

## 粒子群最適化を用いた巡回セールスマン問題の解法

## An Algorithm for Traveling Salesman Problem using Particle Swarm Optimization

山田 悠希† Yuki Yamada  
 穴田 一† Hajime Anada

## 1. はじめに

工業や経済の問題の多くは、最も効率が良い組み合わせを求める組み合わせ最適化問題に帰着することができる。その中に、与えられた全ての都市を巡る最短経路を求める巡回セールスマン問題 (Traveling Salesman Problem, TSP) という問題がある。本庄らは、最適化問題に用いられるアルゴリズムの一つである粒子群最適化 (Particle Swarm Optimization, PSO) [1] を TSP 向けに改良した挿入操作 PSO 戦略 (Insertion-based PSO strategy, IPSO) [2] を提案した。IPSO は、解空間上に配置された各粒子がそれまでの最良解と、近傍の粒子の最良解の情報を基に解の更新を繰り返すことで解空間の探索を行うアルゴリズムである。しかし、この IPSO には探索が十分に行われないうちに、局所解に陥ってしまうという問題点がある。

そこで本研究では、既存手法で用いられた各粒子のそれまでの最良解と近傍の粒子の最良解の情報に加え、解空間上で最も遠い粒子の解の情報を現在の解に重ね合わせた解の集合を用いて、解の更新を行うアルゴリズムを構築した。これにより、既存手法よりも解の探索範囲が拡大することが期待される。

## 2. 既存手法

## 2.1 粒子群最適化

粒子群最適化 (Particle Swarm Optimization, PSO) とは、魚や鳥などに見られる群行動を探索手法に応用した、最適化手法の一つである。解空間上に位置と速度を持った複数の個体 (以下、粒子と表記) をランダムに配置する。各粒子の位置は問題の解を表現しており、適応度の高い粒子の情報を近傍の粒子と交換し、その情報を基に、より良い位置に近づくように速度と位置を更新する。この操作を繰り返すことで、解空間を探索する。 $t$  イテレーション目における粒子  $i$  の位置  $x_i(t)$  と速度  $v_i(t)$  の更新式は次式で定義される。

$$x_i(t) = x_i(t-1) + v_i(t-1) \quad (1)$$

$$v_i(t) = wv_i(t-1) + c_1r_1(pbest_i - x_i(t)) + c_2r_2(lbest_i - x_i(t)) \quad (2)$$

ここで、 $w$  はパラメータ、 $c_1$ 、 $c_2$  は  $[0,1]$  のパラメータ、 $r_1$  と  $r_2$  は  $[0,1]$  の一様乱数、 $pbest_i$  は粒子  $i$  のそれまでの最良解、 $lbest_i$  は粒子  $i$  の近傍の中のそれまでの最良解である。アルゴリズムの流れの詳細は以下の通りである。

## ①初期設定

全粒子の位置と速度をランダムに設定し、各粒子  $i$  のそれまでの最良解  $pbest_i$  を現在位置に設定する。次に、設定した近傍数  $k$  を元に、各粒子  $i$  と距離が近い  $k$  個の粒子を粒子  $i$  の近傍に設定する。そして、各粒子  $i$  の近傍の中で適応度が最も

† 東京都市大学, Tokyo City University

高い解を近傍内の最良解  $lbest_i$  と設定し、全粒子の中で適応度が最も高い解を  $gbest$  と設定する。

## ②位置の更新

(1)式に従い、各粒子の位置の更新を行う。

## ③適応度の評価

全粒子の適応度の評価を行う。適応度は問題に適した粒子ほど高くなるよう、評価関数を事前に設定しておく。

④  $pbest_i$  と  $gbest$  の更新

各粒子  $i$  が新たに得た解の適応度がこれまでの  $pbest_i$  よりも高かった場合、 $pbest_i$  を更新する。近傍内で適応度が最も高い解が  $lbest_i$  よりも高かった場合、 $lbest_i$  をその解で更新する。この操作を全粒子に行う。全粒子の中で最も適応度が高い解が  $gbest$  よりも適応度が高かった場合、 $gbest$  をその解で更新する。

## ⑤速度の更新

(2)式に従い、各粒子の速度の更新を行う。

初期設定を①で行い、②から⑤までの操作を1イテレーションとし、事前に設定したイテレーション数を満たすまで繰り返すことで解空間を探索する。

## 2.2 挿入操作 PSO 戦略

本庄らが提案した IPSO は、PSO に基づき TSP の解空間の探索を行うアルゴリズムである。まず、解空間上に複数の粒子を配置する。これらの粒子は、それぞれ巡回路である解を持っており、各粒子のそれまでの最良解と近傍の粒子の最良解から抽出した部分経路を、各粒子の現在の解に挿入することで解の更新を行い、これを繰り返すことで、解空間を探索する。アルゴリズムの流れの詳細は以下の通りである。

## ①初期設定

各粒子  $i$  に解  $x_i$  をランダムに設定し、各粒子のそれまでの最良解  $pbest_i$  を現在の解  $x_i$  に設定する。粒子  $i$  と粒子  $j$  間の距離  $d_{ij}$  を以下のように定義し、全粒子間の距離を計算する。

$$d_{ij} = \frac{1}{S_{ij}} \quad (3)$$

$$S_{ij} = \frac{|E_i \cap E_j|}{n}$$

ここで、 $E_i$  は粒子  $i$  が持つ解  $x_i$  の経路の集合、 $|E_i \cap E_j|$  は  $E_i$  と  $E_j$  の共通している経路の本数、 $n$  は都市数を表している。距離  $d_{ij}$  は  $x_i$  と  $x_j$  の共通の経路が多くなるほど短くなる。次に、設定した近傍数  $k$  を元に、粒子  $i$  と距離が近い  $k$  個の粒子を粒子  $i$  の近傍に設定する。各粒子  $i$  の近傍の中で総経路長が最も短い解を近傍内の最良解  $lbest_i$  と設定し、全粒子の中で最も総経路長が短い解を全粒子の最良解  $gbest$  と設定する。

## ②解の更新

解  $x_i$  は  $pbest_i$  の部分経路である  $pbest_i'$  と  $lbest_i$  の部分経路である  $lbest_i'$  を総経路長が最も短くなるように挿入す

ることで更新される。粒子  $i$  の解の更新の詳細は以下の通りである。また、9都市の TSP の解の更新の例を図 1 に示す。図 1 の例の  $x_i = (1,4,7,5,6,9,8,3,2)$  は都市 1→都市 4→…→都市 3→都市 2 と都市を巡り、都市 1 に戻る巡回路を表している。

I 部分経路の作成

粒子  $i$  の  $pbest_i$  から、 $p$  本の連続する経路をランダムに抜き出し、部分経路  $pbest'_i$  とする。また、粒子  $i$  の  $lbest_i$  から、 $l$  本の連続する経路をランダムに抜き出し、部分経路  $lbest'_i$  とする。 $p$  と  $l$  は以下の式で表される。

$$p = [c_1 r_1 (n + 1)] \tag{4}$$

$$l = [c_2 r_2 (n + 1)] \tag{5}$$

ここで  $c_1$  と  $c_2$  は  $[0,1]$  を満たすパラメータ、 $r_1$  と  $r_2$  は  $[0,1]$  を満たす一様乱数、 $n$  は都市数である。 $[c_1 r_1 (n + 1)]$  は  $c_1 r_1 (n + 1)$  の整数部分を表している。図 1 の例では  $pbest'_i = (5,4,8,7)$  と  $lbest'_i = (8,9,6)$  を抜き出している。

II  $pbest'_i$  の再形成

$pbest'_i$  と  $lbest'_i$  に共通した都市が含まれていれば、 $pbest'_i$  から該当した都市を削除し、残った都市で総経路長が最も短くなるよう部分経路を再形成する。図 1 の例では都市 8 が共通しているため、 $pbest'_i$  から都市 8 を削除し、 $pbest'_i$  を再形成している。

III  $x'_i$  の形成

$x_i$  に  $pbest'_i$  と  $lbest'_i$  と共通する都市が含まれていれば、 $x_i$  から該当した都市を削除し、残った都市で総経路長が最も短くなるよう巡回路を再形成し、 $x'_i$  とする。図 1 の例では、 $pbest'_i$  と  $lbest'_i$  にある都市 4,5,6,7,8,9 を  $x_i$  から削除し、 $x'_i = (1,2,3)$  となっている。

IV  $pbest'_i$  の挿入

$pbest'_i$  を  $x'_i$  に総経路長が最も短くなるよう挿入する。図 1 の例では都市 1 と都市 3 の間に  $pbest'_i$  を挿入している。

V  $lbest'_i$  の挿入

$lbest'_i$  を  $x'_i$  に総経路長が最も短くなるよう挿入する。図 1 の例では都市 5 と都市 3 の間に  $lbest'_i$  を挿入している。

以上の I ~ V の操作を全粒子で行う。

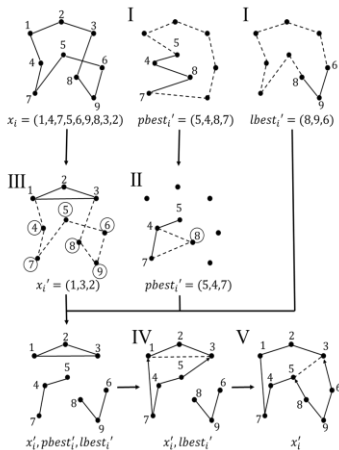


図 1：解の更新の例

③総経路長の計算

全ての粒子が持つ巡回路の総経路長の計算を行う。

④近傍の更新

全粒子間の距離を再計算し、近傍を更新する。

⑤  $pbest$ ,  $lbest$ ,  $gbest$  の更新

粒子  $i$  が新たに得た解  $x'_i$  の総経路長が  $pbest_i$  よりも短かった場合、 $pbest_i$  を  $x'_i$  で更新する。近傍内で総経路長が最も短い解がこれまでの  $lbest_i$  よりも短かった場合、 $lbest_i$  をその解で更新し、この操作を全粒子に行う。全粒子の中で総経路長が最も短い解がこれまでの  $gbest$  よりも短かった場合、 $gbest$  をその解で更新する。

初期設定を①で行い、②から⑤までの操作を 1 イテレーションとし、事前に設定したイテレーション数を満たすまで繰り返すことで TSP の解空間を探索する。

2.3 既存手法の問題点

既存手法である IPSO は、各粒子  $i$  のそれまでの最良解である  $pbest_i$  と、近傍の粒子の最良解である  $lbest_i$  の部分経路である  $pbest'_i$  と  $lbest'_i$  を挿入することで解  $x_i$  の更新を行っている。しかし、この  $pbest'_i$  と  $lbest'_i$  において、抽出する経路の本数と巡回路のどこから経路を抽出するかは乱数によって決定されている。これでは、探索がある程度経った時、 $x_i$  にとっての新規の経路が獲得しにくくなってしまふ。そのため、粒子が厳密解の経路を発見できないまま探索を終了してしまうという問題点がある。

3. 提案手法

提案手法の解の更新は、各粒子  $i$  のそれまでの最良解  $pbest_i$ , 近傍の粒子の最良解である  $lbest_i$ , 最遠の粒子の解  $x_i^f$ , 現在の解  $x_i$  の巡回路を重ね合わせた経路集合  $G$  を用いる。そしてまず、ある都市をランダムに選択する。その都市において、 $G$  に含まれる経路を確率で選択する。選択した経路の次の都市でも同じ操作を行う。これを繰り返すことで巡回路を構築していく。 $G$  から  $pbest_i$  の経路を選択する確率  $P_p$ ,  $lbest_i$  の経路を選択する確率  $P_l$ ,  $x_i^f$  の経路を選択する確率  $P_f$ ,  $x_i$  の経路を選択する確率  $P_x$  はそれぞれ以下の式で表される。

$$P_p = c_1 r_1 \tag{6}$$

$$P_l = c_2 r_2 \tag{7}$$

$$P_f = c_3 r_3$$

$$P_x = 1 - (c_1 r_1 + c_2 r_2) \tag{8}$$

ここで  $c_1, c_2, c_3$  は  $[0,1]$  を満たすパラメータ、 $r_1, r_2, r_3$  は  $[0,1]$  を満たす一様乱数、 $n$  は都市数である。最遠の粒子の解  $x_i^f$  を導入した重ね合わせを解の更新に用いることで、既存手法では得ることができなかった経路の組み合わせを獲得することが考えられる。

結果、考察は発表時に述べる。

参考文献

[1] J.Kennedy, R.C.Eberhart, "Particle swarm optimization" IEEE International Conf. on Neural Networks, pp.1942-1948 (1995)  
 [2] 本庄将也, 飯塚博幸, 山本雅人, 古川正志, "巡回セールスマン問題に対する粒子群最適化の提案と性能評価", 日本知能情報ファジィ学会誌, vol.28, no.4, pp.744-755 (2016)