優良かつ類似性の低い個体を生成することと、生成した個体集団の多様性の維持が重要になる。

Irr-GA では、まず交叉や突然変異などで最近挿入法 (NI: Nearest Insertion method, や 2-opt などの発見の方策を複数融合し、親個体の遺伝子配列情報を部分的に継承しているが一部は異なる比較的優秀な子個体を作り、それをさらに局所改良する。こうして生成された子個体のうち同じ評価値を持つなど類似した個体群、また指定世代においては評価値が平均より悪い個体群をランダムに初期化し直すランダムリスタートと呼ばれるメタ戦略を適用する。次に、上記を含めた発見的知識つまり方策を適用し、これらの新生個体も生存できる様にしながら世代としての最終評価を行い、親個体も含めた中から集団サイズ分の個体を選択（淘汰）する。

こうして、世代の途中に全く新規の個体群を生成するとともに、複数の発見的知識の適用によって、様々な方向から局所的改良を行う。あるいは、それらがすぐに淘汰されないように保護・育成の期間や手段を講じて、集団の多様性を保ちながら局所収束をできるだけ避け求解除する方式である。

以下の節では、Irr-GA に用いられる個体生成から評価・淘汰に至る詳細アルゴリズムについてそれぞれ説明する。

### 3.2 個体の生成

本提案の GA では、各個体（染色体）の各遺伝子は TSP におけるノード番号（配送先の識別番号）を表す。従って染色体はその並びが図 1 に示すような巡回順序すなわち、配送ルートを表すノード列である。

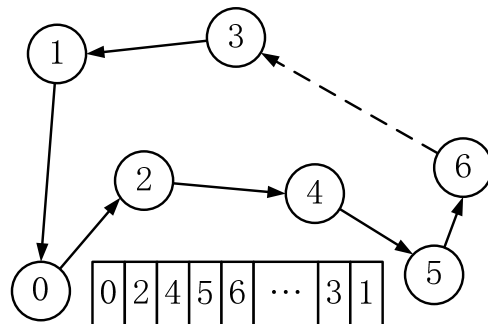


図 1 配送ルートの遺伝子型

Figure 1 Chromosome representing Delivery Route.

このような個体の生成には、ゼロから生成する始祖（初期）個体生成、親個体から子個体を生成する突然変異や交叉などの改良手法がある。一般に、大規模問題には、交叉オペレータが有効であるが、高い計算コストを必要とする。今回提案する GA では、数十から数百拠点の TSP の近似解を対話的な応答時間内に得ることを目的とするので、交叉オペレータは使用しない。

#### 3.2.1 異種方策の交互利用による局所収束の防止

GA の初期段階では、ランダムに解の候補になる個体（集団）のグループ（初期個体集団）が作られる。但し、この場合も、同じ点を 2 回以上通るなど明らかに効率の悪いルート（致死遺伝子）が生成されないように、例えば最近挿入法 (Nearest Insertion method, NI) を使用して始祖個体を生成するランダム NI 生成を用いる。また、その結果に対し、本例では 2-opt を適用し局所最適化した後、初期の親個体とする。こうして、対話リアルタイムな応答性能を目指し、最適化効率を高める。

各世代での新子個体生成においても、初期個体生成と同様に NI を用いたブロック (NI) 型突然変異と 2-opt 型突然変異をカスケードに適用する。前者では個体内の連続した一部の遺伝子列をランダムに取り出し最近挿入法 (NI) 操作を行うことにより子個体の多様性増大を狙う。また、後者では、前者の適用により最適度が低下するのでこれを局所最適化して改良し、淘汰される率を減らす。

#### (1) ランダム NI による始祖個体生成

局所最適解への収束を避けるには、初期個体のランダム性が重要である。しかし、完全にランダムでは、収束速度は遅い。したがって、最近挿入法 (NI) にランダム性を加えたランダム NI 方式を工夫した。即ち、ランダムな順序にノードを並べ、その順に NI(最近挿入)法[2]を使用して、それらをつター（最初は空の部分ツアー）に挿入して個体としてのツアーを構成する。ツアー  $\{x_1, x_2, \dots, x_n\}$  にノードを挿入する NI 法の擬似コードを以下に示す。

begin

$L_{\text{before}} := (x_j \text{ から } x_2 \text{ に行く時, } x_{\text{new}} \text{ を経由する時と, しない時の距離の増分})$

length of tour  $\{x_1, x_{\text{new}}, x_2, \dots, x_n\}$ ;

for (j = 2, 3, ..., n) do

remove node  $x_{\text{new}}$  from tour;

insert node  $x_{\text{new}}$  between node  $x_j$  and  $x_{j+1}$  of tour;

$L_{\text{after}} := (x_j \text{ から } x_{j+1} \text{ に行く時, } x_{\text{new}} \text{ を経由する時の距離の増分})$

if ( $L_{\text{after}} < L_{\text{before}}$ )

then  $L_{\text{before}} := L_{\text{after}}$ ; Insertion location  $i = j$ ;

insert node  $x_{\text{new}}$  between  $x_i$  and  $x_{i+1}$  of sub-tour;

end

## (2) 2-opt 型突然変異

この方法は、2-opt[2]のような単純な局所探索ヒューリスティック法と GA の突然変異オペレータを結合することによって収束速度の改良をはかる。これは以下のステップから成る。

- ① 親個体  $tour_{par} = \{x_1, x_2, \dots, x_n\}$  を選択する。  $tour_{chi}$  は子個体とする。
- ②  $tour_{par}$  からノード  $x_i$  をランダムに選択する。
- ③ 1 順目の場合、  $tour_{par}$  から  $\{x_i, x_{i+1}\}$  を除いた別のノード  $x_j$  をランダムに選択する。 2 順目以降の場合、  $tour_{par}$  から  $\{x_i, x_{i+1}\}$  と③において既に選択されたノードを除いた別のノード  $x_j$  をランダムに選択する。全てのノードが既に選択された場合は終了する。
- ④  $tour_{par} \{x_1, \dots, x_i, x_{i+1}, \dots, x_j, x_{j+1}, \dots, x_n\}$  の部分ツアー  $\{x_{i+1}, \dots, x_j\}$  を反転させることにより、  $tour_{chi} \{x_1, \dots, x_i, x_j, x_{j+1}, \dots, x_{i+2}, x_{i+1}, x_{j+1}, \dots, x_n\}$  を生成する。(図 2 のように、ツアー中の二つのリンクを取り出して、その二つのリンクを交換するのと同じ意味)(図 2 参照)
- ⑤ 改善された場合 ( $tour_{par}$  のツアー長  $L_{par}$  より  $tour_{chi}$  のツアー長  $L_{chi}$  が大きい) は終了。その他の場合は、③に戻り繰り返す。

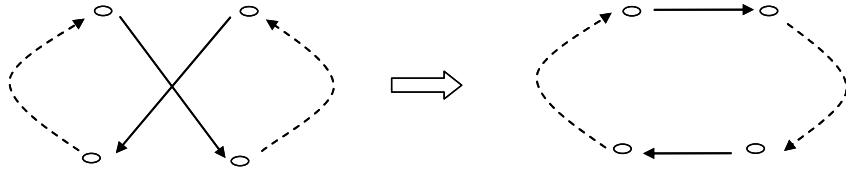


図 2 2-opt 法  
Figure 2 2-opt-type

## (3) ブロック(NI)型突然変異

2-opt 型突然変異は効率的にツアーを改良でき、短時間に良好な解を得ることができる。しかし、このような効率的な探索は、逆に局所最適に陥る危険性を高めてしまう。より最適性の高い解を得るためには、局所最適から脱出するためにツアーの一部を破壊し再構築する操作が必要である。このため 1 つの遺伝子だけでなく、その近傍の遺伝子も再構築するブロック型突然変異を提案する。これは以下のような手順から成る。

- ① 親個体  $tour_{par} = \{x_1, x_2, \dots, x_n\}$  を選択する。  $tour_{chi}$  は子個体とする。
- ②  $tour_{chi}$  からノード  $x_i$  をランダムに選択する。
- ③  $tour_{chi}$  からノード  $x_i$  とその近傍のノード  $\{x_{i-r}, \dots, x_{i+r}\}$  を削除する。近傍の大きさ  $r$  は問題

依存知識により決定する。例えば  $0$  から  $B \times$  (デポと呼ぶ配送基地から最も遠いノードまでの距離)の間の乱数で決定する。  $B$  は問題に依存した定数である。

- ④ ブロックと呼ぶノードの順序集合  $\{x_{i-r}, \dots, x_{i+r}\}$  中のノードを先頭から順に  $tour_{chi}$  に NI 法で挿入していく。全てのノードが  $tour_{chi}$  に挿入されたら終了する。

### 3.2.2 ランダムリスタートによる個体集団の多様性の維持

ある指定された世代においては、生成された子個体の中で適応値が低い半数の個体を、ランダム NI 生成により初期化され、2-opt 突然変異などにより改良された始祖個体へと置き換える。また毎世代、同じ評価値を持つ点で類似した個体群に対し同様に始祖個体への置き換えを行う。このように進化の途中世代の子個体を始祖個体へ置き換える操作をランダムリスタート処理と呼ぶ。

このランダムリスタート処理により、進化への貢献度の低い個体をランダム性の強い始祖個体へと置き換える事により、個体集団の多様性を維持する。

ランダムリスタート処理を実施する世代の決め方として、ここでは 2 つの方式を提案する。1 つは、最初の数世代にのみ実施する方法である。これは、始祖個体が改良されるまでには数世代必要となるため、探索処理の後半世代に実行しても効率が悪いという考えに基づく。もう 1 つは、数世代毎に実施する方法である。これは、定期的に行う事により、局所収束の防止と探索の効率のバランスが取れるであろうという予測に基づいて提案した。

### 3.3 評価・淘汰

GA の収束を速め、かつ、高精度な解を得るために、個体は親子両方を含む集団から選ぶ。即ち、劣った個体にも生存の機会を与え、集団の多様性を維持することを目的に、集団数の指定割合まで、上記の親子集団から各個体の評価値とは無関係にランダムに選ぶ。但し、進化効率を高めるため、親個体と同じ子個体が生成されたら、1 個体だけを選ぶ。逆に、優良個体に一定割合で置き換える方策、つまり増殖もベースとして行われる。

ランダムリスタート処理が実施された世代では、淘汰、すなわち生き残り個体選択のステップですべての子個体を生存させる。または、置き換えられた元の個体の適応度を継承することも可能である。生成されたばかりの始祖個体は他の親個体に比べて適応値が低いため、そのまま競争させるとすぐに淘汰されてしまう可能性が高く、多様性の維持に役立たない。そこで、生成後に保護育成期間を設ける事により、他の進化した個体と対等に競争できる機会を与える。すなわち、ランダムリスタート処理により生成された始祖個体を初期保護することにより、不当競争の防止策を講じる。

```

Algorithm randomStart(root) //全ランダム列の最近挿入と局所改良による始祖個体生成
  root ← ランダム NI による生成個体
  root を 2-opt 突然変異によりを改良
  root の適応値を評価
End /* randomStart */
Algorithm makeChild(child, parent) //部分ランダム列最近挿入と局所改良による子個体生成
  child ← parent からブロック型突然変異で子個体生成
  child を 2-opt 突然変異によりを改良
  child の適応値を評価
End /* makeChild */

Algorithm Irr-GA()
  m ← 集団サイズ; n ← 子個体集団サイズ;
  time_end ← 終了時間
  For i ← 1 to m Do //初期個体集団の生成
    randomStart ( survivali )
  End For
  世代数 g ← 0
  While time_now + time_inter < time_end Do
    親個体集団 parent ← 生き残り集団 survival
    g ← g + 1; // g 世代目
    For j ← 1 to n Do //カスケード処理による子個体生成
      親 parentp をランダム選択
      makeChild( childj, parentp )
    End For
    子個体集団を適応度の高い順に並べ替え
    For j ← 1 to n Do //適応値の同じ個体をランダム NI 生成による個体に置き換え
      IF childj の適応値と childj+1} の適応値が同じ Then
        randomStart ( 子個体 childj )
      End IF
    End For
    If 世代数 g がランダムリスタート処理実施世代 Then
      For j ← n/2 to n Do //子個体集団の適応度の低い半分をランダムリスタート
        randomStart ( 子個体 childj )
      End For
      For j ← 1 to n Do survivalj ← childj End For //子個体はすべて保護
      For j ← n+1 to m Do survivalj ← parentj-n-1} End For
    Else
      p ← 1; c ← 1;
      For j ← 1 to m Do //親子集団の適応値の高い上位 m 個が生存
        IF parentp > childc Then
          survivalj ← parentp ; p ← p + 1;
        Else
          survivalj ← childc ; c ← c + 1;
        End If
      End For
    End If
    time_inter ← 現在実行時間 - time_now; time_now ← 現在実行時間;
  End While
End /*Irr-GA*/
  
```

図 3 提案方式(複数方策活用内部ランダムリスタート GA : Irr-GA)の擬似コード  
 Figure 3 Pseudo code of Inner Random Restart GA (Irr-GA)

### 3.4 提案手法の詳細

上記の手法により構成される複数方策活用内部ランダムリスタート GA (Irr-GA) の詳細アルゴリズムの基本部分を図 3 に示す. 実行前に設定すべきパラメータとしては, 集団サイズ *m*, 子集団サイズ *n*, 終了時間 *time\_end*, ランダムリスタート処理実施世代などがある. 本論文の実験では, 対話リアルタイム応答を実現するため, 終了時間は 3 秒と設定している. また, 的確に設定した終了時間内に終了させるため, 各世代終了時に次世代の終了予測時間を計算している.

## 4. 関連研究

近似解法は様々な手法が提案されている. TSP のヒューリスティックスサーチ手法としては, Lin-Kernighan (LK)法が有名であるが, 複雑である. このため, LK 法や, その改良方式[2][6]は, 得られる近似解の最適性は高いが計算時間が長い. これらのヒューリスティックスは SA (Simulated Annealing:焼きなまし法) や TS (Tabu Search:タブーサーチ), GA などのメタヒューリスティックスサーチ手法と組み合わせて使われることが多い.

SA[7]は理論的には, 局所解に陥る危険性を少なくして, 最悪誤差が 3%以下の解を得ることができる. しかし, 焼きなまし用の冷却速度やコスト関数など SA の多数のパラメータを問題に合わせて設定することは非常に難しく, また, 上記の準最適解を求めるには多大の計算時間を要するため, 物流や配送の現場では使い難い.

TS[8]も実用的に最適解に近い解を求めるためには計算時間が長い. また, 局所探索としての特性が強く, 解の多様化能力が弱いなどの欠点が指摘されているが, その改良手法が金澤らにより提案されている[9].

いわゆるランダムリスタート法[10]は, ランダムな初期解を改善するために 2-opt 法などのローカルサーチを適用し, 近似最適解を得ることができる. 具体例として, 貪欲ランダム化適応型探索手法 (GRASP : Greedy Randomized Adaptive Search Procedure) [11]や繰返し数の制限により応答性保証を可能とした改良型ランダムリスタート法[12]がある. この改良型ランダムリスタート法は, Mow-GA のベースとなるもので, 40 都市の TSP を約 100 ミリ秒 3%以内の誤差で解いたが, 200 拠点を越えると精度の保証が困難な場合が生じた.

TSP を効率的に解く為の GA についても様々な方式が提案されている. たとえば, LK 法や Edge Assembly Crossover (EAX) [13], Distance-preserving crossover (DPX) [14]などを利用することにより 1000 都市から 10000 都市にも及ぶ超大規模 TSP を効率的に解くための GA 手法が提案されている[15][16]. EAX や DPX は親ツアの枝の特

質を厳密にツアーへと継承させることができる。しかし、これらの交叉オペレータはツアーの解析が必要なため、計算時間が長く、超大規模な TSP 以外に用いるのは効率的でない場合がある。文献[15]は 2 種類の方法を多くのケースで比較している。それは、300-10000 都市 TSP には Cga-LK が有利であるが、198 都市 TSP にはランダム-LK が有利であることを示している。したがって、数千都市の TSP を効率よく解ければ、1—2 桁低い巡回拠点数や都市数の TSP が必ず効率よく解けるというわけではない。

本研究が対象とするようなスケール(数十から数百都市以上、2千都市未満)の TSP に関して、文献[15]では、TSPLIB の問題 lin105 を解いている。また、文献[17]でも、この問題を解く上での様々な交叉オペレータの性能比較実験が行われている。

本研究が対象とするスケールの TSP を高速に解くための GA 手法としては、Yan らによる GA 手法[18]が提案されている。

文献[3][5]において、前提案方式 Mow-GA、Miw-GA と Random-LK、文献[17]での実験の最良交叉オペレータを用いた GA 及び金澤らによる TS や Yan らによる GA との比較実験を行い、提案手法の速度と精度における優位性を実証した。

次節では、筆者らの以前の提案方式(Mow-GA,Miw-GA)と今回の提案方式 Irr-GA を比較する。さらに、TSP の厳密解法では世界最高速レベルの Concorde [19]と本論文の提案方式を比較する。これは厳密解法は最悪誤差が 0 という点において、本論文の前提条件を満たす解法だからである。

## 5. 実験と結果

### 5.1 実験

本節では、TSP のベンチマークテストである TSPLIB を用いて、前提案解法 (Mow-GA、Miw-GA) と本提案解法である Irr-GA との比較実験を行う。

実験では、AMD Athlon 64 X2 3800+ 2GHz プロセッサ (シングルコアで起動したため、Athlon 64 3200+ 2GHz とほぼ同性能) と 1GB のメモリを装備している PC を使用し、C 言語によりプログラムを作成した。コンパイルには Microsoft Visual C++ .NET 2003 ver. 7.1.3091 を使用し、コンパイルオプションは/O2 (実行速度優先) に設定し、Windows XP Professional 上で実行した。

また、GA のパラメータについては、集団サイズは Mow-GA が 50、Miw-GA が 100、突然変異率は Mow-GA が 50%、Miw-GA の各突然変異オペレータの選択確率は、ブロック型が 60%、2-opt 型が 40%である。試行回数は各問題 1000 回とした。これらのパラメータは事前の各々数回の試行結果に基づいて優良な値に設定した。Mow-GA、Miw-GA、Irr-GA の終了条件は、3 秒以内となる様に設定した。

### 5.2 結果

表 1 は前回の提案方式である Mow-GA、Miw-GA、と本論文の提案方式 Irr-GA を 12 種類のベンチマーク問題 (st70, eil76, kroA100, pr107, pr136, pr144, pr152, pr226, a280, lin318, pr439, rat575) に適用して得た比較結果である。試行回数はいずれも 1000 回、各回の実行時間はほぼ 3 秒である。Irr-GA は設定の異なる 2 種類、1Ar と 5Ar を用いた。いずれも指定世代 (本実験では最初の 5 世代) で子個体にランダムリスタート処理を実行した後、生存確率を高めるなど最適化効率を上げるための後処理として、改善可能なだけ 2-opt を施し、局所最適化をした。但し、後者の 5Ar は、2-opt による局所最適化の前に、ブロック(NI)型突然変異を施すことも含めたループを、全体として評価値 (ツアーの総距離) がランダムリスタート直後以上に向上するまで繰り返す。これは、ブロック(NI)型突然変異がブロック内の遺伝子を全てツアーに最近挿入する (拠点数の 2 乗オーダー) の処理をしても改良されない場合が多いため処理を重くするが、2-opt との複数方策 (発見的知識) による局所最適化によりランダムリスタートで生成された個体の最適化効率を少しでも上げ、その個体による進化のブレーキをできるだけ減らすためのものである。但し、本実験では 1Ar においても、この進化のブレーキを減らすため、ランダムリスタートの実施世代として他の個体の進化のあまり進んでいない第 1~4 世代を指定した。TSP の名前 (例えば、st70) に含まれる数字 (例えば、70) は、都市 (巡回拠点) の数を示す。各データ欄は、3 秒以内の対話リアルタイム応答時間で得られた解の全試行中の最悪ケースの誤差率 (%) を示す。誤差は、求められた解と厳密解における差を厳密解で割ることにより得られる相対誤差である。厳密解は TSPLIB に公開されている各問題の最短経路長を利用した。

表 1 の a280 から rat575 までの 200 拠点以上の比較的規模の大きい TSP では、Mow-GA や Miw-GA では、3 秒以内の対話リアルタイム応答時間での最大誤差率が 4% を越えた。これに対し、Irr-GA では、目標とする最大誤差率 3% 以下 (2.5% 前後) を達成した。200 以下の拠点数の問題では、いずれも 2% 以下の誤差で大差はなかった。

実験結果によると、Irr-GA は性能に問題による偏りが少なく、幅広い問題において最大誤差率を低く抑える事ができた。これはランダムリスタート処理による多様性の維持が有効に働いた結果であると推測される。また、Irr-GA は比較的大きい数百拠点の TSP に対しても最大誤差率を低く抑える事ができた。これは、ブロック突然変異による局所収束からの脱出と 2-opt 突然変異による局所探索のカスケード処理が比較的大きな問題にたいしても有効に働き、無駄な探索が押さえられた結果であると推測される。

表 1 TSPLIB による関連研究との比較結果  
 Table 3 Results compared with related works on TSPLIB

TSP 問題名	3 秒実行時の最適解からの誤差の最悪値[%]			
	Mow-GA	Miw-GA	Irr-GA	
			1Ar	5Ar
st70	0.194	0.592	0	0
eil76	0.788	1.858	1.487	1.673
kroA100	0	0	0	0
pr107	0.000	0.189	0.305	0.099
pr136	0.105	1.002	1.129	0.989
pr144	0.008	0.090	0.090	0.091
pr152	0.076	0.184	0.184	0.185
pr226	2.767	0.004	0.005	0.005
a280	14.036	4.187	2.016	2.055
lin318	9.331	3.117	2.270	2.460
pr439	11.535	4.603	0.782	1.381
rat575	9.139	7.027	2.156	2.288

次に、表 2 に応答性能において世界最高速レベルであると考えられる厳密解法、ABCC (Concorde として有名) と本論文の提案方式 Irr-GA 及び以前提案方式である Mow-GA, Miw-GA との比較結果を示す。第 2 列から最右列の手前までは、Mow-GA, Miw-GA と Irr-GA の各種方式を最左列に名前のある TSPLIB の問題に対し 1000 回試行した時の最大誤差率 (%) を示す。最右列は、ABCC (厳密解法なので誤差は 0) の求解時間 (秒) である (文献[19])。CPU は異なるが、a280 などに対し本実験環境で実測した場合と求解時間に一割前後の違いしかなかったため、表 2 にそのまま掲載した。結果を見ると、Concorde で 10 秒以上かかる 200 拠点前後以上の規模の TSP を Irr-GA は 3 秒以内に 3% 程度以下の誤差で解くことを確認することができた。

表 2 TSPLIB を用いた Concorde との比較結果  
 Table 4 Results compared with Concorde on TSPLIB

TSP 問題名	Irr-GA 3 秒実行時の最適解からの誤差の最悪値[%]		ABCC 求解時間 [sec]
	1Ar	5Ar	
pr136	1.129	0.989	4
pr152	0.185	0.185	8
rat195	1.593	1.291	22
kroA200	1.056	1.056	7
ts225	0	0	21
pr226	0.005	0.005	4
gil262	2.145	1.766	13
a280	2.016	2.055	38
pr299	0.807	0.811	39
lin318	2.270	2.460	10
pr439	0.783	1.381	216
rat575	2.156	2.288	313

## 6. 結論

小包や手紙の配送では、30~数百拠点 (都市) の最短道順探索問題 (巡回セールスマン問題: TSP) を対話的応答時間 (3 秒以下) かつ誤差 3% 以下で解く必要がある。このため、メタ戦略と発見的知識を組み合わせ、GA への融合する方式を提案した。これは、1 つの GA の指定世代において、メタ戦略としてのランダムリスタートを評価の悪いまたは同種の (ここでは評価が同じ) 子個体群に適用し、複数の発見的知識を用い局所最適化した後に世代の最終評価をして、集団の多様性を保ちながら局所最適をできるだけ避け求解する方式である。従来の近似解法では規模が 250 都市以上で最悪誤差が 4% を越えていた。また、厳密解法では世界最速レベルの Concorde でも 200-600 都市前後になると 3 秒以内では求解ができなかった。本方式では、300 都市前後までの TSPLIB の問題を 1000 回試行した結果、最悪ケースでも 3% 以下の誤差の解が 3 秒以内に探索可能なことを実験で確認した。

今後は、上記の多様な問題パターンに対しても必要な応答性・最適性を保証するため、分かり易く拡張性の高い本方式の特徴を生かし、更なるメタ戦略や多種・多数のヒューリスティクスの GA への効率の統合・協調方式を研究する。特に、拠点数が千前後でも

3%以下の誤差の解を3秒以内に探索できるように調整・改良する。また、一方通行などのある、より実用的・一般的な配送道順の最適化問題（非対称 TSP）についても研究を行う。このため、方策やその組み合わせ統合パラメータの自動調整方式・学習方式についても研究を進める。

**謝辞** 本研究は柏森情報科学振興財団の助成を受けて遂行された。本研究は、東京電機大学総合研究所研究 02Q823 として行ったものである。

### 参考文献

- 1) Lawer, E., Lenstra, J., Rinnooy, K., and Shmoys, D.: "The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization", John Wiley Sons, (1985)
- 2) 山本芳嗣, 久保幹雄: 「巡回セールスマン問題への招待」, 朝倉書店, (1997)
- 3) 櫻井義尚, 小野山隆, 久保田仙, 鶴田節夫: 「配送問題を対話的時間で実用レベル最適化する多段知能型 GA」, 日本知能情報ファジィ学会誌, Vol.20, No.4 pp.639-652 (2008.08)
- 4) Sakurai, Y., Onoyama, T., Kubota, S., Nakamura, Y., Tsuruta, S.: "A Multi-world Intelligent Genetic Algorithm to Interactively Optimize Large-scale TSP", Proc. of The 2006 IEEE International Conference on Information Reuse and Integration (IEEE IRI2006), Hawaii, USA, pp.248-255 (2006)
- 5) Y. Sakurai, T. Onoyama, S. Kubota, S. Tsuruta: "A Multi-inner-world Genetic Algorithm to Optimize Delivery Problem with Interactive-time", Proc. of 4th IEEE Conference on Automation Science and Engineering (CASE 2008), Washington DC, USA, pp.583-590, (2008.8).
- 6) Lin, S. and Kernighan, B.W.: "An effective heuristic algorithm for the traveling salesman problem", Operations Research, Vol. 21, No.2, pp.498-516 (1972)
- 7) Miki, M., Hiroyasu, T., Jitta, T.: "Adaptive Simulated Annealing for maximum temperature", Systems, Man and Cybernetics, 2003. IEEE International Conference, vol.1, pp.0-25 (2003)
- 8) Fang, Y., Liu, G., He, Y., Qiu, Y.: "Tabu search algorithm based on insertion method", Proceedings of the 2003 International Conference on Neural Networks and Signal Processing, pp.420-423 (2003)
- 9) 金澤貴彦, 安田恵一郎: 「Proximate Optimality Principle に基づく Tabu Search」, 電気学会論文誌. C, 電子・情報・システム部門誌, Vol.124, No.3 pp.912-920 (2004). Kanazawa, T., Yasuda, K., "Proximate Optimality Principle Based Tabu Search", IEEE Transactions on Electronics, Information and Systems, Vol.124, No.3, pp.912-920, 2004.
- 10) 柳浦 睦憲, 茨木 俊秀: 「組合せ最適化問題に対するメタ戦略について」, 電子情報通信学会論文誌. D-I, 情報・システム, I-情報処理 Vol. J85-D-II, No8 pp.3-25 (2000). Yanagiura, M. and Ibaraki, T., "On Metaheuristic Algorithms for Combinatorial Optimization Problems", Transactions of the IEICE, Vol.J85-D-II, No.8, pp.3-25, 2000
- 11) Feo, T.A., Recende, M.G.C., and Smith S. H.: "A Greedy Randomized Adaptive Search Procedure for Maximum Independent Set", Operations Research, Vol. 42, No.5, pp.860-878 (1994)
- 12) Kubota, S., Onoyama, T., Oyanagi, K. and Tsuruta, S.: "Traveling Salesman Problem Solving Method fit for Interactive Repetitive Simulation of Large-scale Distribution Network", Proc. IEEE SMC'99, pp.533-538 (1999)
- 13) 永田 裕一, 小林 重信: 「巡回セールスマン問題に対する交叉 : 枝組み立て交叉の提案と評価」, 人工知能学会誌, Vol.14, No.5 pp.848-859 (1999). Nagata, Y. and Kobayashi, S., "The Proposal and Evaluation of a Crossover for Traveling Salesman Problems: Edge Assembly Crossover", Journal of Japan Society for AI, Vol.14, No.5, pp.848-859, 1999.
- 14) Whiteley, D. and Starkweather, T.: "Scheduling Problem and Traveling Salesman: The Genetic Edge Recombination Operator", Proceeding of ICGA'89, pp.133-140 (1989)
- 15) Baraglia, R., Hidalgo, J.I., Perego, R.: "A hybrid heuristic for the traveling salesman problem", IEEE Transactions on Evolutionary Computation, Vol.5, Issue 6, pp.613-622 (2001)
- 16) Nguyen, H.D., Yoshihara, I., Yamamori, K., Yasunaga, M.: "Implementation of an Effective Hybrid GA for Large-Scale Traveling Salesman Problems", IEEE Transactions on Systems, Man and Cybernetics, Part.B, Vol.37, Issue. 1, pp.92-99 (2007)
- 17) Cheng, C., Lee, W., Wong, K.: "A genetic algorithm-based clustering approach for database partitioning", IEEE Transactions on Systems, Man and Cybernetics, Part C, Vol.32, Issue 3, pp.215-230 (2002)
- 18) Yan, X., Liu, H., Yan, J., Wu, Q.: "A Fast Evolutionary Algorithm for Traveling Salesman Problem", Proc. of Third International Conference on Natural Computation (ICNC 2007), Vol.4, pp.85-90 (2007)
- 19) G. Gutin, A. P. Punnen (Eds.): "The Traveling Salesman Problem and its Variations", Springer, NY, USA, (2007).