原著論文

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[†]

櫻井 義尚^{*1}・小野山 隆^{*2}・久保田 仙^{*2}・鶴田 節夫^{*1}

配送問題を巡回セールスマン問題（TSP）に定式化し，これを対話的応答時間内に解き，しかも専門家レベルの最適解を得ることを目的とする多段知能型GAを提案する．GAにおいて最適化問題を効率的に解くためには問題固有の特性に従ったヒューリスティックスや知識を用いて探索をするが，これにより偏った探索になる可能性がある．提案するGAは，TSPに有効なヒューリスティックスや知識を導入することにより，効率的な探索を実現し，探索特性の異なる相補的性質を持つGAを複数多段に組み合わせて用いることにより，探索の偏りを防ぎ，安定的に解を求める事ができる．

キーワード：TSP（Traveling Salesman Problem, 巡回セールスマン問題），GA（Genetic Algorithm, 遺伝的アルゴリズム），ヒューリスティックス

1. はじめに

物流の効率化は経済面からだけではなく，環境・社会面から見てもCO₂や排気ガスの節減などにもつながるため重要な問題であり，配送ネットワークを構築するなどの効率化が図られている．また，局所的な効率化のみを考えるのではなく，生産から最終需要にいたる商品供給の流れすべてを効率化することを目指したサプライチェーンマネジメントを導入する企業が増えている．そこで，多数の配送先と多数のトラックにより構成されるより大規模な配送ネットワークの最適化の要求が高まっている．

このような配送ネットワークの構築では，まず製品の製造を行う工場や，輸送の中継に用いるデポの配置，部品輸送用のトラック配置等の条件設定を人間が行う．次に，1000近くの部品メーカーとデポ・工場を結ぶ数100の集配送ルートを作成して輸送コストとCO₂等の排出量を算出する．最後に経済面及び環境面から人間が総合的に評価する．これらの手順を繰り返す事により，配送ネットワークの最適化を行う．

配送ネットワーク最適化作業の効率化を考えた場合，集配ルートを自動生成することができればその効

果は大きい．このルートの最適性の評価には単純な走行コストだけでなく，社会面からの人間の判断が不可欠になる．このため，対話的応答時間内での解の導出が求められる．許容できる対話的応答時間を例えば数10秒未満としても，大規模な配送ネットワークの構築には，2.1節に詳述する様に，1回のシミュレーションで数10都市のTSP(Traveling Salesman Problem)と同等の問題を数100，解く必要がある．つまり集配ルートの自動生成には，数10都市TSP相当の問題を数10ミリ秒以内に解く実時間応答性を持ったTSP解法が必要である．

また，解の最適性について考えると，人間の専門家は集配ルートを生成する時，数学的な最適解に対して2～3%の誤差を出すことも多いが，それ以上の誤差を出し実用上の問題を起こすことはない．そのため，集配ルートを生成するTSP解法には，数学的に最適でなくとも良いから安定的に高い最適性が得られる専門家レベルの最適性が求められる．

TSPの厳密解法としては，Branch-and-cut法[Grotschel1991]やDynamic Programming(DP)を用いた解法[Bellman1962],[Bertsekas1987],[Arora1998]が提案されている．しかし，厳密解法は最適解を得られるが，計算時間が長い．それに対して，最適解を求める代わりに短い時間で良い解を出す近似解法がある．汎用的な近似解法としては，Simulated Annealing(SA)法やTabu Search(TS)法を用いる方式[Ibaraki1993],[Hooker1995]，遺伝的アルゴリズム(Genetic Algorithm, GA)[Davis1991]を用いる方式が提案されている．SAやTSは，最適性の高い解を得られるが，計算

[†] A Multi-step Intelligent Genetic Algorithm to Optimize Delivery Problem within Interactive-time
Yoshitaka SAKURAI, Takashi ONOYAMA, Sen KUBOTA and Setsuo TSURUTA

^{*1} 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

^{*2} 日立ソフトウェアエンジニアリング株式会社
Hitachi Software Engineering Co.,Ltd.

時間が長く、数10ミリ秒以内の実時間応答性が要求される問題には適していない。一方GAは、それに用いられるオペレータを工夫することにより、様々な特性を持った解法になる。これらの汎用的な近似解法は適用する問題に特化したヒューリスティックスを用いる事により、より効率的に探索することが可能になる。

TSPに対しては、これまで様々なヒューリスティックスが提案されている [Lin1972], [Lawer1985], [Kolen1994], [Yamamoto1997], Lin-Kernighan (LK) 法やその改良手法 [Lin1972] は高い最適性が得られる一方、計算コストが高く、また専門家レベルの最適性を常に保証することはできない [Kubota1999]。2-opt 法やNearest Insertion (NI) 法 [Yamamoto1997] はLK法に比べると解の最適性はそれほど高く無いが、計算コストが小さい割に高い最適性が得られる。

そこで、著者らの近似解法では、遺伝的アルゴリズム (Genetic Algorithm, GA) [Davis1991] のオペレータに2-opt法やNI法などのヒューリスティックスを用いた [Onoyama2000]。この手法はGAの世代数を制限し、初期解生成、交叉、突然変異などのGAオペレータに2-opt法やNI法を導入することにより数10ミリ秒以内の実時間応答性を保証することができた。しかし、幾つかの巡回拠点配置パターンでは、この手法は局所最適解に陥り、専門家レベルの最適性を保証することができなかった。そのため、すべてのケースにおいて専門家レベルの最適性を保証するための更なる改良が必要であった。

本論文では、上記を更に改良しTSPに有効なヒューリスティックスあるいは知識を利用した多種の初期解生成、交叉、突然変異オペレータを用意し、これらの組合せを違えた複数のGAをカスケードする多段進化型の近似解法を提案した。これにより、多様な巡回拠点配置パターンに対して、対話的応答時間内に専門家レベルの最適性で解くことを保証可能とした。

以下、2節では、大規模配送ネットワークシミュレーションの概要を説明し、その技術的課題を明らかにする。3節で上記問題に対する解法を提案し、4節で提案解法の評価実験方法とその結果を示す。5節では実験結果に基づき提案解法の有効性を示す。6節で他手法との比較を行い、7節で結論を述べる。

2. 大規模配送ネットワークシミュレーションの技術的課題

2.1 大規模配送ネットワークシミュレーション

部品メーカーと工場とを繋ぐ製品生産のための大規模な配送ネットワークの構成を図1に示す。

従来は、部品メーカーが個別に工場へ直接輸送して

図1 大規模な配送ネットワーク

いたが、物流の効率化を図るために、1つのサプライチェーンに属する部品メーカーが図1のようなデポ (物流センター) を利用した共同配送を行うための配送ネットワークを構築している。部品メーカーから工場への配送には、図1(a)のようなデポ経由の配送と、図1(b)のように工場から直接部品メーカーを巡回する配送の2種類がある。デポ経由の配送では、各デポは複数のトラックを所有し、決められた配送エリア毎に担当のトラックが決められている。配送エリアは部品メーカーの所在する地域毎に決められている。部品配送の際には、各デポから複数のトラックが発出し、各トラックが複数の部品メーカーを巡回して部品を回収してデポに戻る。そして、デポから工場へは、回収した部品を一括して配送するが、これは本論文の提案手法による最適化の対象ではない。このデポ経由の配送は、一般に出荷量がそれほど多くない部品メーカーに対して行われる。一方、出荷量の多い部品メーカーに対しては、工場から出発したトラックが直接部品メーカーを巡回して工場に部品を配送する。このデポまたは工場を起点とした複数の部品メーカーへの巡回路を配送ルートと呼ぶ。配送ルートは、1台のトラックにより構成される場合と、ある配送エリアを担当する複数のトラックにより構成される場合がある。また、配送先には、時間制約や優先順位ではなく、1台のトラックが集配する荷物量はそれほど大きくないため、トラックの積載量制約を考慮する必要もない。

本論文では、1台のトラックによる配送ルートにおける配送順を最適化する手法を提案する。これを配送ルートが複数のトラックにより構成される場合に拡張する事は容易であり、5.2節において、その適用例を示す。

このような配送ネットワークを最適化することにより、配送効率を高め、販売費の節減と、排気ガスの減量に貢献する。

上記の配送ネットワークには、部品メーカーが数

図2 シミュレーションの過程

100から1000強、デポおよび工場が合わせて10近くから大きいものでその倍程度存在する。また、デポもしくは工場の1つを起点とする配送ルートは数100にもなる。現場の調査によると、それぞれの配送ルートの規模は数10から100強の部品メーカーを巡回するものとなる。

上記の大規模な配送ネットワークの最適化は、図2に示すように様々な条件のもとで配送ネットワークのシミュレーションを繰り返すことにより行われる。各シミュレーションでは、手作業により設定された様々な条件に基づいて、コンピュータにより数100もの配送ルートを生成し、最終的に配送コストだけでなく社会的側面もふまえて、人間により最適な配送ネットワークが判断される。

2.2 技術的課題

上記の配送ルート生成問題は、理論的にTSPと同等である。したがって、上記のような大規模な配送ネットワークを最適化するために、以下の技術的課題を解決しなければならない。

1) 最適性：専門家が数100の配送ルートを作成する場合、必ずしも数学的に最適解を作成しているわけではなく、2～3%程度は最適解から誤差を含んだルートを作成している。しかし、実用面で問題が生じるような大きな誤差を含んだルートを作成することは無い。そのため、誤差率3%以下の最適性を保証する必要がある。

2) 応答性：トラックの配送ルートの評価にはコストだけではなく、社会、環境的要因など多様な要因についても考慮を必要とするため、コンピュータが生成するトラックの配送ルートを人間の専門家が手動でチェックする必要がある。そのため、ルート生成システムには対話的応答時間内での実時間応答性が求められ、応答時間が最長でも数10秒以内であることが求められる。大規模な配送ネットワークでは、通常およそ数100から1000程度のルートが存在するので、1つの

配送ルートを数10ミリ秒以内で生成しなければならない。

3. 多段知能型GA (MIGA)

1節で述べたように、大規模な配送ネットワークシミュレーションにおける配送ルートの作成問題は、TSP (巡回セールスマン問題)に分類される。本論文では、特に対称ユークリッド型TSP[Yamamoto1997]を前提に解法を提案する。

3.1 多段知能型GAの概要

2.2で述べた技術的課題、①専門家レベルの最適性、②実時間応答性を様々な巡回拠点配置パターンにおいて満たすTSPの近似解法として、多段知能型GA (Multi-step Intelligent Genetic Algorithms, MIGA)を提案する。GA[Davis1991]は生物の進化をモデル化した近似解の探索アルゴリズムである。GAは、はじめに初期解生成により解の候補となる個体群(初期集団)を生成する。そして、その個体群をもとに交叉、突然変異オペレータにより新しい個体(解候補)を生成し、評価、選択・淘汰などにより解の改善を行う手法である。

また本解法では、解の候補となる個体はTSPにおけるツアー(配送ルート)を表現する染色体を持ち、その遺伝子型は図3のような巡回する順番にノード番号(配送先のID番号)を並べた構造(各遺伝子はノード番号を表現する)になっている。

(1) ヒューリスティックスを用いたGA：

GAで効率的な最適化を行うためには、問題の特徴に合った突然変異オペレータと交叉オペレータを利用する必要がある。そのため、TSPに有効なヒューリスティックスをGAのオペレータに組み込むことにより探索の効率化を図ることができる。そこで以下のようなGAを提案した。

2-opt型GA：シンプルで汎用性の高いヒューリスティックスである2-opt法を突然変異オペレータに用

図3 TSPにおける遺伝子型

いる。2-opt法は1組の枝を組み替える事によりツアーを改善していく探索法である。2-opt型突然変異はランダムに指定された1組の枝に対し、これを用いて素早くツアーを改善する。そのため、一般的に短い時間で優良解の導出が望める。しかし局所最適解に陥るリスクも持っている。

この手法は、ある特定の巡回拠点配置パターンにおいては効率的でない解を導出することが実験により確認されている。そこで2-opt型GAの苦手な問題を克服することを目的として以下のようなGAを提案する。

ブロック型GA: 巡回拠点数の2乗の計算コストがかかるが高い最適化が望めるヒューリスティックスであるNI法を主に用いる。NI法は挿入した結果、ツアー長が最も短くなる位置にノードを挿入する挿入法であり、このNI法とTSPを解く専門家の知識をモデル化した後述の挿入順制御NI生成を用いることにより最適性の高いツアーの生成を行う。また、一部の遺伝子だけでなく、その近傍の遺伝子を再構築するブロック型突然変異を用いることにより局所最適解への収束を防止する。これにより2-opt型GAとは違う探索特性を持つ。

(2)多段知能型GA(MIGA):

様々な巡回拠点配置パターンに対して、専門家レベルの最適性と実時間応答性を実現するのは大変困難である。探索を効率化するための様々なヒューリスティックスを利用した、突然変異オペレータや交叉オペレータが提案されているが、特定のパターンに効果的なヒューリスティックスは他のパターンには有効とは限らない。また、過度に複雑なアルゴリズムやヒューリスティックスは応答性を悪化させる。

そこで、相補的な性質を持つ2つのGAを組み合わせることにより様々な巡回拠点配置パターンに対して、専門家レベルの最適性と実時間応答性を実現する多段知能型GA(Multi-step Intelligent GA; MIGA)を提案する。

この手法は、主にシンプルで汎用性の高いヒューリスティックスを用いているため高速に最適化が行える2-opt型GAと、計算コストは高いが最適化能力と局所最適解からの脱出効果の高いブロック型GAの2つの異なる探索特性を持つGAを組み合わせる事により、多様な巡回拠点配置パターンで局所最適解に陥るのを回避しつつ、高い応答性を保証する。上記の2つの手法は独立に実行され、結果が統合される。

(3)GAの実行世代の制限:

実時間応答性を保証するためのGAの実行時間は以

下の表1のように算出される。

表1 GAのパラメータと実行時間

n	個体集団数
l	個体の遺伝子の数
$T(X)$	X の計算時間
$Prob(X)$	X の確率
$Ave(X)$	X の平均
$T(\text{初期集団生成})$	$n \times C_{ini} \times f_{ini}(l)$
$T(\text{GA の } l \text{ 世代})$	$T(\text{交叉}) + T(\text{突然変異}) + T(\text{評価}) + T(\text{選択})$
$T(\text{交叉})$	$n \times Prob(\text{交叉}) \times Ave(T(\text{交叉}))$
$Ave(T(\text{交叉}))$	$C_{cros} \times f_{cros}(l)$
$T(\text{突然変異})$	$n \times Prob(\text{突然変異}) \times Ave(T(\text{突然変異}))$
$Ave(T(\text{突然変異}))$	$C_{mut} \times f_{mut}(l)$
$T(\text{評価})$	$C_{fit} \times f_{fit}(l, n)$
$T(\text{選択})$	$C_{sel} \times f_{sel}(n)$
$GA \text{ の実行時間}$	$T(\text{初期集団生成}) + \text{世代数} \times T(\text{GA の } l \text{ 世代}) + \text{マージン定数 } C$

GAのパラメータとしては、 n 、 l 、 $Prob(\text{交叉})$ 、 $Prob(\text{突然変異})$ は既知である。 $f_{ini}(l)$ は初期集団の生成、 $f_{cros}(l)$ は交叉、 $f_{mut}(l)$ は突然変異、 $f_{fit}(l, n)$ は評価、 $f_{sel}(n)$ は選択の各処理の計算の複雑さである。但しCPUなどハードウェア性能に依存しないものを考えている。これらはアルゴリズムを解析することにより導出される。例えば、クイックソートアルゴリズムが実装されている $f_{sel}(n)$ は次式により導出される。 $f_{sel}(n) = n \times (\log n) + C_l \times n + C_0 \dots$ (式1)。 C_{ini} は初期集団の生成、 C_{cros} は交叉、 C_{mut} は突然変異、 C_{fit} は評価、 C_{sel} は選択の各処理の計算の複雑さから計算時間を求めるための定数である。クイックソートによる計算の複雑さから選択処理の複雑さを求めるための式1の定数は C_l と C_0 である。これらは実験結果から特定される。また、プログラムのステップ数を調べ、CPUアーキテクチャのような詳細を考慮することにより、より正確に特定される。GAの1世代あたりの実行時間は確率的に変動する。しかしながら、実行時間を基本的なコンポーネント単位に分割し、コンポーネントの計算の複雑度と定数を同定するための実験結果から計算することにより、推定誤差は許容範囲内に納めることができる。さらに、この推定誤差を吸収し実時間応答性を保証するため、定数マージン C がユーザにより設定される。

この実行時間を用いることにより、要求反応時間内に実行可能なGAの世代数が計算される。

3.2 多段知能型GAの構成要素

(1) 初期集団生成：

局所最適解に収束するのを防ぎ、最適性の高い解を得るためには、初期集団の多様性が重要になる。しかし完全にランダムな初期集団を用いると収束速度は遅くなる。迅速に最適性の高い解を得るためには、初期集団の生成段階からある程度の最適化が必要になる。このため以下のように最適化の強さの違う3種類の方式を用いる。

a) ランダム生成

ノードの順序をランダムに並べるによりツアーを生成する。

b) ランダムNI生成

ランダムな順序でノードをNI法を用いてツアーに挿入していくことによりツアーを生成する。

c) 挿入順制御NI生成

専門家がツアーを生成するとき、彼らはたいてい全体のルートの概観を考えながら近いものから順に巡回する順番を決めていく。このような専門家の全体指向的な決め方をモデル化することにより、挿入順制御NI生成を提案する。

具体的には、以下のような手順でツアーを生成する。

- ① $Que(nodes)$ はノードのキューであり、各ノードのチェックカウンタ $check\ count$ は0で初期化されている。 $tour_{gen}$ は生成されたツアー、 w は閾値であり、1以上の実数である。その初期値は問題依存知識により決められる。例： $w = (Dist_{ave} + d \times \sigma) / Dist_{ave} \dots$ (式2)。 $Dist_{ave}$ はそれぞれのノードとデポとの平均距離、 σ はノード間距離の標準偏差、 d の初期値は1.0。
- ② すべてのノードをキュー $Que(nodes)$ にランダムにエンキューする。
- ③ $Que(nodes)$ から1つのノード $node_{add}$ をデキューする。
- ④ 一時的に $node_{add}$ を $tour_{gen}$ にNI法を用いて挿入する。
- ⑤ 挿入前の $tour_{gen}$ の長さ L_{pre} と挿入後の $tour_{gen}$ の長さ L_{after} を評価する※。
- ⑥ もし $L_{after} < (L_{pre} \times w)$ なら $node_{add}$ はそのまま $tour_{gen}$ に挿入され、 w は初期値に戻る。そうでなければ、 $node_{add}$ を $Que(nodes)$ にエンキューして $node_{add}$ の $check\ count$ を1増やす。
- ⑦ もし $Que(nodes)$ の先頭ノードの $check\ count$ が0でないなら、 w は増加し、 $Que(nodes)$ の全てのノードの $check\ count$ は0に初期化される。この

w の増加量も問題依存知識により決定される。

例：①の例 $d := d + 0.5$ のように d を更新した後、(式2)により w を更新する。

- ⑧ もし $Que(nodes)$ が空なら、終了。そうでないなら、手順③へ戻り繰り返す。

※評価方法としては、ダイクストラ法で事前に計算した表から2地点間の距離を取り出し全ての拠点間でこれを加算して計算する。

(2) 評価：

各個体の評価(適応度)は、その世代の個体集団の中の最小のツアー長を L_Min とするとき、ツアー長が L の個体の適応度を $F = L_Min / L$ で計算する。つまり、ツアー長の短い個体ほど適応度は高い。

(3) 選択：

決められた個体数だけを次の世代に残し、残りの個体を淘汰する。選択する個体は全個体(親個体集団と子個体集団)の中から適応度の高い順に選ぶ。また、たとえ適応度が低くても改善の機会を与えるため個体数の10%はランダムに決定される。但し、同じ個体は重複しては選択しない。

(4) 交叉オペレータ(NI型交叉)：

交叉で二つの親の良い形質を子に継承してGAで迅速な解の収束を実現するために、NI(Nearest Insertion)法を用いた交叉オペレータを提案する。これをNI型交叉と呼び、以下の手順から成る。(図4参照)

- ① $tour_{par1} = \{x_1, x_2, \dots, x_n\}$ は親個体1、 $tour_{par2} = \{y_1, y_2, \dots, y_n\}$ は親個体2、 $tour_{chi}$ は子個体。
- ② 親個体1 $tour_{par1}$ の交叉点 x_i を求める。
- ③ 親個体1 $tour_{par1}$ の交叉点 x_i より前の部分で構成される部分ツアー $\{x_1, x_2, \dots, x_i\}$ を子個体 $tour_{chi}$ にコピーする。
- ④ 親個体1 $tour_{par1}$ の残りの部分ツアー $\{x_{i+1}, \dots, x_n\}$ を親個体2 $tour_{par2}$ の順序に並べ替える。
- ⑤ 並べ替えた部分ツアーを先頭から順に子個体 $tour_{chi}$ にNI法を使って挿入していく。親個体1 $tour_{par1}$ の全てのノードが子個体 $tour_{chi}$ に挿入されたら終了。

このようにして作成した子個体を新しい個体として用いる。この交叉オペレータは、二つの親個体に含まれているノードの順序を子個体に継承でき、GAの収束性を高めることができると考えられる。

図4 NI型交叉具体例

(5) 突然変異オペレータ:

突然変異にも、問題の特徴に合ったオペレータを用いないと迅速な解の収束を実現できない。このため次の二つの突然変異オペレータを用いる。

a) 2-opt 型突然変異

このオペレータは単純な局所探索ヒューリスティクスである2-opt法を用いることにより迅速な解の収束を可能とする。これは以下のような手順から成る。

- ① $tour_{par} = \{x_1, x_2, \dots, x_n\}$ は親個体, $tour_{chi}$ は子個体。
- 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ていた場合は終了する。
- ④ $tour_{par} \setminus \{x_1, \dots, x_i, x_{i+1}, \dots, x_j, x_{j+1}, \dots, x_n\}$ の部分ツアー $\{x_{i+1}, \dots, x_j\}$ を反転させることにより, $tour_{chi} \setminus \{x_1, \dots, x_i, x_j, x_{j+1}, \dots, x_{j+2}, x_{i+1}, x_{j+1}, \dots, x_n\}$ を生成

図5 2-opt法

する。(図5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)

- 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。

b) ブロック型突然変異

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、逆にローカルミニマム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、ローカルミニマムから脱出するためにツアーの一部を破壊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 $tour_{par} = \{x_1, x_2, \dots, x_n\}$ は親個体, $tour_{chi}$ は子個体。
- ② $tour_{chi}$ からノード x_i をランダムに選択する。
- ③ ノード x_i とその近傍に含まれるノード $\{x_{i-r}, \dots, x_{i+r}\}$ を除いて $tour_{chi}$ にコピーする。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 $B \times (\text{デポから最も遠いノードまでの距離})$ の間の乱数で決定する。 B は問題依存知識により決定される定数である。
- ④ ノード集合 $\{x_{i-r}, \dots, x_{i+r}\}$ 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。

3.3 多段知能型GAの構成

3.2で紹介されたオペレータを導入することにより、以下のようなGAを提案する。

(1) 2-opt型GA:

この手法は図6のように、初期集団生成にランダムNI生成, NI型交叉, 2-opt型突然変異を用いることにより、迅速な解の収束を可能にする。

NI型交叉の計算量は $O(n^2)$ 。一方、2-opt型突然変異の計算量はずっと小さい。最悪の場合は $O(n^2)$ となるが、ランダムNI生成とNI型交叉により、既に高い最適化が行われている個体がほとんどなので、これはほとんど起こらない。このため、2-opt型突然変異の実行時間はNI型交叉のようにNI法を使ったオペレータより断然小さくなる。

図6 2-opt型GA

図7 ブロック型GA

(2) ブロック型GA:

この手法は図7のように構成され、挿入順制御NI生成を使うことにより、最適性の高い初期集団を生成し、ブロック型突然変異により局所最適化されたツアーの一部を再構築し、局所最適解に陥ることを防ぐ。

(3) 多段知能型GA(MIGA):

最後に、多段知能型GA(MIGA: Multi-step intelligent GA)を提案する。本手法は、2-opt型GAと、これに相補的なブロック型GAとを組み合わせることにより構成する、つまり、この2つのGAを独立に試行し、結果として良い方を採用する。これにより、様々な巡回拠点配置パターンに対して高い最適性と実行時間応答性の両方を実現する。

また、実行時間応答性を保証するため、指定時間制限内に実行可能なGAの世代数をオフラインに計算し、それぞれのGAの世代数を制限する(例:それぞれのGAで15ミリ秒実行する)。

4. 評価実験と結果

4.1 実験環境

本実験では、Intel Pentium II(450MHz)プロセッサと256MBのメモリを装備しているPCを実験に使用し、C言語によりプログラムを作成した。

ターゲットとしている配送ネットワークでは、1台のトラックが1回に巡回する拠点数はおよそ40を超えないので、40都市のTSPで各解法の評価を行った。現実には様々な拠点数の配置が発生するために、40都市をランダムに配置した20,000個の異なる巡回拠点配置パターンを用意した。そして、各解法の効果を評価するために、それぞれの解法で100回、テストパターンを解き、数10ミリ(30ミリ)秒以内での最適解との誤差が3%以内の解の出現頻度を調べた。

GAのパラメータは、集団個体数を100とし、交叉率と突然変異率はそれぞれ10%に設定した。ランダム生成と挿入順制御NI生成の混合では、両生成法による初期集団の50%ずつを混ぜ合わせて用いた。また、挿入順制御NI生成の閾値 w には、4段階の倍率(0.5, 1.0, 1.5, 制限なし)を用いた。これらのパラメータは、幾つかのパラメータセットによる予備実験の結果に基づいて決定した。

4.2 実験結果

表2に各方式で集団サイズを100とした場合の30ミリ秒以内で計算可能なGAの世代数を示す。

20,000個の異なる40都市TSPを100回ずつ実行した結果から、それぞれの手法における最適解の獲得率、専門家レベルの誤差率3%以内の解の獲得率を調べた。これを表3に示す。

表2 30 ミリ秒以内に実行可能な世代数

#	手法	初期化	突然変異	世代数
1	2-opt型	ランダムNI	2opt型	24
2	ブロック型	ランダム + 挿入順制御NI	ブロック型	20

表3 誤差率3%以内の解の割合

#	手法	最適解	誤差率3%以内
1	2-opt型GA	84.45%	99.885%
2	ブロック型GA	83.75%	99.785%
3	多段知能型GA	92.05%	100.0%

図 8 特定の拠点配置パターンにおけるツアー形状の違い

5. 評価

5.1 実験結果に基づく評価

表 3 は提案手法多段知能型GAの有効性を示している。これを見ると、40都市TSPにおいて、30ミリ秒以内に誤差率 3 %以内の解を 100%獲得できたのは多段知能型GAだけである。

(1) ブロック型GAの効果：

2-opt型GAによる実行結果から誤差率 3 %以上の解を解析したところ、図 8 (a) のようなギアホイール型のツアー形状をした解が多数見られた。人間の専門家は、図 8 (b) のような、より直線状のツアー形状を生成する。もし専門家が図 8 (a) のような非効率的なツアーを見つけたら、彼らはそのシステムを信頼できないものとして使うことをやめるだろう。

図 8 (a) のようなギアホイール型のツアー形状をした解が生成されやすい巡回拠点配置パターンにおいても、ブロック型GAを使った場合は、図 8 (b) のような直線状のツアー形状の解を得ることができるであろう。なぜなら、ギアホイール型のツアー形状が非効率だった場合はブロック型突然変異により改善される可能性が高いからである。事実、ギアホイール型の形状のツアーはブロック型GAによる解では見られなかった。

(2) 多段知能型GA(MIGA)の効果：

実験の結果によると、2-opt型GAの場合、20000テストケースの内43ケースで解の誤差率が 3 %以上であった。ブロック型GAの場合は、23ケースで解の誤差率が 3 %以上であった。しかしながら、多段知能型GAでは20000テストケースの全てで誤差率が 3 %以内であった。これは、多段知能型GAが2-opt型GAとブ

ロック型GAというタイプの違う、相補的特性を持ったGAの良い方の解を採用することにより、幅広い巡回拠点配置パターンに対応できるためである。これにより、技術目標であった実時間応答性(各ルート数10ミリ秒以内、1000ルートで数10秒)と専門家レベルの最適性(誤差率 3 %以内)を実現できた。

5.2 VRPへの適用

配送ネットワーク構築問題への本提案手法の有用性を評価するため、提案手法をパートサプライ・ロジスティクス・ネットワークの構築シミュレーションに適用する実験をおこなった。このネットワークでは、複数のトラックがデポを出発し、各トラックは担当する複数の部品メーカーを巡回し、荷物を収集してデポに戻る。もちろん、必要な時に、デポから工場に必要な荷物を輸送するが、これは本提案のTSP解法を適用する本構築シミュレーションの対象とはしない。この構築シミュレーションには、トラックの配送ルートの最適化のみならず、各トラックの担当エリアの配置の最適化が必要である。これらのトラックのエリアの配置と配送ルートの最適化問題はVRP(Vehicle Routing Problem) [Laporte1992]に分類される。

提案手法をVRPに適用するため、図 9 に示すように、1つの染色体が1つの配送者やトラックなどのツアーの代わりに複数のトラックのツアーを表現するように遺伝子型を拡張する。図 9 は、2台のトラックが、それぞれ6箇所と5箇所の合計11箇所のノードを巡回するツアーを表現した染色体の例である。

ここで、NI法はツアーに新しいノードを挿入する際、挿入された結果、最もツアー長が短くなる位置にノードを挿入する挿入法だが、TSPの場合、挿入先は1つのツアーのどこかであった。しかし、VRPでは、複数ツアーのどこか(同じ染色体なら他のツアーでも良い)に挿入された結果、各ツアーの距離の合計が最短となる位置(どのツアーの何番目か)に挿入する。

図 9 VRPの遺伝子型

表4 VRPシミュレーションの結果

#	条件 (荷物数)	トラックの 台数	全体ツアー長 (誤差率)
1	700	29	1015km (2.6%)
2	900	41	1345km (2.8%)
3	1000	48	1612km (2.8%)

本構築シミュレーションの実験では、工場を1箇所、デポを7箇所、部品メーカーを30箇所としたパーツサプライ・ロジスティクス・ネットワークを対象とした。

このネットワークは現場から実験用に取得できるデータの関係で小規模のものとなった。このため、本実験では拠点数40箇所程度のTSPを数10回解けば良いことになる。但し、この構築シミュレーションでは、各トラックの担当エリア、つまり巡回する部品メーカーの範囲(同じ染色体に入れるトラックに相当)を最適化するため、種々変更しながら試行錯誤する必要がある。そこで本実験では、10回程度のデータを入力して1回の実行で10回分まとめて試行錯誤ができるようにした。このため1回の実行(上述の通り40拠点TSPを数10回解く)を対話的応答時間、つまり2~3秒以下で行う必要がある。そこで、世代数を制限し、応答時間が1秒以下となるようにして実験した。これにより、数100のTSPを解く大規模配送ネットワーク構築と同様、1つのTSPを数10秒未満でかつ、全体として3%以下の誤差で解けるかを検証することになった。このためトラックの担当エリアの配置の最適化も同時に行う配送ネットワーク構築にも、この配置の最適化は手動で個別に行う大規模配送ネットワーク構築にも本提案法の有用性が示せたと考える。本構築シミュレーションは、表4のように荷物量以外、同じシミュレーション条件を持つ3ケースにおいて実施した。実験の結果、表4のトラックの数は最適であると実証された。さらに、全体のツアーの長さの最適解からの誤差率は3%以内であった。

したがって、提案したGA手法はTSPだけでなく、より一般的なVRPなどの問題にも有用な事が示された。つまり、これらの手法はTSPやVRPのような、距離空間上に定義された問題に有効である。さらに、ブロック型突然変異のコンセプトは距離の定義は持たないが近傍の定義を持つ位相空間上に定義された問題へと適用可能である。

6. 他手法との比較

TSPの解法は、これまでに様々なものが提案されて

いる。ここでは、提案解法と他の解法を比較する。

本節では、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を実験に使用し、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、比較対象となる他の解法は異なるスペックのマシンで実行されているため、そのマシンスペック差を考慮する必要がある。そのため、演算処理能力計測用のベンチマークテストとしてRC5-72によるベンチマークの統計[RC5-72]の結果を参考にして、スペック差修正係数を導出した。これは、他の解法の実験環境でのベンチマークテストの結果を、本実験で用いた実験環境でのベンチマークテストの結果で割ることにより得られる。これを他の解法の実行時間にかけることにより、本実験と同スペックの環境での推定実行時間を計算した。

厳密解法としては、Branch-and-cut法[Grotschel1991]やDynamic Programming(DP)を用いた解法[Bellman1962],[Bertsekas1987]が提案されている。しかし、これらの解法は最適解が得られる反面、多くの計算時間を要する。DP法を適用してユークリッド型TSPを多項式時間で非常に最適解を得る方式[Arora1998]も提案されている。しかし、著者らが適用対象としているアプリケーションに対しては、実行時間が長く、また、これらアルゴリズムは非常に複雑であり、通常のシステム開発者が実際のアプリケーションの様々な要求に合わせて修正することは困難である。

近似解法も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、LK法が有名であるが、LK法や、その改良方式[Lin1972],[Yamamoto1997]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やTS、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[Kirkpatrick1983][Ingber1993][Miki2003]は理論的には、局所解に陥る危険性を少なくして、非常に最適解に近い解を得ることができる。しかし、実用的にはクーリングスケジュールやコスト関数などSAの多数のパラメータを問題に合わせて決め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計算に時間を要する。

TS[Glover1989][Glover1990][Hooker1995]

[Fang2003]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[Kanazawa2004]。

ランダム性のある複数の初期解に、2-opt法やLK法などの局所探索法を適用して、得られた最良解を出力するランダムリスタート法[Kubota1999][Yanagiura2000]やGRASP[Feo1994]は繰り返し回数を制限することで応答性を保証できる解法である。しかし著者らの実験では、TSPにランダムリスタート法を適用した場合、どの問題に対しても誤差3%以内の精度を保証するためには、Intel Pentium II(450MHz)プロセッサ、メモリ256MBのマシンで500ミリ秒以上の処理時間を要しており、このスペック差修正係数は0.108なので、本実験環境でも50ミリ秒以上の処理時間を要する。

制約充足問題の解法であるMin-conflicts[Minton1990]やGSAT[Selman1992]も大規模な組合せ最適化問題に対して有効な手法であり、更に性能を向上させるために多くのアイデアが提案されている。しかしGSATをTSPに適用するには、TSPを制約として表さなければならない。局所的な条件の表現は容易であるが、ツアー自体以外に閉路を含まないというグローバルな条件の表現は複雑であり、更にその条件の判定にも処理時間を要する。

これに対して、提案方式は制限時間内での応答性能の保証とエキスパートレベルの最適性の両立を図ったものであり、大規模な配送ネットワークの最適化に有効な方式である。

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。

LK法やEdge Assembly Crossover(EAX)[Nagata2004]、Distance-preserving crossover(DPX)[Whitley1989]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[Tamaki1994],[Baraglia2001],[Tsai2004],[Nguyen2007]。EAXやDPXは親ツアーの枝の特質を厳密に子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、小規模なTSPに用いるのは効率的でない場合がある。文献[Baraglia2001]では300都市以上から10000都市TSPではCga-LKの方が優れているが198都市TSPではRandom-LKが優れているとの実験結果が示されている。つまり、1000都市以上のTSPを効率的に解ける解法が100都市程度のTSPでも効率的に解けるとは限らない。

本研究が対象とするような数十都市程度のTSPに対

しては、文献[Baraglia2001]においてTSPLIB問題lin105を解いている。しかし、平均誤差1.77%を実現するのに200-MHz PentiumPro PC running Linux 2.2.12で231秒を要している。このスペック差修正係数は0.048なので、本実験環境でも11.088秒を要する。また、文献[Cheng2002]でも、TSPLIB問題lin105での様々な交叉オペレータでの性能比較実験が行われているが、最も良い交叉オペレータでも平均誤差3.1%を実現するのにSUN SPARC Ultra-5 10 machineで平均750秒を要している。このスペック修正係数は0.065なので、本実験環境でも48.75秒を要する。それに対して、MIGA1では、1.11秒で最適解を得ている。MIGA2では、0.15秒で平均誤差0.31%を実現した。

本研究が対象とするような数10から数100程度のTSPを対象として、これを高速に解くためのGA手法としては、YanらによるGA手法[Yan2007]が提案されている。そこで、YanらによるGA手法、金澤らによるTS手法と提案手法である多段知能型GA(MIGA)の計算時間と解の最適性の比較を行うため、提案手法をTSPLIBのst70からa280までの9問題に対して、それぞれ10回解き、最適解からの誤差の平均と平均実行時間を求めた。実験は、2種類の終了条件で行った。MIGA1は、収束するまで実行した場合である。MIGA2は、その1/10の実行時間で終了した場合である。MIGA3は30ミリ秒で実行を終了した場合である。提案手法は、探索初期の早い収束特性を利用して、最適性は高くなくてもよいので、素早く近似解を導出する事を目的としているため、通常MIGA2やMIGA3の様な終了条件を用いる。

各実験環境のスペック差修正係数を表5に、MIGAと先行研究のTSPLIBのst70からa280までの9問題の実行結果を表6に示した。得られた近似解の最適性は、最適解からの誤差の平均により、%で表した。実行時間は括弧の中に秒単位で表記した。MIGA3については、最悪値も最適解からの誤差の平均により、%で表した。データの無い箇所は、表中に「-」を表記した。

まず、YanらによるGAの結果と比較する。st70の結果を見ると、YanらによるGAの20%程度の時間で、MIGAは、ほぼ同じ精度の解を得ている。eil76では、YanらによるGAの解の誤差が1.184%であるのに対して、最適解をほぼ同程度の時間で得ている。特にa280では、YanらによるGAの7%程度の時間で、約4%誤差の小さい解を得ている。

次に、金澤らによるTSの結果と比較する。MIGAの実行時間を見ると、pr107では、約19%、pr144では、約63%、pr152では、約45%、pr226では、約5%の

表5 スペック差修正係数

	Yanらによる GA	金澤らによる TS
実験環境	CPU : Pentium4 2.4GHz, メモリ:256MB	CPU : Pentium4 2.55GHz, メモリ : 1GB (DDR266)
スペック差 修正係数	0.519	0.590

表6 TSPLIB 問題による先行研究との比較結果

問題名	最適解からの誤差の平均[%] (平均実行時間[秒])					
	Yanらに よる GA	金澤らに よる TS	MIGA1	MIGA2	MIGA3	
					平均	最悪値
st70	0.312 (0.348)	-	0 (0.750)	0.370 (0.074)	0.400 (0.030)	1.333
eil76	1.184 (0.602)	-	0 (0.844)	1.914 (0.080)	2.193 (0.030)	2.974
kroA 100	0.016 (0.877)	-	0 (0.937)	0.176 (0.131)	0.873 (0.030)	1.588
pr 107	-	0.290 (0.826)	0.086 (0.985)	0.256 (0.156)	0.449 (0.030)	0.899
pr 136	0 (3.690)	0.190 (4.378)	0.624 (2.016)	1.067 (0.233)	2.460 (0.030)	3.481
pr 144	0 (4.136)	0.019 (4.685)	0 (2.625)	0.149 (0.284)	1.665 (0.030)	2.896
pr 152	-	0.120 (7.558)	0.153 (3.375)	0.551 (0.338)	1.175 (0.030)	2.77
pr 226	-	0.510 (12.685)	0 (3.125)	0.515 (0.652)	1.854 (0.030)	2.566
a280	10.770 (17.371)	-	3.180 (10.453)	6.572 (1.100)	-	-

金澤らによるTSの実行時間で、同精度の解を得ている。

一般的に、YanらによるGAや金澤らによるTSより、MIGAが効率的に探索できていることが実験結果から確認できた。

また、MIGA3(実行時間30ミリ秒)の結果を見ると、a280では、30ミリ秒で十分な世代を実行できなかったが、st70からpr226までの8問題では、平均誤差が1%前後で、最悪値でもおよそ3%以内に収まる事が確認できた。

7. おわりに

本論文では、TSPを解くための知能的GAを提案・評価し、これが繰り返し対話型のシミュレーションが求められる大規模な配送ネットワークの最適化へと適用可能であることを確認した。この種のアプリケーションには、最適性だけでなく、応答性も求められ、本論文が対象としている大規模な配送ネットワークの最適化では、1000ものTSPを数10秒以内に、かつ専門家レベルの最適性を確保するために誤差率3%以内で解かなければならなかった。

シンプルなヒューリスティックスである2-opt法を用いた2-opt型突然変異及び計算コストはかかるが高い最適化が可能なヒューリスティックスであるNI法を用いたランダムNI生成、NI型交叉により構成される2-opt型GAを提案した。2-opt型GAは迅速に収束するので、短時間で最適性の高い解を得ることができたが、ある特定の巡回拠点配置パターンでは、専門家レベルの最適性を保証できなかった。

そこで、2-opt型GAの弱点を克服するため、NI法と専門家の求解プロセスから作られた挿入順制御NI生成とランダム生成、局所最適解からの脱出を目的に作られたブロック型突然変異とNI型交叉から構成されるブロック型GAを提案した。この手法の計算コストは多少高いが、2-opt型GAの弱点パターンを克服することができた。

そして、様々な巡回拠点配置パターンで専門家レベルの最適性と実時間応答性を実現するため、特性の違う2つのGA、2-opt型GAとブロック型GAを組み合わせる多段知能型GA(MIGA)を提案した。

著者らの実験によると、2-opt型GAだけでは、20000テストケース中23ケースで3%以上の誤差率を出していたが、提案した多段知能型GA(MIGA)では20000全てのケースで数10ミリ秒内に専門家レベルの最適性(誤差率3%以内)の解を得ることができた。

これらの結果から提案した手法は大規模な配送ネットワークの対話的シミュレーションに必要な最適性と応答性でTSPを解くことができることが示された。

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02Q638として行ったものである。

参考文献

[Arora1998] S. Arora: "Polynomial Time Approximation Schemes for Euclidean TSP and Other Geometric

- Problems", *Journal of the ACM*, Vol.45, No.5, pp.753-782, (1998).
- [Baraglia2001] R. Baraglia, J.I. Hidalgo, R. Perego, : "A hybrid heuristic for the traveling salesman problem", *IEEE Transactions on Evolutionary Computation*, Vol.5, Issue 6, pp.613-622, (2001).
- [Bellman1962] R.E. Bellman, and S.E. Drefus, : "Applied Dynamic Programming", Princeton University Press, (1962).
- [Bertsekas1987] D.P. Bertsekas, : "Dynamic Programming: Deterministic and Stochastic Models", Englewood Cliffs, NJ Prentice-Hall, (1987).
- [Cheng2002] C. Cheng, W. Lee, K. Wong, : "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).
- [Davis1991] L. Davis, editor, : "Handbook of Genetic Algorithms", Van Nostrand Reinhold, N.Y., (1991).
- [Fang2003] Y. Fang, G. Liu, Y. He, Y. Qiu, : "Tabu search algorithm based on insertion method", *Proceedings of the 2003 International Conference on Neural Networks and Signal Processing*, pp.420-423, (2003).
- [Feo1994] T.A. Feo, M.G.C. Recende and S.H. Smith, : "A Greedy Randomized Adaptive Search Procedure for Maximum Independent Set", *Operations Research*, Vol.42, No.5, pp.860-878, (1994).
- [Glover1989] F. Glover, : "Tabu Search - Part I", *ORSA Journal on Computing*, 1, pp.190-206, (1989).
- [Glover1990] F. Glover, : "Tabu Search - Part II", *ORSA Journal on Computing*, 2, pp.4-32, (1990).
- [Grotschel1991] M. Grotschel, and O. Holland, : "Solution of large-scale symmetric traveling salesman problems", *Mathematical Programming*, Vol.51, pp.141-202, (1991).
- [Hooker1995] J.H. Hooker, and N.R. Natraj, : "Solving a General Routing and Scheduling Problem by Chain Decomposition and Tabu Search", *Transportation Science*, Vol.29, No.1, pp.30-44, (1995).
- [Ibaraki1993] 茨城俊秀, "離散最適化とアルゴリズム", 岩波書店, (1993).
- [Ingber1993] L. Ingber, : "Simulated Annealing Practice versus Theory", *Journal of Math. Comput. and Modelling*, Vol.18, No.11, pp.29-57, (1993).
- [Kanazawa2004] 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).
- [Kirkpatrick1983] S. Kirkpatrick and C.D. Gelatt and M.P. Vecchi, : "Optimization by Simulated Annealing", *Science*, Vol.220, Num.4598, pp.671-680, (1983).
- [Kolen1994] A. Kolen, and E. Pesch, : "Genetic Local Search in Combinatorial Optimization", *Discrete Applied Mathematics*, Vol.48, pp.273-284, (1994).
- [Kubota1999] S. Kubota T. Onoyama K. Onayagi and S. Tsuruta, : "Traveling Salesman Problem Solving Method fit for Interactive Repetitive Simulation of Large-scale Distribution Network", *Proc. IEEE SMC'99*, pp.533-538, (1999).
- [Laporte1992] G. Laporte, : "The Vehicle Routing Problem: An overview of exact and approximate algorithms", *European Journal of Operational Research*, Vol.59, pp.345-358, (1992).
- [Lawer1985] E. Lawer, J. Lenstra, K. Rinnoy, and D. Shmoys, : "The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization", John Wiley Sons, (1985).
- [Lin1972] S. Lin, and B.W. Kernighan, : "An effective heuristic algorithm for the traveling salesman problem", *Operations Research*, Vol.21, No.2, pp.498-516, (1972).
- [Miki2003] M. Miki, T. Hiroyasu, T. Jitta, : "Adaptive Simulated Annealing for maximum temperature", *Proc. of IEEE International Conference Systems, Man and Cybernetics 2003*, vol.1, pp.20-25, (2003).
- [Minton1990] S. Minton, M.D. Johnston, A.B. Philips, and P. Laired, : "Solving Large Scale Constraint Satisfaction and Scheduling Problem Using Heuristic Repair Method", *Proc. AAAI'90*, pp.17-24, (1990).
- [Nagata2004] Y. Nagata, : "The EAX algorithm considering diversity loss", *Proc. of The 8th International Conference on Parallel Problem Solving from Nature-PPSN VIII*, Birmingham, UK, Springer-Verlag, pp.332-341, (2004).
- [Nguyen2007] H.D. Nguyen, I. Yoshihara, K. Yamamori, M. Yasunaga, : "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).
- [Onoyama2000] T. Onoyama, S. Kubota, K. Oyanagi, and S. Tsuruta, : "A Method for Solving Nested Combinatorial Optimization Problems", *Proc. IEEE SMC2000*, pp.340-345, (2000).
- [RC5-72] RC5-72 によるベンチマークの統計, <http://www.orange.co.jp/~masaki/rc572/>
- [Selman1992] B. Selman, H. Levesque, and D. Mitchell, : "A New Method for Solving Hard Satisfiability Problems", *Proc. AAAI'92*, pp.440-446, (1992).
- [Tamaki1994] H. Tamaki et al. : "A comparison study of genetic codings for the traveling salesman problem", *Proceedings of the 1st IEEE Conference on Evolutionary Computation*, pp.1-6, (1994).
- [Tsai2004] H. Tsai, J. Yang, Y. Tsai, and C. Kao, : "An evolutionary algorithm for large traveling salesman problems", *IEEE Transactions on Systems, Man and Cybernetics, Part.B*, Vol.34, Issue.4, pp.1718-1729, (2004).
- [Whiley1989] D. Whiley, T. Starkweather, and D. Fuquay, : "Scheduling problems and traveling salesmen: The genetic edge recombination operator", *Proceedings of the 3rd International Conference on Genetic Algorithms and their Applications*, pp.133-140, (1989).
- [Yamamoto1997] 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).
- [Yan2007] X. Yan, H. Liu, J. Yan, Q. Wu, : "A Fast Evolutionary Algorithm for Traveling Salesman Problem", *Proc. of Third International Conference on Natural Computation (ICNC 2007)*, Vol.4, pp.85-90,

(2007.8).

[Yanagiura2000] M. Yanagiura, and T. Ibaraki, "On Metaheuristic Algorithms for Combinatorial Optimization Problems", Transactions of the IEICE, Vol.J85-D-II, No.8, pp.3-25, (2000).

(2007年5月22日 受付)

(2008年4月7日 採録)

[問い合わせ先]

〒270-1382 千葉県印西市武西学園台2-1200

東京電機大学情報環境学部

櫻井 義尚

TEL: 0476-46-9081

FAX: 0476-46-8449

E-mail: ysakurai@sie.dendai.ac.jp

著者紹介

さくらい よしたか
櫻井 義尚 [正会員]

2000年電気通信大学電気通信学部電子情報学科卒業。2002年同大学院電気通信学研究科電子情報学専攻博士前期課程修了。2005年同博士後期課程単位取得済み退学。同年博士(工学)。2005年4月東京電機大学情報環境学部情報環境学科助手。進化・学習アルゴリズムの理論及び応用に関する研究に従事。日本知能情報フェジィ学会、情報処理学会、IEEE各正会員。

おの やま たかし
小野山 隆 [非会員]

1981年3月大阪大学理学部数学科卒業。同年日立ソフトウェアエンジニアリング(株)入社。同社技術開発本部研究部に勤務。物流システム等の研究に従事。博士(情報科学)。情報処理学会、IEEE各会員。

くぼた せん
久保田 仙 [非会員]

1999年3月東京大学工学部電気電子情報工学科卒業。同年日立ソフトウェアエンジニアリング(株)入社。同社技術開発本部研究部に勤務。画像処理、物流システム等の研究に従事。情報処理学会会員。

つるた せつお
鶴田 節夫 [非会員]

1971年3月早稲田大学工学部電気通信学科卒業。1973年3月名古屋大学大学院電気・電子・電機第2工学研究科修士課程修了。同年日立製作所入社。2003年4月より東京電機大学情報環境学部情報環境学科教授。工学博士。知的システム等の研究に従事。情報処理学会、IEEE各正会員。

A Multi-step Intelligent Genetic Algorithm to Optimize Delivery Problem within Interactive-time

by

Yoshitaka SAKURAI, Takashi ONOYAMA, Sen KUBOTA and Setsuo TSURUTA

Abstract :

To optimize large-scale distribution networks, solving about 1000 (around 40 cities) TSPs (Traveling Salesman Problems) within an interactive length of time (max. scores of seconds) is required. Yet, expert-level (less than 3 % of errors) accuracy is necessary. To realize the above requirements, a Multi-step intelligent GA method was developed. This method combines a high-speed GA with an intelligent GA holding problem-oriented knowledge that is effective for some special location patterns. If conventional methods were applied, solutions for more than 20 out of 20,000 cases were below expert-level accuracy. However, the developed method could solve all of 20,000 cases at expert-level.

Keywords : TSP (Traveling Salesman Problem), GA (Genetic Algorithm), Heuristics

Contact Address : **Yoshitaka SAKURAI**

*Department of Information Environment, School of Information Environment, Tokyo Denki University
2-1200 Muzai Gakuendai Inzai, Chiba, 270-1382, JAPAN*

TEL : 0476-46-9081

FAX : 0476-46-8449

E-mail : ysakurai@sie.dendai.ac.jp