

## 制約付き TSP のための局所利己的遺伝子動的制御 GA の提案

櫻井 義尚<sup>†</sup> 小野山 隆<sup>††</sup> 久保田 仙<sup>††</sup> 中村 嘉宏<sup>†</sup> 鶴田 節夫<sup>†</sup>

サプライチェーンマネジメントに適用可能な広域物流網シミュレータなどの実現には時間などの制約がある数十から数百都市の大規模巡回セールスマン問題 (TSP) を対話的応答時間内に専門家レベルの最適で解くことを要求される。この要求を満たすために、局所利己的遺伝子動的制御 GA (Locally Selfish-gene Dynamic Control 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を強制的な修正率および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する。本解法の適用で、時間制約が存在する大規模 TSP において最大誤差が 1 割前後の解が数秒以内に求まることを実験により確認した。

### Locally Selfish-gene Dynamic Control GA to Solve Large-scale Constraint TSPs

Yoshitaka Sakurai<sup>†</sup> Takashi Onoyama<sup>††</sup> Sen Kubota<sup>††</sup>  
Yoshihiro Nakamura<sup>†</sup> Setsuo Tsuruta<sup>†</sup>

Large-scale distribution network simulation applicable to supply-chain management requires to solve tens of time-constraint large-scale (max 100 cities) Traveling Salesman Problems (TSP) within interactive response time, with practicable optimality. To meet this requirement, a Locally Selfish-gene 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 generations and are given the chance of improvement. Moreover, evolution is promoted by dynamically changing the degree of the tolerance. Our experiment proves that this method provides expert-level solutions for time constraint large-scale TSPs within a few seconds.

## 1. はじめに

物流の効率化は重要な問題であり、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サプライチェーンマネジメントの観点からの生産効率向上には、物流の効率化だけでなく、「ジャストインタイム」などの厳しい時間制約を満たすことによる生産フロー全体の効率化が求められる。一方で、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も高まっている。これらの要求を満たすためには配送時間の制約を守った上で、コスト面でも最適な大規模なトラックの配送ネットワークを作成する必要がある。

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。しかし、厳しい時間制約の状況下では、専門家ですら最適なルートを見つけるのは非常に困難な作業であり、専門家が出した答えと最適解とを比べると、場合によっては誤差が 50% ぐらいになることもあり、問題になっていた。そこで、コンピュータにより対話的な応答時間内に最適なルートを作成し、人間の専門家が社会面からそのルートの評価することにより大規模な配送ネットワークの構築を試みる。

対話的な応答時間を例えば 10 分未満とすると、大規模な配送ネットワークには何百もの配送ルートがあるので、1 つのルートが数秒以内に作成されなければならない。また、それぞれのルートの生成は数十都市の TSP (Traveling Salesman Problem) を解くのと等価である。ゆえに TSP に対して対話的な応答性と高い最適性の両方を可能とする近似解法が必要である。

そこで、我々は厳しい時間制約の状況下における TSP を 20% 未満の誤差 (最適解との差) と対話的な応答性を保証して解決することを目的として GA によ

<sup>†</sup> 東京電機大学

Tokyo Denki University

<sup>††</sup> (株)日立ソフトウェアエンジニアリング

Hitachi Software Engineering Co., Ltd.

る近似解法を提案した[1]。この手法は特に厳しい制約問題において、時折最適度の低い解を出してしまうことがあり、より安定的に解を求める必要があった。本論文では、これに動的制御(Dynamic Control)の概念を取り入れることにより、より制約が厳しい問題に対して最適度を改善させる局所利己的遺伝子動的 GA (Locally Selfish-gene Dynamic Control Genetic Algorithms, LSDC-GA)を提案する。

## 2. 局所利己的遺伝子動的 GA (LSDC-GA)

### 2.1. LSDC-GA の概要

技術的課題、①専門家レベルの最適性、②対話的応答性を満たす時間枠制約付き TSP の解法として、我々は局所利己的遺伝子許容型 GA (Locally Selfish-gene Tolerant Genetic Algorithms, LST-GA)を提案した。

GA による最適化では各遺伝子に制約が有る場合、全ての制約を充足する個体のみを生成したり、初期集団から、あるいは毎世代毎に、制約を充足しない個体を淘汰したりすると、新しい固体の生成が困難になり、集団が小さくなる。すると、個体の多様性が失われて局所最適に陥り、有効な最適化を実現できない。つまり、制約を充足する個体だけで集団が構成されると、大きな改善は期待できず、逆に局所最適に陥るおそれ大きい。

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局所利己的 NI 法という手法を使った GA オペレータを用いる。そしてこのようなオペレータにより生まれた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事を許し(許容)、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が与えられる。そして、特定な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のを許容することで、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。

本論文では、これに「動的制御(Dynamic Control)」の概念を取り入れることにより、より安定的に最適解を求める局所利己的遺伝子動的 GA (Locally Selfish-gene Dynamic Control Genetic Algorithms, LSDC-GA)を提案する。

動的制御とは、「環境を動的に変えることにより、平衡状態にある進化を推し進める」というものである。これにより、局所最適解へ陥るのを防止し、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、探索の効率化が望める。

### 2.2. 許容による制約緩和

#### (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  $V=L+P*T$  となる。P はペナルティ定数であり、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。その世代の個体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。

#### (2) 許容型選択:

本手法では、GA 集団中の固体は時間枠制約を全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、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し、数世代毎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。

##### a) 適応度のみによる選択

時間枠制約を満たすかどうかはチェックせず、適応度のみから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が満たすことが出来なかった個体は淘汰される。

### 2.3. 動的制御による進化の促進

本手法では、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ま

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、選択における修正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取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が生まれやすくなり、局所最適解からの脱出効果が見込まれる。そこで許容度(ペナルティ値)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。

#### (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。ペナルティ値が低い状態から世代毎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。

#### (2) 新入り個体数をトリガーとした変化:

突然変異率と許容度を新入り個体数により上下させる。ペナルティ値が低い状態から世代毎に指数関数的にその値を増やしていく。そしてその間、各世代で集団に新しく入ってくる個体の数をカウントする。この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。

### 2.4. LSDC-GA の構成

#### (1) 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。ペナルティ値は常に決められた一定値を用いる。したがって、この解法は時間枠制約を違反する個体にも自由に繁殖することを一定期間許容する。

#### (2) 局所利己的遺伝子動的 GA (LSDC-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。LST-GA ではペナルティ値や突然変異率などの GA パラメータが一定であった。LSDC-GA では動的制御を用いることによりペナルティと突然変異率を動的に制御する。

### 3. 評価実験と結果

本実験では、AMD Athlon 64 X2 3800+ 2GHz プロセッサ(シングルコアで起動)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。問題の配送ネットワークの規模を考慮し、100都市の TSPs(各々異なったタイプの時間枠制約を持つ20都市を含む)を用意した。2つのタイプの時間枠制約があり、ひとつはそれほど厳しくない半日単位の期間タイプ(弱い時間枠制約)、そしてもうひとつは厳しい1時間単位の期間タイプ(強い時間枠制約)である。

#### 3.1. 実験 (LST-GA vs LSDC-GA)

動的制御の有効性を検証するために、LST-GA と LSDC-GA の比較を行った。LST-GA は1タイプ、LSDC-GA は、25世代周期変化のタイプ1、50世代周期変化のタイプ2、新入り個体数5以下で変化するタイプ3の3タイプを用いた。

問題は、弱い時間枠制約問題と強い時間枠制約問題をそれぞれ3問作成し、それぞれ1秒以内で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。LSDC-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。

#### 3.2. 実験2の結果

弱い時間枠制約問題での最適度を図1に、強い時間枠制約問題での最適度を図2に示す。

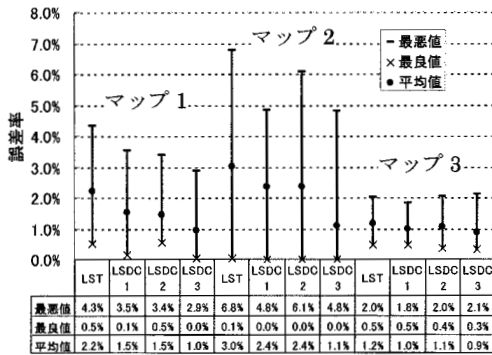


図 1. 弱い時間枠制約問題での最適値

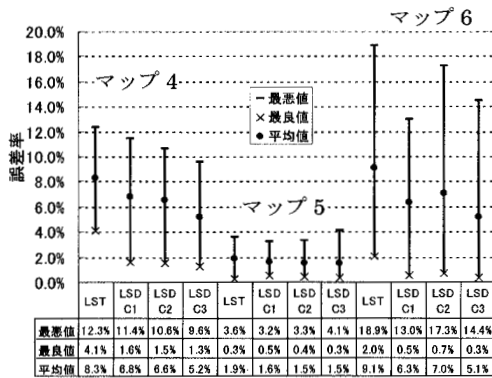


図 2. 強い時間枠制約問題での最適値

実験 2 の結果を見ると、LSDC-GA の方が LST-GA より平均的に優れているのが見て取れる。弱い時間枠制約問題での結果を見ると、マップ 3 などではほとんど差が無いが、他のマップでは最良解には差が無いが平均解と最悪解はおおよそ 1% 改善している。強い時間枠制約問題での結果を見ると、マップ 5 ではほとんど差が見られないが、他のマップではすべての項目がおおよそ 2% 程度改善されている。

弱い時間枠制約問題、強い時間枠制約問題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 LSDC-GA タイプ 3 であった。

### 3.3. 動的制御の効果

実験結果を見ると、弱い時間枠制約の問題では大きな差は見られないが、強い時間枠制約の問題では動的制御を用いた方が圧倒的に良くなってい

る例がある。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。LSDC-GA では、動的制御を用いることにより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。

LSDC-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ー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、その最適な周期は問題依存などによりそれぞれ違うために探索の効率化の効果が低かった可能性がある。一方、新入り個体数をトリガーとして変化させる場合は、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処理が行われた可能性が高い。

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ずに進化が足踏み状態になっている世代が多く見られた。一方 LSDC-GA では後半の世代においても、新しい個体の生成が行われており、これにより解が少しずつ改善されるようが見取れた。

## 4. おわりに

本論文では、時間枠制約がある大規模な TSP の解法として、局所利己的遺伝子許容型 GA (LST-GA) について説明し、その改良手法である局所利己的遺伝子動的 GA (LSDC-GA) を提案し、実験によりその有効性を検証した。実験結果から、「動的制御」の概念を用いて、ペナルティ値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改善できることを示した。

また、提案された方法で強い時間枠制約を持つ大規模(例えば 100 都市)TSP において対して対話応答時間内(数秒)で専門家レベルの解を得ることが可能なことが実験結果により示された。これにより、この手法は大規模な最適配送ネットワークの構築を目的とする対話的シミュレーションに適用可能なことが示された。

<参考文献>

[1] Sakurai, Y., Onoyama, T., Kubota, S., Nakamura, Y., Tsuruta, S.,: "Selfish-gene Tolerant Generic Algorithms to solve large-scale constraint TSPs", Proc. of The 2006 IEEE International Conference on Information Reuse and Integration, pp154-159, (2006)