

先見積もり分割処理を用いた並列型免疫的共進化アルゴリズムの提案

A-017 A Proposal of A Parallelized Immune Co-evolutionary Algorithm Using Pre-estimated Division Processing

當間 愛晃[†]
Naruaki Toma

遠藤 聡志[‡]
Satoshi Endo

山田 孝治[‡]
Koji Yamada

1. はじめに

近年、PC クラスタに代表されるように、大規模計算のための計算環境を安価に整えられるようになってきた。これに伴い、遺伝的アルゴリズム（以下、GA）も母集団の並列化による高速化・探索効率向上を実現した手法（並列分散 GA）[2] や、パラメータ設計に要する労力を軽減する分散環境型並列分散 GA[1] 等が提案されている。これまでに、我々は分業問題の典型例である分業巡回セールスマン問題（以下、 n -TSP）[5] を対象とした探索手法として、共進化型免疫 IA を提案してきた[4]。本稿では、共進化型免疫 IA を並列化することによる探索速度の向上および、今回提案する先見積もり分割処理による探索性能の向上のための一手法を提案する。

2. 分業巡回セールスマン問題 (n -TSP)

n -TSP とは、巡回セールスマン問題 TSP におけるエージェント数を n 人に拡張した問題 [5] であり (1) 各エージェントの担当する作業領域を平等に分割し (2) 担当領域における作業コストを最小化しつつ (3) 全作業コストを最小化するような解（分業解）を求めることを目標とする（図 1）。

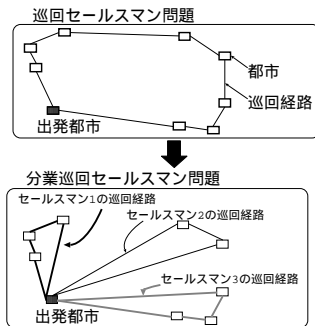


図 1: n -TSP の概念図

3. 提案手法

本原稿では、[4] に並列化による探索速度の向上・先見積もり分割処理による探索性能の向上という 2 点の拡張を施したアルゴリズムを提案する。

3.1 アルゴリズム全体

本提案アルゴリズムにおける並列化は (1) 提示された n -TSP を都市空間上でサブ問題へと切り分け (2) 各サブ問題を独立した n -TSP とみなして共進化型 IA で探索し (3) 得られた解集団に対し逃避処理で全体のコスト調整を図るものとして実現する。

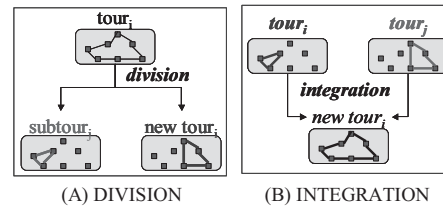


図 2: 分割・統合処理の例

3.2 サブ問題化

問題空間を複数のサブ問題へと切り分ける方法は、クラスタリングなど様々な方法が考えられるが、本稿ではシンプルで以下の手続きを用いた。提示された都市空間を 2×2 の 4 格子からなる二次元格子状に均一に切り分ける。このとき、格子内に予め設定した上限都市数 $MaxCity$ 以上の都市が配置されているならば、それが $MaxCity$ 未満となるまでその格子に対し同一の処理を施す。

3.3 共進化型 IA[4]

共進化型 IA は (1) 免疫エージェント間の相互作用を基本動作とする分割・統合処理と (2) 抗原エージェントが免疫エージェントと相互作用を行う逃避処理、という二種類のエージェントによる共進化現象に基づく探索により分業問題を解く。

[Step1. 抗原の認識] 問題（都市数，都市配置），およびセールスマン（数，MHC）を定義する。また、初期セールスマン数や都市の MHC 情報はランダムに初期値を決定する。MHC とは各セールスマンの ID であり、各都市がその都市を巡回するセールスマン ID を情報として有しており、他のセールスマンを認識し、分割・統合する際に参照するために使用される。なお、セールスマンの巡回経路はパス表現でコード化し、初期巡回経路は重複都市、未訪問都市が存在しないようにランダムに生成される。

[Step2. 目的関数計算] 各セールスマンの巡回経路の評価値として巡回経路長を計算する。

$$Cost(S_i) = \sum distance(tour_{S_i}), \quad (1)$$

(S_i : セールスマン i , $tour_{S_i}$: S_i の巡回経路)

[Step3. 分割処理] 各セールスマンは、コスト最適化のために自順路を二つのサブツアーへと分割する。図 2-(A) はイメージ図である（詳細は 3.4 節）。

[Step4. 統合処理] コスト最適化のため、任意の 2 セールスマンの順路を統合する（図 2-(B)）。

1. 順路 $tour_i, tour_j$ の結合点を任意に決定。

[†]財団法人沖縄県産業振興公社
[‡]琉球大学工学部情報工学科

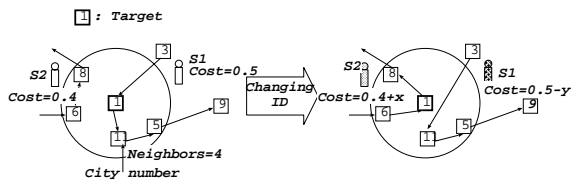


図 3: 逃避処理

2. 決定した結合ポイントに, 一方の巡回経路を結合し, 新たな順路 $new\ tour_i$ を作成.
3. 新たに生成された順路のコストを計算する. もし, 結合後のコストが減少しているならばその結合処理を実行する. 逆に, コストが増加しているならば元の順路へと戻す.

[Step5. 突然変異] 各セールスマンは突然変異 (任意の二都市の巡回順序をスワップ) により新たな順路を生成し, その後のコストが減少しているならばその順路を採用する.

[Step6. 目的関数計算] 抗原エージェントである都市の進化に必要な目的関数を計算する. ここでは, その都市を訪問しているセールスマンのコストをそのまま採用する.

[Step7. 逃避処理] 各都市が近傍都市を訪問するセールスマンのコストを比較し, コストの低いセールスマンの ID へ変更する (例図 3).

1. 予め設定された近傍都市数によって規定される近傍都市群に対し, それらの都市を訪問しているセールスマンのコストを確認する.
2. 自都市のセールスマン S_i のコストを下回るセールスマン S_j が存在するなら, 自都市の ID を S_j へと変更する.

以下, 終了条件 (処理ステップ数) に達するまで Step2~7 を繰り返す.

3.4 先見積もり分割処理

分割処理は, セールスマンの持つ順路を二つのサブ順路として分割することによりコスト最適化を図る. 以前の分割処理はサブツアーをランダムに決定する方法により探索を行っていたため, 一セールスマンの順路に含まれる都市数が 100 都市程度になると, 十分な探索が行われずにコスト最適化の側面が不足していた. 本稿では以下に示す先見積もり分割処理によりこの点を改善する. 先に見積っているコストは「現在の経路に含まれる部分経路 sub の巡回順番を変更した際に生じるコスト差」であり, これが改善される場合には分割を行うものとした (下記ステップ 4).

1. 任意の二点間により部分経路 sub を決定する. sub とその前後 1 都市ずつを含めた経路を $Before1$, $Before1$ から sub を取り除いた経路を $After1$ と呼ぶ.
2. sub の先頭都市に最近傍の都市 $nearest$ を求める.

3. $nearest$ の前後に sub を結合した場合の巡回コストを算出し, コストが低いほうを求める. このとき, 選ばれた都市 ($nearest$ + 次の都市 or 前の都市 + $nearest$) からなる経路を $Before2$, $Before2$ に sub を結合した経路を $After2$ と呼ぶ.

4. 上記 1~3 により求めた 4 経路のコスト比較により分割処理を実行するかどうかを決定する (下式).

$$If\ Cost(A) > Cost(B)\ then,\ True\ division$$

$$False\ no\ division,$$

$$where\ A = Before1 + Before2,$$

$$B = After1 + After2. \quad (2)$$

4. 基礎実験

先見積もり分割処理の導入による探索性能の改善度を調査するため, 論文 [4] でも取り扱った同一の二重同心円環型問題 (都市数 49, $R=0.5$, $r=0.384$, 出発都市数 1 (円の中心), 近傍都市数 3, 最小セールスマン数 1, 終了ステップ数 200) に適用した. 表 1 は各ノードで得られたコストを示しており, 旧分割処理と比較してコストの低い解を獲得していることが分かる.

表 1: コスト比較

ノード番号	旧分割処理	先見積もり分割処理
node_0	2.200276	2.053606
node_1	2.569807	2.379430
node_2	2.200277	2.148661
node_3	1.822837	1.822837

5. おわりに

本稿では, 大規模 n -TSP を解くことを目的とした, 問題空間のサブ問題化 (並列化) による探索速度の高速化および, 先見積もり分割処理の導入によるコスト最適化性能の改善を行った. 紙面の都合上省かざるを得なかったが, 並列化に伴う速度改善やコストの先見積もり分割処理の導入による優位性については発表時に示す予定である.

参考文献

- [1] Hintending, R., et. al.: "The Nature of Mutation in Genetic Algorithms, In H.-P. Schwefel and R. Manner, eds., Parellel Problem Solving from Nature", Volume 496 of Lecture Notes in Computer Science, pp.23-32, Springer (1995).
- [2] R. Tanese, : "Distributed Genetic Algorithm", Proc. 3rd International Conf. on Genetic Algorithm, pp.434-439 (1989).
- [3] 當間, 遠藤, 山田: "二種類の記憶機構を導入した適応的免疫アルゴリズムの提案と評価", 人工知能学会, Vol.15, No.6, 2000. pp.1097-1106 (2000).
- [4] 當間, 遠藤, 山田: "免疫細胞間協同作用および抗原との共進化現象を用いた免疫的分業問題最適化手法の提案と評価", 日本ファジイ学会誌, Vol.4, No.6, 2002. pp.616-629 (2002).
- [5] 中村 友洋, 他: "分業セールスマン問題のニューラルネットワークによる解法", 人工知能 94-2 (1994).