

制約付き TSP を解くための局所利己的遺伝子許容動的制御 GA

櫻井 義尚[†] 小野山 隆^{††} 久保田 仙^{††}
中村 嘉宏[†] 鶴田 節夫[†]

リアルタイムに配送の効率化を図る配送ルート最適化システムの実現には、時間などの制約がある数十から百都市前後の規模の巡回セールスマン問題 (TSP) を対話応答時間である数秒以内に解き、しかも専門家レベルの最適解を得ることが要求される。この要求を満たすために、局所利己的遺伝子許容型 GA (Locally Selfish-gene Tolerant Genetic Algorithms, LST-GA) に加え、これに動的制御の概念を取り入れた局所利己的遺伝子許容動的制御 GA (Locally Selfish-gene Tolerant Dynamic Control Genetic Algorithms, LSTDC-GA) を提案した。この手法では、1 つの個体の染色体を構成する個々の遺伝子は同じ個体内の他の遺伝子の制約を無視して局所的利己的にその遺伝子の制約だけを満たす。こうして、制約違反を起こした個体がある程度許容し、改善の機会を与える。またこの許容度合いであるペナルティ係数を環境変数である突然変異率などと同期させて動的に制御することにより進化を促進させる。時間制約が存在する数十から百都市前後の規模の TSP において最大誤差が 20% 以下の解が、実験では 1 秒以下で求まることを確認した。

Locally Selfish-gene Tolerant Dynamic Control GA to Solve Constraint TSPs

YOSHITAKA SAKURAI,[†] TAKASHI ONOYAMA,^{††} SEN KUBOTA,^{††}
YOSHIHIRO NAKAMURA[†] and SETSUO TSURUTA[†]

Achievement of delivery route optimization system that improves the delivery efficiency in real time requires to solve time-constraint 30-100 cities Traveling Salesman Problems (TSP) within interactive response time, with practicable optimality. To meet this requirement, a Locally Selfish-gene Tolerant Dynamic Control GA is proposed. Here, each gene of an individual satisfies only its constraints selfishly, disregarding the constraints of other genes in the same individual. Further, to some extent, even individuals that violate constraints can survive over several generations and are given the chance of improvement. Moreover, evolution is promoted by dynamically changing the degree of the tolerance and GA operations. Our experiment proved that this method provides expert-level solutions for 30-100 cities time constraint TSPs within one second.

1. はじめに

物流の効率化は経済面からだけでなく、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る。そのため、コンピュータによる配送順路の最適化が必要になる。本論文では、書簡を含む各荷物の配送先到着時刻が指定されている場合、それらの配送先を巡回する順番、したがって時刻を最適化するような問題を対象とする。

これらの配送では、毎日あるいは毎回、宛先の異なる

配送要求が行われる。したがって、そのたびに異なった配送ルートの作成とその最適化が必要になる。その労力を軽減させるためにもコンピュータの自動求解によるサポートが有用である。

このような複数の配送先への順路の最適化問題は TSP (Traveling Salesman Problem) として定式化され、様々な研究が行われている。本論文で対象とする荷物や書簡の配送問題では、巡回箇所規模からして、最適な配送ルートの計算は時間制約のある数十から百都市前後の規模の TSP を解くことと同じと考えられる。時間制約とは各配送先への到着時間指定のことである。

時間制約付き配送問題の解法としては、すでに整数計画法 (Integer Programming)¹⁹⁾ に基づいた解法が提案されている。この解法は高精度ではあるが、計

[†] 東京電機大学情報環境学部

Department of Information Environment, Tokyo Denki University

^{††} 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.

算時間が長い．また，タブーサーチ (tabu-search)⁴⁾ や Simulated Annealing (SA 法)¹⁷⁾ を利用した解法も計算時間が長いという問題がある．遺伝的アルゴリズム (Genetic algorithm, GA)²⁾ はそれを解決する 1 つの方法であり，TSP を解決するために多くの種類の GA が提案されている^{11),13)}．しかし，これらも十分な即応性，つまり以下に述べる対話応答性を持っているとはいえない．

さて，このような最適化手法を用いて，実際の現場において有用な解を得るためには，走行距離や走行時間などの評価項目に対する適切な重み付けが必要になる．しかし，適切な重み付けをするのは難しく，経験上どうしても試行錯誤が必要となる．さらに，評価項目として定式化が難しい社会的・人間的な条件が絡む．そこで，実用可能な解がどうかを人間が総合的にチェックし，必要に応じ解の簡単な修正あるいは代替案の選択をすることが必要となる．このため，対話応答性が実用上不可欠となる．

許容できる対話応答時間は数秒，時間的に切迫した現場では基本的に 1~2 秒以下である．このため時間枠制約付き TSP に対して 1~2 秒以内の実時間応答性つまり対話応答性と誤差 (求められた解と最適解の走行距離における差を最適解の走行距離で割ることにより得られる相対誤差) 20% 未満の専門家レベルの高い最適性の両方を満足する近似解法が必要となる．

そこで，以上の条件を満足する GA の近似解法を研究してきた⁷⁾．本論文では，局所利己的遺伝子許容型 GA (Locally Selfish-gene Tolerant Genetic Algorithms, LST-GA) に加え，局所利己的遺伝子許容動的制御 GA (Locally Selfish-gene Tolerant Dynamic Control Genetic Algorithms, LSTDC-GA) を提案する．LST-GA は局所利己的遺伝子と許容の概念を用いた GA による近似解法である．一方，LSTDC-GA は LST-GA に動的制御 (Dynamic Control) の概念を取り入れることにより，これを改良したものである．

以下，2 章では時間枠制約がある状況での配送スケジューリングの要求とその技術的課題を明らかにし，3 章で，上記の必要条件を満たす本提案の解法を示す．4 章では，この解法の性能を検証する実験結果を示し，5 章でその有効性について議論する．

2. 巡回ルート最適化システム

2.1 巡回ルート最適化問題

本論文で対象としている問題は，時間指定のある荷物や書簡などの配送を行うものである．配送は，たとえばバイクなど小型車両により 1 日に 1~3 回程度行

われる．1 台の車両による 1 回に配送する区域の広さは，地域により異なるが，都市部では，配送先は比較的密集しており，これらの車両で 100 力所程度は巡回できる範囲に区切られている．すなわち配送先の数は，地域や時間帯によって異なるが，多くの場合，数十から百程度となる¹⁴⁾．

このような巡回ルートの最適化問題は後述で定式化するように TSP，結局は中国人郵便配達問題に対応し，これを解決する本提案手法の適用範囲は大きい．また，配送事業の規制緩和にともない，この分野の競争激化があり，本提案手法の適用範囲はさらに大きくなると考える．

時間指定は開始時刻と終了時刻の形で与えられる．これを時間枠制約と呼ぶと，時間枠制約のある配送先を巡回する際，早く着いてしまう場合に待機を許すか，待機を許さないかの 2 つの場合が考えられる．本論文で提案する手法自体は，どちらの場合にも適用可能であるが，このうち待機なしの場合のみを本論文では扱う．また，所要時間 (人件費など) と走行距離 (ガソリン使用量など) はトレードオフ関係となる場合があるが，ここでは走行距離の最小化を目的とする．つまり，時間枠制約を満たしつつ，走行距離を最小化する．

さらに，本問題は評価項目として定式化が難しい社会的・人間的な条件が絡む．そこで，コンピュータの解が実用可能な解かを人間が総合的にチェックし，必要に応じ簡単な修正あるいは代替案の選択をすることが必要となる．このため，コンピュータによるこの問題の求解には数秒以下の対話 (可能な) 応答性が要求される．

次節でこの問題の定式化を行う．

2.2 時間枠制約付き TSP への定式化

本論文で対象とする巡回ルート最適化問題を定式化すると以下ようになる．

配送網として重み付き完全グラフ $G = (V, E, w)$ を与える．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送先を表す． 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 地点間距離であり $d_{ij} = d_{ji}$ とする．

ノード v_{i-1} からノード v_i への移動時間を t_i とし，車両の速度を R とすると， t_i は d_{i-1}/R で算出され，ノード v_i への到着時刻は $\sum_{k=1}^i t_k$ と表される． l_i をノード v_i の時間枠制約の下限， u_i をノード v_i の時間枠制約の上限とすると，時間枠制約は以下のように

表される．

$$l_i \leq \sum_{k=1}^i t_k \leq u_i, \forall i \in N \quad (\text{条件 1})$$

条件 1 を満たすようなグラフ $G = (V, E, w)$ 上で最も短い巡回路長を持つハミルトン閉路を求める問題を時間制約付き TSP と呼び、この解を最適解、条件 1 を満たすハミルトン閉路を実行可能解と呼ぶ．

2.3 技術的課題

巡回ルート最適化システムを実用的なレベルで実現するためには、以下の技術的課題を解決する必要がある．

2.3.1 最適性と時間制約充足

数種類の現場で、その管理者も含め合計十数人の専門家に調査させたところ、時間制約が厳しい場合、人間の専門家が導き出したルートでさえ、最適解からおよそ 50% の誤差が生じることがある．また、時間制約がある場合でも、最適解からの誤差が 10 ~ 20% を超えると、最適解と比べた時に、遠回りが目立つ．これを解決するため、10 ~ 20% 未満の最大誤差とその半分以下の平均誤差の保証が要求される．

2.3.2 応答性

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車両の配送ルートを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応答性が求められる．上記現場の管理者やエキスパートの意見では、作業を効率的に進めるために応答時間が数秒以内であることが求められる．つまり、本論文で対象とする巡回ルート最適化問題では、数十から百都市前後の規模の時間制約付き TSP を数秒以内に解く必要がある．

具体的には、配送要求があってから配送に出るまでに 20 ~ 30 分以下で用意をする必要がある．しかし、コンピュータシステムを用いる場合、配送箇所や住所の入力に 3 ~ 10 分、コンピュータが配送ルートを提案・表示し、人間がそれをチェック・確認し、承認した後、印刷に 1 ~ 2 分、配送物を提案順に並べ替えるのに 3 ~ 10 分を要する．つまり、提案ルートの巡回イメージや妥当性の確認（回りやすさの確認や代替案の選択・確認も含む）とその承認は数秒、修正などがあっても数十秒で行う必要がある．このため、コンピュータによる解の提案と表示には、時間の切迫する現場での対話レベルの実時間応答性（1 ~ 2 秒以内）が重要になる．

3. 局所利己的遺伝子許容動的制御 GA

3.1 解法のコセプト

2 章で説明した ① 専門家レベルの最適性、② 対話応答性の 2 つの技術的課題を満たす時間制約付き TSP の解法として、局所利己的遺伝子許容型 GA (Locally Selfish-gene Tolerant Genetic Algorithms, LST-GA) に加え、これに動的制御の概念を組み込んだ局所利己的遺伝子許容動的制御 GA (Locally Selfish-gene Tolerant Dynamic Control Genetic Algorithms, LSTDC-GA) を提案する．

GA 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、まず、初期集団生成により解の候補となる複数の個体からなる集団を生成する．そして、その集団をもとに交叉、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．

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つ．その遺伝子型は図 1 のように巡回順にノード番号（配送先の ID 番号）を並べた構造になっている．各遺伝子は配送先の ID 番号を示し、問題により配送先（の荷物）ごとに時間制約が与えられる．

LST-GA は、制約を満たす解を効率的に探索するために、GA に以下に示す 2 つの概念を適用したものである．

1 つ目の概念は、「局所利己的遺伝子 (Locally Selfish-gene)」である．これは、「染色体を構成する各遺伝子が、自己やその周辺との関係だけを局所的に最適化しようとするあまり、他の遺伝子や染色体全体のことを考慮せず、利己的に振る舞うこともある」という新しい考え方である．Dawkins の主張した利己的遺伝子³⁾は、各個体が利己的であるとあるというものであった．局所利己的遺伝子はこれとは異なり、より局所的に染色体内の各遺伝子が利己的な振舞いを見せるというものである．

2 つ目の概念は「許容 (Tolerant)」である．これは、「利己的遺伝子によって生成された個体のように

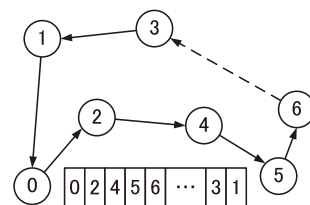


図 1 TSP における遺伝子型
Fig. 1 Genetic type of TSP.

制約を満たしていない不完全な個体が再生産されること、すなわち次世代に生き延びることをある程度は許すが、反省や適当な世代ごとの厳罰により結局は目的や制約を満たしている個体のみを再生産つまり繁栄を許す」という考えである。

GA による最適化では各遺伝子に制約がある場合、制約を満たさない個体をすべて致死個体とすると、新しい個体が集団に加わることが難しくなる。そのため集団の多様性が低下し、局所最適解に陥るおそれが大きくなる。制約が厳しいときは最悪の場合、集団が死滅するなど最適化計算自体ができないこともある。

そこで LST-GA では、各遺伝子は、その遺伝子に直接関連する局所的な制約だけの充足を図り、大域的に他の遺伝子の制約まで充足することは考慮しない。そしてこのような大域的制約を満たさない個体がある程度の世代数を生き延びることを許し、その間に各構成遺伝子の構成順序を変えるなどの処理により大域的な制約充足の機会を与える。そして、特定の世代において大域的制約を充足していない個体を淘汰する。こうして、各構成遺伝子の利己性を認め、つまり局所的な制約充足だけを図り、特定の世代以外は大域的な制約充足チェックを行わないことで、GA 処理の高速化を図り、様々なタイプの個体が生き残ることを許容する。これにより、局所最適への落ち込みを回避し、より最適かつ大域的に制約を満足させる可能性を高め、高い最適性と迅速な応答性能の両立を図っている。

より安定的に最適解を求めることを目的として、これに「動的制御 (Dynamic Control)」の概念を取り入れた手法が局所利己的遺伝子許容動的制御 GA (Locally Selfish-gene Tolerant Dynamic Control Genetic Algorithms, LSTDC-GA) である。

動的制御とは、「環境を動的に変えることにより、平衡、つまりこう着状態にある進化を推し進める」というものである。これによって、局所最適解へ陥ることを防止し、進化の足踏み状態を短くすることにより、無駄な探索を省き、探索を効率化し、探索時間を減らせる。

上記の概念を実現し、時間制約付き TSP を対話応答時間内に高精度に解く本提案の各要素の詳細とそれを組み合わせた具体的解法を以下に示す。

3.2 局所利己的遺伝子による制約充足

時間制約付き TSP を解くには、時間制約を充足している解空間を探索する必要がある。しかし、完全に制約条件を満たす解の生成は計算時間がかかるだけでなく、局所最適解に陥る可能性が高い。そこで、局所利己的遺伝子の概念に基づいて、すべての制約を

充足する保証はないが、制約を充足する方へと解の改善を行う仕組みを提案する。

本提案の解法は次に示す局所利己的 NI 法を用いた、初期集団生成、交叉、突然変異オペレータにより実現する。

3.2.1 局所利己的 NI 法

TSP のための近似解法として、我々は交叉と突然変異に関する遺伝子操作の際に、NI (Nearest Insertion) ヒューリスティクス¹⁷⁾ を求解効率を高めるように独自の工夫をして埋め込んだ GA⁸⁾ を提案し、これが高精度かつ高速な解法であることを示した。上記の概念を実現するために、この手法をさらに改良した「局所利己的 NI 法」を提案する。これはノードの部分ツアーへの NI 法での挿入を繰り返してツアー全体を構成するが、時間制約のあるノードをツアーに挿入するときには、そのノードの時間制約だけを満足する位置に NI 法で挿入する。つまり、このノードの挿入による、ツアー中の既存ノードへの影響は考慮しないので利己的である。また、他のツアーつまり交叉などを行う他の個体や個体の集団全体のことまでは考えないので局所的である。

部分ツアー $\{x_1, x_2, \dots, x_n\}$ にノード x_i を挿入する手順を以下に示す。部分ツアーは出発から到着までを取り扱うが、ツアー中には部分的なノードしか含まれない。これに残りのノードを挿入し、長くしていくことにより全体のツアーが得られる。

- (1) 部分ツアー $\{x_1, x_2, \dots, x_n\}$ の先頭から順にノード x_i を各ノードの間に挿入し、ノード x_i の時間制約を満たすかどうかをチェックしていく。制約を満たすまで、これを続ける。
- (2) 制約を満たしたら、引き続き各ノードの間にノード x_i を挿入していき、挿入による距離の増分を調べる。
- (3) 制約を満たさなくなったら探索を終了し、挿入により制約を満たし、しかも、これまでの探索において挿入による距離の増分の最も少ないノード間にノード x_i を挿入する。

局所利己的 NI 法では、大域的制約条件を満たすことを目的とせず、挿入するノードの制約条件しか考慮しないため計算コストが低い。一方、大域的制約条件を優先し、すべてのノードの制約条件が充足されるように他の全ノードの制約条件を考慮する挿入法を「大域利他的 NI 法」と呼ぶ。これは、すべての制約条件を満たす解を生成できるが、計算コストが高い。また、ノードを挿入できる箇所が見つからなかった場合、生成された子の遺伝子は致死遺伝子として淘汰される。

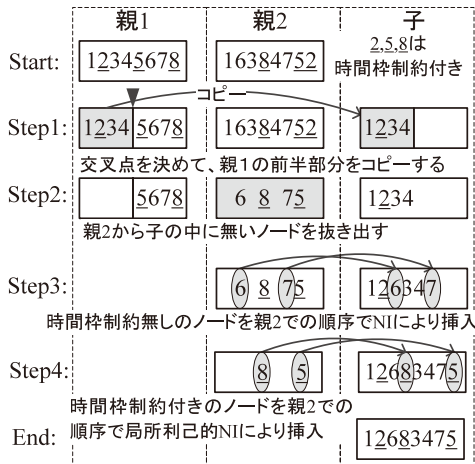


図 2 局所利己的 NI 法を用いた交叉
Fig. 2 Crossover using Locally Selfish NI method.

3.2.2 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。最適性の高い解を迅速に得るためには、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。そこで、「ランダム局所利己的 NI 生成」を提案する。これは最初は空である部分ツアーに局所利己的 NI 法を適用し、すべてのノードがツアーに含まれたら停止する。この方法の適用時のノードの追加順序はランダムに決定され、これにより得られたツアーを 1 個体とする。この過程を集団の大きさが指定値に達するまで繰り返す。

一方、「大域利他的 NI 法」を適用し、制約充足解のみを生成する生成法を「ランダム大域利他的 NI 生成」と呼ぶ。

3.2.3 交 叉

我々は TSP を効率的に解く GA のオペレータとして、親の優秀な特徴を継承しつつ迅速な解の収束を実現するために NI 法を用いた交叉オペレータ⁸⁾を提案している。ここでは、このオペレータに「局所利己的 NI 法」を適用することにより、効率的に時間枠制約に対処可能なオペレータ「局所利己的 NI 型交叉」を提案する。

拡張された交叉オペレータは以下の手順からなる (図 2 参照)。

- ① $tour_{par1} = \{x_1, x_2, \dots, x_n\}$ と $tour_{par2} = \{y_1, y_2, \dots, y_n\}$ を親ツアー、 $tour_{chi}$ を子ツアーとする。
- ② $tour_{par1}$ の中から交叉点 x_i を決める。

- ③ $tour_{par1}$ の交叉点より前の部分ツアー $\{x_1, x_2, \dots, x_i\}$ を $tour_{chi}$ にコピーする。
- ④ $tour_{chi}$ に含まれているノードを $tour_{par2}$ から削除する。
- ⑤ $tour_{par2}$ の残りのノードのうち、時間枠制約のないノードを NI 法により $tour_{chi}$ に先頭から順に挿入する。
- ⑥ $tour_{par2}$ の残りのノードのうち、時間枠制約のあるノードを局所利己的 NI 法により $tour_{chi}$ に先頭から順に挿入する。

一方、「大域利他的 NI 法」を適用し、制約充足解のみを探索する交叉オペレータを「大域利他的 NI 交叉」と呼ぶ。

3.2.4 突然変異

GA における突然変異は、局所最適解から脱出するのに役に立つ。しかし、局所探索や問題固有の知識を導入しないランダム過程のみを用いる単純な突然変異は効率的ではない。

そこで、我々はブロック型突然変異オペレータ⁸⁾を提案した。これに局所利己的 NI 法を適用することにより「局所利己的 NI によるブロック型突然変異」を提案する。これは以下の手順からなる。

- ① $tour_{par} = \{x_1, x_2, \dots, x_n\}$ を親ツアー、 $tour_{chi}$ を子ツアーとする。
- ② ノード x_i をランダムに親ツアー $tour_{par}$ から選択する。
- ③ ノード x_i とその周辺のノード $\{x_{i-p}, \dots, x_{i+p}\}$ を除いた部分ツアーを子 $tour_{chi}$ にコピーする。周辺の大きさ p は、たとえば、0 から $C \times$ (デポから最も遠いノードへの距離) の間のランダムな数のように決められる。 C は問題固有の知識により決められる定数。
- ④ $\{x_{i-p}, \dots, x_{i+p}\}$ の内で時間枠制約のないノードを NI 法により子ツアー $tour_{chi}$ に挿入する。
- ⑤ 残りの時間枠制約のあるノードを局所利己的 NI 法により子ツアー $tour_{chi}$ に挿入する。

一方、「大域利他的 NI 法」により拡張し、制約充足解のみを探索する突然変異オペレータを「大域利他的 NI によるブロック型突然変異」と呼ぶ。

3.3 許容による制約緩和

許容の概念に基づいた評価、選択、を以下のように提案する。

3.3.1 ペナルティによる評価

各個体の評価 (適応度) は、ツアー長と制約違反度合いから計算される。ツアー長が L 、制約違反度合いが T の個体の実質評価は $V = L + P \times T$ とな

る．制約違反度合い T は到着時刻と制約時刻の上限値と，下限値の近い方との差の絶対値である． P はペナルティ係数であり，制約違反度合いによる評価の減点割合を決めるものである．この実質評価をスケールリングすることにより適応度が計算される．その世代の集団の中での最小の V を V_{Min} とするとき，適応度は $F = V_{Min}/V$ となる．つまり，ツアー長が短く，制約違反が少ない個体ほど適応度は高くなる．

3.3.2 許容型選択

先に説明した局所利己的 NI 法を用いた交差や突然変異を用いると，集団中の個体は時間枠制約をすべて充足しているとは限らない．そのため，このままでは淘汰されてしまう可能性が高い．したがって，各世代では，通常，適応度のみによる選択により制約違反遺伝子が生き延びる可能性を残す．そして，数世代ごとに制約違反チェック付き選択を行う「許容型選択」により，制約違反遺伝子を淘汰する．制約違反チェック付き選択とは以下 b) で詳述するが，個体内の全遺伝子に対して制約違反のチェックと修正を行い，修正できない個体は淘汰する選択 (Selection) 方式である．

a) 適応度のみによる選択

同一個体内の他の遺伝子が時間枠制約を満たすかどうかはチェックせず，結果的に制約違反を起こしても，それに対するペナルティを考慮した適応度により次世代に選択される個体を選ぶ．

b) 制約違反チェック付き選択

すべての遺伝子が時間枠制約を満たすかどうかチェックする．そして，時間枠制約違反を起こした各個体で時間枠制約違反を起こしたノードを取り除き，局所利己的 NI 法によってノードを再挿入することでツアーを修正する．この修正作業を行っても時間枠制約を満たすことができなかつた個体は淘汰される．

もし時間枠制約が世代ごとにチェックされて，制約違反が修正されてしまうと，進化の可能性は減少し，局所解に陥る可能性は逆に増えてしまう．さらに悪いことに，頻繁な時間枠制約チェックと修正は多くの時間を要する（これは，対話応答性の確保において不利である）．したがって，上記のチェックと修正は指定された世代だけに制限する．しかし，このように明示的に修正しチェックしない世代においても交叉と突然変異を通して局所利己的 NI 法を使用するため，時間枠制約を満たさない個体を改善し，制約違反を解消できる可能性がある．したがって，この方法によりチェックの際のオーバーヘッドを少なくし，対話応答性を確保しながら，より良い最適化を実現できる．

3.4 動的制御による進化の促進

以上に述べた手法では，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迅速に収束して解を求めることができる．しかしその反面，局所最適解へ収束してしまうというリスクもある．局所最適解へ収束するのを防ぐためには，新しい個体が集団内に生まれ続ける必要がある．

評価におけるペナルティ値，選択における制約違反チェックの間隔は制約違反に対する許容度と考えることができる．そしてこの許容度は新しい個体の生成に大きく関わっている．

許容度が大きい場合は制約違反をあまり考慮しないため，目的関数の最適化が進む．一方，許容度が小さい場合は，制約違反に対する罰則が厳しいため制約充足解での探索が進む．しかし，制約の厳しい問題ほど新しい個体が集団に加わり難くなり，局所最適解に収束していく可能性が高い．

そこで，最初は許容度を大きくとることにより目的関数の最適化を進め，許容度を下げていくことにより徐々に制約違反遺伝子を排除し，制約を充足しつつ目的関数の最適化を進める．また，これを繰り返すことにより，局所最適解へ陥ることを防止する．

また，突然変異率を上げると現在の個体とは違う個体生まれやすくなり，局所最適解からの脱出効果が見込まれる．そこで許容度 (ペナルティ係数) と突然変異率を同期して変化させる方式を以下のように提案する．

3.4.1 周期的変化

突然変異率と許容度を周期的に上下させる．ペナルティ係数が低い状態から世代ごとに指数関数的にその値を増やしていく．そして，決まった周期でまた値を下降させる．これと同期して突然変異率を一時的に上昇させる．以上の操作を繰り返す．

3.4.2 新入り個体数をトリガとした変化

交叉や突然変異などのオペレータにより生成された子個体の集団のうち，次世代へと選択された個体をその世代における新入り個体，その数を新入り個体数と呼ぶ．突然変異率と許容度をこの新入り個体数により上下させる．ペナルティ係数が低い状態から世代ごとに指数関数的にその値を増やしていく．そして新入り個体数がある一定の値を下回ったときに値を下降させる．これと同期して突然変異率を一時的に上昇させる．以上の操作を繰り返す．

3.5 具体的解法の提案

以上に提案した各要素方式に基づいて次の 3 つの方法を定義する．すなわち，LST-GA，LSTDC-GA な

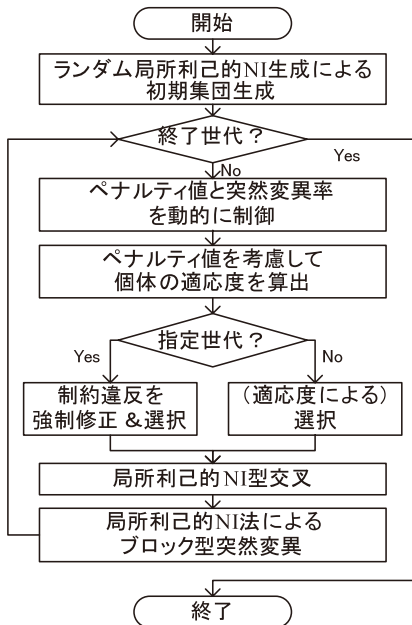


図 3 局所利己的遺伝子許容動的制御 GA (LSTDC-GA)

Fig. 3 LSTDC-GA.

ど本提案の具体的解法を示す。

3.5.1 大域利他的遺伝子型 GA (GA-GA)

大域利他的遺伝子型 GA (Global Altruistic-gene Genetic Algorithms; GA-GA) は、大域利他的 NI 法を用いたオペレータを用いることにより、制約充足解のみを探索する。

3.5.2 局所利己的遺伝子許容型 GA (LST-GA)

局所利己的 NI 法を交叉とブロック型突然変異に適用する。選択には、許容型選択を用い、決められた数世代と最終世代において制約違反チェック付き選択を行う。ペナルティ係数はつねに決められた一定値を用いる。したがって、この解法では制約違反チェック付き選択の行われる世代を除き、制約違反個体でも淘汰されない。

一方、制約違反チェック付き選択を毎世代してしまうと、子個体の生成時には制約違反を許しても選択時に淘汰されてしまうため、許容の効果がない。このため、これを局所利己的遺伝子非許容型 GA (Locally Selfish-gene Non-Tolerant Genetic Algorithms, LSNT-GA) と呼んで LST-GA と区別する。

3.5.3 局所利己的遺伝子許容動的制御 GA (LSTDC-GA)

LST-GA と同様に局所利己的 NI 法を交叉とブロック型突然変異に適用し、選択には、許容型選択を用い、決められた数世代と最終世代において制約違反チェック付き選択を行う。一方、LST-GA では一定であった

ペナルティ係数や突然変異率などの GA パラメータを LSTDC-GA では動的に制御する。この LSTDC-GA のフローチャートを図 3 に示す。

次章では、上記の 3 つの GA の効果を評価する実験を行う。

4. 評価実験と結果

本実験では、AMD Athlon 64 X2 3800+2 GHz プロセッサ (シングルコアで起動) と 1 GB 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。

本論文では実験のために、配送先の 20% つまり 20 都市が時間枠制約を持つ 100 都市 TSP を次のように 2 タイプ用意した。

(タイプ 1) 緩い時間枠制約 TSP: 時間枠制約を持つ 20 都市すべてが 4 時間の時間枠制約を持つ TSP。

(タイプ 2) 厳しい時間枠制約 TSP: 時間枠制約を持つ 20 都市のうち、10 都市が 4 時間の時間枠制約、他の 10 都市が 1 時間の時間枠制約を持つ TSP。

4.1 実験 1 (GA-GA, LSNT-GA vs LST-GA)

局所利己的遺伝子と許容型選択の有効性を検証するために、GA-GA (大域利他的遺伝子型 GA) と LSNT-GA (局所利己的遺伝子非許容型 GA)、LST-GA (局所利己的遺伝子許容型 GA) を用いて実験を行った。

GA-GA は制約充足解だけを探索する。LSNT-GA は子個体の生成時には制約違反を許すが制約違反チェック付き選択つまり制約違反のチェックと修正を毎世代行う。LST-GA は 25 世代間隔で制約違反チェック付き選択を行う。

実験には、それぞれランダムに 50 ケース生成した 100 都市の緩い時間枠制約 TSP と 100 都市の厳しい時間枠制約 TSP を用いた。

各手法を用いて、これら 100 ケースの時間枠制約付き TSP を乱数の種を変化させて 1 ケースに対しそれぞれ 100 回解き、最適性と応答性を調べた。

GA のパラメータは予備実験の結果に基づいて決定した。計算時間を一定 (実験では 1 秒) に設定して性能比較を行った。集団数は 50, 100, 200 で試行し、最も高い性能を示した集団数 100 を採用した。交叉率と突然変異率は 5 ~ 40% の間で試行しすべての手法で平均的に高い性能を示した 20% を交叉率と突然変異率ともに本実験に採用した。予備実験ではどの手法にお

表 1 各解法の応答性

Table 1 Response of each method.

#	GAタイプ	緩い時間枠 制約TSP	厳しい時間枠 制約TSP
1	GA-GA	0.74 sec (Ave) 1.14 sec (Max)	0.96 sec (Ave) <u>5.14 sec (Max)</u>
2	LSNT-GA	0.78 sec (Ave) 1.16 sec (Max)	0.62 sec (Ave) 3.42 sec (Max)
3	LST-GA	0.50 sec (Ave) 0.60 sec (Max)	0.52 sec (Ave) <u>1.28 sec (Max)</u>

表 2 各解法の誤差

Table 2 Error rate of each method.

#	GAタイプ	緩い時間枠 制約TSP	厳しい時間枠 制約TSP
1	GA-GA	6.5%(Ave) 40.0%(Max)	<u>21.1% (Ave)</u> 57.2% (Max)
2	LSNT-GA	3.2%(Ave) 21.0%(Max)	8.1%(Ave) 24.5%(Max)
3	LST-GA	2.5%(Ave) 5.1%(Max)	<u>6.5%(Ave)</u> 16.5%(Max)

いても 40 世代付近でほぼ収束したため本実験の世代数は 50 世代とした。

4.2 実験 1 の結果

LST-GA の計算時間は、緩い時間枠制約 TSP で平均 0.5 秒と最も小さかった。表 1 に緩い時間枠制約 TSP と厳しい時間枠制約 TSP のときの各解法の応答性を計算時間の平均値と最大値で示した。また、表 2 に各解法の最適性を誤差の平均値と最大値で示した。誤差は、実験で用いた世代数より十分に長い世代数で 100 回実行した中での最良解との差として計算した。

実験 1 の結果によると、LST-GA が表 1 の応答性でも表 2 の解の最適性でも最も優れていた。緩い時間枠制約 TSP では差が少ないが、厳しい時間枠制約 TSP では LST-GA と GA-GA とで表 1 のように最大応答時間で 4 倍 (1.28 秒と 5.14 秒)、表 2 のように誤差の平均値で 3 倍以上 (6.5%と 21.1%) の差が出た。

4.3 実験 2 (LST-GA vs LSTDC-GA)

動的制御の有効性を検証するために、LST-GA (局所利己的遺伝子許容型 GA) と以下に示す 3 タイプの LSTDC-GA (局所利己的遺伝子許容動的制御 GA) の比較を行った。すべての手法で、制約違反チェック付き選択は 50 世代間隔で行った。

LSTDC-GA タイプ 1: 25 世代周期で突然変異率

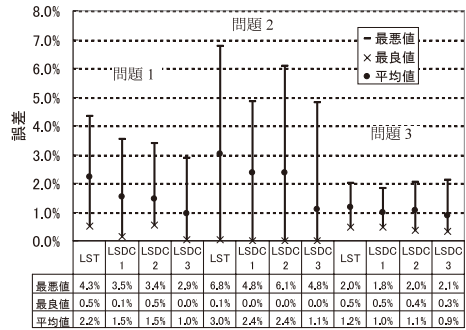


図 4 緩い時間枠制約 TSP での最適性

Fig. 4 Optimality at Non-Severe Time Constraints TSP.

とペナルティ係数を変化させる LSTDC-GA。ペナルティ係数を低い値から指数関数的に上げていき、25 世代周期で低い値に戻ると同時に突然変異率をその世代だけ上昇させる。

LSTDC-GA タイプ 2: 50 世代周期で突然変異率とペナルティ係数を変化させる LSTDC-GA。ペナルティ係数を低い値から指数関数的に上げていき、50 世代周期で低い値に戻ると同時に突然変異率をその世代だけ上昇させる。

LSTDC-GA タイプ 3: 新入り個体数が 5 以下になった世代でペナルティ係数を低下させると同時に突然変異率を上昇させる LSTDC-GA。

問題は、より困難な問題に対応するため、実験 1 で用いた問題の中から特に難しかった問題を参考に新たに緩い時間枠制約 TSP を 3 問 (問題 1~3) と厳しい時間枠制約 TSP を 3 問 (問題 4~6) 作成し、各問を 1 秒で打ち切るとともに、乱数の種を変化させて 100 回解き、その最適性を調べた。

GA のパラメータとしては、LST-GA は、交叉率と突然変異率をそれぞれ 20%、ペナルティ係数を 50 に設定した。LSTDC-GA は交叉率を 20%、突然変異率を 10~40%、ペナルティ係数を 5~50 で動的に変動させた。集団数は両手法ともに 100 とした。

4.4 実験 2 の結果

緩い時間枠制約 TSP での最適性を図 4 に、厳しい時間枠制約 TSP での最適性を図 5 に示す。表中では LST-GA を LST, LSTDC-GA タイプ 1 を LSDC1, LSTDC-GA タイプ 2 を LSDC2, LSTDC-GA タイプ 3 を LSDC3 と表記している。最適性は誤差の平均値、最良値、最大値により示した。誤差は、世代数を 1,000 として 100 回実行した中での最良解との差として計算した。

実験 2 の結果である図 4, 図 5 を見ると、LSTDC-GA の方が LST-GA より平均的に最適性が高かった。

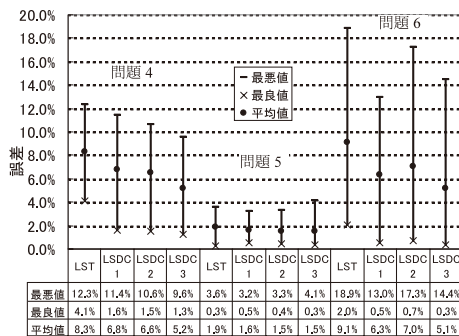


図 5 厳しい時間制約 TSP での最適性

Fig. 5 Optimality at Severe Time Constraints TSP.

図 4 の緩い時間制約 TSP での結果を見ると、問題 3 を除き最良解に差がなくても平均解と最悪解は 1~2%改善された。

図 5 の厳しい時間制約 TSP での結果を見ると、問題 5 ではほとんど差が見られないが、他の問題ではすべての項目が約 2~5%以上改善された。

緩い時間制約 TSP, 厳しい時間制約 TSP の両方で全般的に最も良い性能を示したのは、新入り個体数 5 以下で変化する LSTDC-GA タイプ 3 であった。

5. 評価

5.1 実験結果に基づく評価

5.1.1 局所利己的な遺伝子タイプ GA の効果

4.2 節の表 1 の結果が示すとおり、GA-GA は求解に時間がかかり、目的とする対話応答性を実現できなかった。これは、GA-GA はノードの挿入ごとに距離の増分以外にすべてのノードの時間制約をチェックするため 1 世代あたり都市数の 2 乗オーダーの高い計算コストを要するためであると考えられる。

また、解の最適性に関して表 2 の示す GA-GA の実験結果では、緩い時間制約 TSP の場合、誤差はそれほど高くないが、厳しい時間制約 TSP (時間の間隔が狭い) の場合、最大誤差が 20%以上になった。これは、多数の 1 時間の時間制約をすべて満たすのは非常に難しいため、集団の中の多くの個体が死滅したためと考えられる。実験では、5 世代目に集団内にはそのサイズの 5%程度の個体しか存在しなかった。このような理由から GA-GA は局所解収束の回避に必要な集団内の個体の多様性を確保できなかったため要求する最適性を達成することができなかったと考えられる。

5.1.2 許容の効果

4.2 節の表 1 の結果が示すとおり、LSNT-GA の応答性は LST-GA より 2 倍程度悪かった。

表 2 を見ると、両者の解法は緩い時間制約 TSP の場合、ほぼ同程度の最適性であった。しかし厳しい時間制約 TSP の場合の結果を見ると、LST-GA の最大誤差は 20%未満だが、LSNT-GA の最大誤差は 20%を超えていた。そこで厳しい時間制約 TSP の場合の LSNT-GA の集団内の個体数の変化を調べたところ、初期世代あるいはその後の交叉または突然変異で生成された個体の大部分が制約違反を修正できず死滅しており、10 世代目でも集団内に指定サイズ (実験では 100) の 10%以下の個体しか存在しなかった。つまり、LSNT-GA では厳しい時間制約 TSP を解く場合、集団が小さくなり、集団内の多様性が維持できず局所解収束に陥ってしまう可能性が高いため、最大誤差が大きくなってしまふものと考えられる。

これに対し、表 2 の LST-GA の実験結果を見ると、実験 1 で実験された 3 つの手法の中で最適性が最も高く、最大誤差は 10%未満であった。これは、制約を満たすことができなかった個体でも世代を越えて生き残らせ、局所利己的な NI 法を用いた交叉や突然変異により制約違反の修正機会を個体に与えることにより、集団内の多様性が維持され効率的な探索が行えたためと考えられる。

5.1.3 動的制御の効果

実験 2 の結果である図 4, 図 5 を見ると、緩い時間制約 TSP では大きな差は見られなかったが、厳しい時間制約 TSP では動的制御を用いた方が圧倒的に良くなっている例があった。LST-GA に比べて LSTDC-GA の方が最良値、最悪値、平均値すべてで約 2~5%以上性能が改善された。

これは、制約の厳しい問題ほど局所最適解に陥る可能性が高いためと考えられる。LSTDC-GA では、動的制御を用いることにより、下記の場合など特に、集団内の多様性が維持され局所最適解へ陥る可能性が減少したため最悪解の値が改善されたものと考えられる。

LSTDC-GA では、周期的に変化させるよりも、新入り個体数をトリガとして変化させた方が良い結果が得られた。これは、周期的に変化させる場合は、その最適周期は問題依存のため探索の効率化の効果が低いと考えられる。一方、新入り個体数をトリガとして変化させる方式では、新入り個体数が減少するときは解が収束してきた場合が多いため、適切なタイミングで集団内の多様性を回復する処理が行われたと考えられる。

進化の過程での新入り個体の数を観測したところ、LST-GA では新しい個体の生成ができない世代が多く見られ、進化が足踏み状態になっていると考えられ

る．一方 LSTDC-GA では後半の世代においても、新しい個体が集団に加わる様子が見られた．こうして、集団の多様性が維持され解が少しずつ改善されていると考えられる．

また、一般に制約が厳しい場合、制約を満たす十分な数の新しい子個体の生成が困難なばかりか、制約違反の個体を淘汰すると親個体の数も減り集団が小さくなる．本手法では制約違反により淘汰を行う世代では、その結果、定められた個体数より集団が小さくなくても制約を満たす個体が定められた個体数になるまで子個体の生成を行うなどの処理をしない．これを行うと 1 世代の計算に時間を要し、対話応答時間内に十分な世代の計算が行えず、また、これにより得られた個体は似たような個体である可能性が高いため、結果的に性能が悪くなる．しかし、集団が小さくなりすぎても探索の効率が悪くなることが予想される．そのため適度な集団の大きさを保つように動的制御をさらに工夫することは今後の課題の 1 つである．

5.2 他の解法との比較

現在までに、時間枠制約の存在する TSP を解決するための他のいくつかの方法が提案されている^{12),19)}．IP とヒューリスティックス¹⁹⁾ を組み合わせた解法は 50 都市の TSP において、3%未満の誤差という高い最適性を示している．しかしながら、この解法はこのような規模の TSP でも 30 秒も要するので、対話応答性を必要とするケースには適切ではない．

理論上、SA¹⁷⁾ は局所解に陥るリスクを減少させることによって最適な解を捜し出すことができるといわれている．しかし、通常、SA では、最適に近い解を得るには計算時間がかかる．

SA は実験では 15 都市の TSP を解くのに 400 ミリ秒かかり、100 都市の TSP を解くのに 40 秒かった．また、Hooker らによる Tabu Search⁴⁾ においても、実際に最適解を得るのに長い計算時間を要する．したがって、これらの方法も、本論文が対象とする用途に必要な対話応答性を満たさない．

TSP を解くために多くの種類の GA が提案されている^{2),6),11),13),15)}．Nguyen らの研究⁵⁾ では 1,000 ~ 10,000 都市以上の規模の TSP を対象としており、規模は非常に大きく、その解法の価値は高い．一方、本論文の解法は実用上必要な時間枠制約と規模を持つ TSP を対話時間内に解くうえで重要性が高いと考える．

また、制約問題を扱った GA も様々なものが提案されているが¹⁶⁾、これらは一般的な制約問題に対して最適化を行うものであり、時間枠制約付き TSP を効率的に解くことはできない．

時間枠制約のある配送先への巡回ルートを求める組合せ最適化問題としては、Vehicle Routing Problem with Time Windows (VRPTW)⁹⁾ が知られている．VRPTW とは、制約条件として各車両の容量制約と各顧客への時間枠制約の下で、複数の車両を用いて、すべての顧客をちょうど 1 回ずつ訪問するような経路集合の中で、コストが最小のものを求める問題である．

VRPTW の解法^{1),12)} としては様々な手法が提案されている．VRPTW と本論文で対象とする荷物や書簡の配送問題との間には、以下のような相違がある．

VRPTW では各車両の荷物の容量制約も扱うが、本論文が対象とする問題では、1 つの荷物が小さいため多数の配送先を巡回しても容量制約に違反するようなことはなく、総距離の最適化がコストに最も影響するような環境を想定している．そのため容量制約を扱う必要はない．

また、VRPTW では複数の車両で巡回し、車両数と各車両が担当する配送先の最適化も行う．しかし、本論文で対象とする荷物や書簡の配送問題では、車両数や各車両の担当地域は前もって決められている．というも実際の現場では、そのときの配送先によって毎回、担当地域や車両数を変更するのは労務上の問題も絡み効率的でない場合が多い．これらは事前にその配送元が担当する区域の過去の配送実績から経験的、あるいは GA 以外のより単純な最適化手法で決定される．

また、実際の現場では配送ルートでさえ、システムにより計算された結果をそのまま用いることは少なく、システムが計算した結果に対し社会的事情などを考慮した修正が人間の手により行われることが多い．担当している地域が事前に決められている場合は、このようなルートの修正はその地域の担当者により簡単に行えるが、VRPTW のように各車両や配送者の担当地域まで機械的に最適化されてしまうと他の地域まで絡むので難しい．

次に既存の VRPTW に対する解法と本論文で提案した解法との比較を試みる．VRPTW における一般的なベンチマーク問題として、Solomon が作成した問題¹⁰⁾ がある．この問題はすべての巡回箇所には厳しい時間枠制約がある．このように厳しい制約のある問題は、制約により解探索範囲が大幅に絞れるので、いわゆる制約指向アプローチでの解法が適している．一方、本論文の対象としているような実用的な時間枠制約付き TSP では、制約が厳しいといっても制約指向で探索範囲が絞りきれないほどではないのでかえって難しく、本提案手法が必要になる．以上の理由で上記 Solomon

のベンチマーク問題は性能比較に多用されるが、本提案手法で解くには適していないので、この問題による実験比較はあまり意味がないと考える。

この点、Solomon の問題の中でも最も制約が緩やかで、本論文で扱う問題に比較的近い実用的なものとしては 100 都市 VRPTW である C1 問題がある。村田らの研究¹⁸⁾では、この C1 問題において、解が収束するまでにおよそ 1,000 世代、計算時間に 120 秒かかっている。また、実験環境は Pentium IV 3.2 GHz, 512 M RAM の計算機を使用しており本論文で用いた計算機とほぼ同程度の計算能力である。

本論文では 100 都市の時間枠制約付き TSP をおおよそ 100 世代ほどで収束し、1 秒程度で解くことができた。複数台による最適化では巡回する配送先の数が同じならより簡単に時間枠制約を満たすことができるため、車両の容量制約の最適化にかかる計算コストを考慮しても問題の難しさにそれほどの差はないものと考えられる。トラック数を決定するために数回程度トラック数を変えて実行しても、村田らの上記手法と比較して数十分の一程度の時間しかかからない。解いている問題が異なるため、単純な比較はできないが、十分な計算時間がとれる場合には先行研究の方が良い解を導出できるかもしれない。しかし、本論文が目的としているような対話応答性が求められるような場合には、以上のように本提案手法の方が適していると考えられる。

6. おわりに

本論文では、数十から百都市前後の規模の時間枠制約付き TSP の解法として、局所利己的遺伝子許容型 GA (LST-GA) とその改良手法である局所利己的遺伝子許容動的制御 GA (LSTDC-GA) を提案した。これらの手法はすでに示した「局所利己的遺伝子」、「許容」、「動的制御」の 3 つの概念に基づくものであった。

これらの手法を用いて、100 都市の時間枠制約付き TSP において最大誤差が 20% 以下の専門家レベル以上の解が、実験では 1 秒以下の対話応答時間内で求まることを確認した。

また、局所的に制約を満たそうとする「局所利己的遺伝子」の概念を導入した局所利己的 NI 法を用いることにより、時間枠制約付き TSP を効率的に解き、短い計算時間でも高い精度の解が得られることを示した。

そして、「許容」の概念を導入した選択法を用いて、制約違反個体の生存を許すことにより、局所最適解に陥るのを回避し、制約違反を起こした個体でも世代を越えた反省か修正の機会を数世代にわたって与えるこ

とにより、最終的に制約を満たし、かつ比較的最適性の高い解が得られることを示した。

また、「動的制御」の概念を用いて、ペナルティ係数と突然変異率を動的に制御することにより、局所最適解に陥りやすい制約の厳しい問題に対して解の最適性を約 2~5% 以上改善することができることを示した。

今後の課題としては以下の項目が考えられる。

- (1) 動的制御を利用した適切な集団サイズの制御。
- (2) 重み付きマップと呼ばれる各道路の走行速度情報を用いた、よりきめ細かい最適化：渋滞などの影響に関しては曜日や時間帯ごとに重み付マップを用意することで対応。
- (3) リアルタイムでの渋滞への対応：渋滞情報をリアルタイムに反映して重み付マップの変更を行ってルートを再計算することで対応。

謝辞 有用なコメントをくださった査読者の方に、謹んで感謝の意を表す。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

- 1) Bent, R. and Hentenryck, P.V.: A Two-Stage Hybrid Local Search for the Vehicle Routing Problem with Time Windows, Technical Report, CS-01-06, Brown University (2001).
- 2) Davis, L.: *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, NY (1991).
- 3) Dawkins, R.: *The Selfish Gene*, Oxford Univ. press (1989).
- 4) Hooker, J.H. and Natraj, N.R.: Solving a General Routing and Scheduling Problem by Chain Decomposition and Tabu Search, *Transportation Science*, Vol.29, No.1, pp.30-44 (1995).
- 5) Nguyen, H.D., Yoshihara, I., Yamamori, K. and Yasunaga, M.: Implementation of an Effective Hybrid GA for Large-Scale Traveling Salesman Problems, *IEEE Trans. Systems, Man and Cybernetics*, Part B, Vol.37, Issue1, pp.92-99 (2007).
- 6) Reeves, C.R.: Genetic algorithms, *Modern Heuristic Techniques for Combinatorial Problems*, Reeves, C.R. (Ed.), McGraw-Hill Book Co., Europe (1995).
- 7) Sakurai, Y., Onoyama, T., Kubota, S., Nakamura, Y. and Tsuruta, S.: Selfish-gene Tolerant Generic Algorithms to solve large-scale constraint TSPs, *Proc. 2006 IEEE International Conference on Information Reuse and Integration (IEEE IRI2006)*, pp.154-159 (2006).

- 8) Sakurai, Y., Onoyama, T., Kubota, S., Nakamura, Y. and Tsuruta, S.: A Multi-world Intelligent Genetic Algorithm to Interactively Optimize Large-scale TSP, *Proc. 2006 IEEE International Conference on Information Reuse and Integration (IEEE IRI2006)*, pp.248–255 (2006).
- 9) Solomon, M.M.: Algorithms for Vehicle Routing and Scheduling Problem with Time Window Constraints, *Operations Research*, 35, pp.254–265 (1987).
- 10) Solomon, M.M.: VRPTW BENCHMARK PROBLEMS. <http://web.cba.neu.edu/~msolomon/problems.htm>
- 11) Tamaki, H., et al.: A comparison study of genetic codings for the traveling salesman problem, *Proc. 1st IEEE Conference on Evolutionary Computation*, pp.1–6 (1994).
- 12) Tan, K.C., Lee, T.H., Ou, K. and Lee, L.H.: A Messy Genetic Algorithm for the Vehicle Routing Problem with Time Window Constraints, *Proc. CEC2001*, pp.679–686 (2001).
- 13) Whitley, D., Starkweather, T. and Fuquay, D.: Scheduling problems and traveling salesmen: The genetic edge recombination operator, *Proc. 3rd International Conference on Genetic Algorithms and their Applications*, pp.133–140 (1989).
- 14) 磯部俊吉, 佐野設夫, 佐藤政則: デリバリー・プランニング・システム, 郵政研究所月報, No.107, pp.79–90 (1997).
- 15) 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).
- 16) 高濱徹行, 阪井節子: ϵ 制約遺伝的アルゴリズムによる制約付き最適化, 情報処理学会論文誌, Vol.47, No.6, pp.1861–1871 (2006).
- 17) 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).
- 18) 村田裕章, 越野 亮, 白山政敏, 林 貴宏, 木村春彦: 改善集団と改悪集団による進化的停滞を回避する遺伝的アルゴリズムと時間枠つき配送計画問題への適用, 情報処理学会論文誌：数理モデル化と応用, Vol.47, No.SIG14, pp.1–8 (2006).
- 19) 池上敦子, 丹羽 明: 時間指定のあるピークルーティング問題, 日本オペレーションズ・リサーチ学会論文誌, Vol.38, No.1, pp.107–123 (1995).

(平成 19 年 4 月 23 日受付)

(平成 19 年 7 月 2 日再受付)

(平成 19 年 7 月 29 日採録)



櫻井 義尚 (正会員)

1977 年生. 2000 年電気通信大学電気通信学部電子情報学科卒業. 2002 年電気通信大学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 年電気通信大学大学院電気通信学研究科電子情報学専攻博士後期課程単位取得済み退学. 同年工学博士. 現在, 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および応用に関する研究に従事. 日本知能情報ファジィ学会, IEEE 各正会員.



小野山 隆 (正会員)

1959 年生. 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. IEEE 会員. 博士(情報科学).



久保田 仙 (正会員)

1975 年生. 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.



中村 嘉宏

1984 年生. 現在東京電機大学大学院情報環境学研究科修士課程在学中. 遺伝的アルゴリズム, 分散遺伝的アルゴリズムを用いた巡回セールスマン問題の解法に関する研究に従事.



鶴田 節夫 (正会員)

1947 年生. 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. 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.