

粒子群最適化を用いた巡回セールスマン問題の解法

山田 悠希^{†1} 穴田 一^{†1}

概要：工業や経済に関する問題には、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ものが多くある。その組み合わせ最適化問題の中に、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)がある。本研究では、この TSP に粒子群最適化(Particle Swarm Optimization, PSO)を適用させて解くことを目的とする。PSO は生物の群れ行動をモデルにしたアルゴリズムであり、素早く問題の解に到達する多点探索であるという特徴がある。しかし、PSO は実数値最適化手法であるため、そのまま TSP へ適用させるのは難しい。そこで本研究では PSO の特徴を維持しつつ TSP に適用させるアルゴリズムを構築し、TSPLIB に掲載されているベンチマーク問題を用いて、他の手法との性能を比較した。

キーワード：粒子群最適化, 巡回セールスマン問題

An Algorithm for Traveling Salesman Problem using Particle Swarm Optimization

YUKI YAMADA^{†1} HAJIME ANADA^{†1}

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。その組み合わせ最適化問題の中に、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)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、巡回路である解を保持している複数の粒子が、それまでの個々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。そして、TSPLIB に掲載されているベンチマーク問題を用いて既存手法と提案手法の比較をすることで、性能の向上を確認した。

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れ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、評価の高い粒子の情報を近傍の粒子から入手し、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。PSO はこの操作を繰り返すことで解空間を探索するアルゴリズムである。 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。

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

$$v_i(t) = wv_i(t-1) + c_1r_1(pb_{best_i} - x_i(t)) + c_2r_2(lb_{best_i} - x_i(t)) \quad (2)$$

ここで、 w はパラメータ、 c_1 と c_2 は $[0,1]$ のパラメータ、 r_1 と r_2 は $[0,1]$ の一様乱数、 pb_{best_i} は粒子 i のそれまでの最良解、 lb_{best_i} は粒子 i の近傍内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 pb_{best_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 の近傍内で適応度が最も高い解を近傍内の最良解 lb_{best_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), (2)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。

^{†1} 東京都市大学大学院 総合理工学研究科
Graduate School of Integrative Science and Engineering, Tokyo City University

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。また、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ついて行う。最後に、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

(1), (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

本庄らが提案した挿入操作 PSO 戦略 (Insertion-based PSO strategy, 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保持しており、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う。IPSO はこの操作を繰り返すことで解空間を探索するアルゴリズムである。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

各粒子 i に解 x_i をランダムに設定し、各粒子 i の $pbest_i$ を現在の解 x_i に設定する。次に、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ように定義されている。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 の近傍に設定し、その中で総経路長が最も短い解を近傍内の最良解 $lbest_i$ と設定する。最後に、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。またここでは、9都市の TSP の解の更新の例 (図1) を共に示す。図1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。

$$p = [c_1 r_1 (n + 1)] \quad (4)$$

$$l = [c_2 r_2 (n + 1)] \quad (5)$$

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を削除した後、残った都市を最も短くなるよう繋ぎなおして、 $pbest_i' = (5,4,7)$ を再形成している。

III : x_i' の形成

x_i に $pbest_i'$ と $lbest_i'$ に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ことで、解の更新を行う。

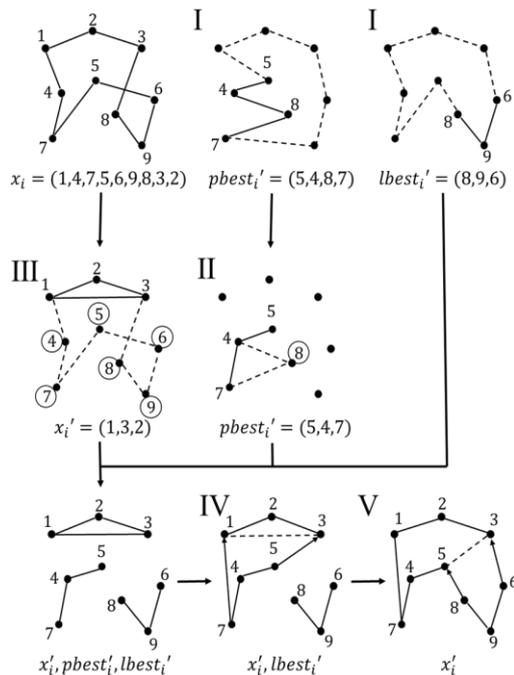


図1：解の更新の例

③総経路長の計算

各粒子の巡回路の総経路長の計算を行う。

④近傍の更新

各粒子間の距離を再計算し、近傍を更新する。

⑤pbest, lbest, gbestの更新

各粒子 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。また、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、 $lbest_i$ をその解で更新する。この操作を全粒子に行う。最後に、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 $pbest_i$ の部分経路である $pbest_i'$ と、近傍の粒子の最良解である $lbest_i$ の部分経路である $lbest_i'$ 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 $lbest_i'$ は探索がある程度経った時、経路のほとんどが x_i と同じ巡回路となってしまう。それらの解から抽出した部分経路は既に x_i が保持している経路である可能性が高く、新規の経路の獲得が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 $pbest_i$ 、近傍内の最良解である $lbest_i$ 、そして、最遠の粒子の解である x_i^f を現在の解 x_i に重ね合わせた経路集合 G_i を用いる。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_i 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。この時、 G_i 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。

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

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

$$P_f = c_3 \tag{8}$$

$$P_x = 1 - (c_1 + c_2 + c_3) \tag{9}$$

ここで c_1, c_2, c_3 は $[0,1]$ を満たすパラメータであり、これらの和は1より小さくなるよう設定している。 G_i に含まれる経路が全て訪問済みであった場合、未訪問都市の中から最も近い都市の経路を選択する。この操作を繰り返すことで巡回路を構築し、解の更新を行う。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できると思われる。

4. 結果

提案手法の有用性を確認するため、TSPLIBに掲載されているTSPのベンチマーク問題であるrd100, kroA150, pr299を用いて評価実験を行った。既存手法は実際にアルゴリズムを再現し、事前実験で最も結果が良かった粒子数 $m = 64$ 、近傍数 $k = 2$ 、 $c_1 = 0.9$ 、 $c_2 = 0.1$ というパラメータを使用した。また、提案手法における c_1, c_2, c_3 もまた、事前実験で最も結果が良かった $c_1 = 0.6$ 、 $c_2 = 0.2$ 、 $c_3 = 0.1$ を使用した。終了条件はrd100とkroA150は30000イテレーション、pr299は500000イテレーションとした。各問題50試行平均の結果を表1~3に表す。また、表中で用いられている誤差率は、試行内で得られた最良解の厳密解に対する誤差の割合を表し、最終更新イテレーションは最後に $gbest$ を更新したイテレーションを表している。

表1:rd100の50試行の結果

rd100(厳密解 7910)	既存	提案
厳密解到達率(%)	88	98
平均誤差率(%)	0.0099	0.0002
平均最終更新イテレーション	3210.4	15565.66

表 2:kroA150 の 50 試行の結果

kroA150(厳密解 26524)	既存	提案
厳密解到達率(%)	10	20
平均誤差率(%)	0.25	0.18
平均最終更新 イテレーション	6951.2	27379.68

表 3:pr299 の 50 試行の結果

pr299(厳密解 48191)	既存	提案
厳密解到達率(%)	0	2
平均誤差率(%)	0.91	0.29
平均最終更新 イテレーション	254626.8	377088.3

実験の結果, 全ての問題において, 既存手法よりも提案手法の精度が上回ったことが分かる. しかし, 平均最終更新イテレーションは提案手法の方が軒並み長くなっている. これは, 提案手法の方が既存手法よりも広範囲で解を探索しているため, 解の収束が遅くなっていることが理由であると考えられる.

5. 今後の課題

今後の課題として, 解の更新の見直しをする必要があることが挙げられる. 提案手法における pr299 の厳密解カバー率の推移を表したグラフを図 1 に表す. 厳密解カバー率とは, 全粒子の経路を合わせて厳密解の経路をどれだけ保持しているかを割合で表したものである.

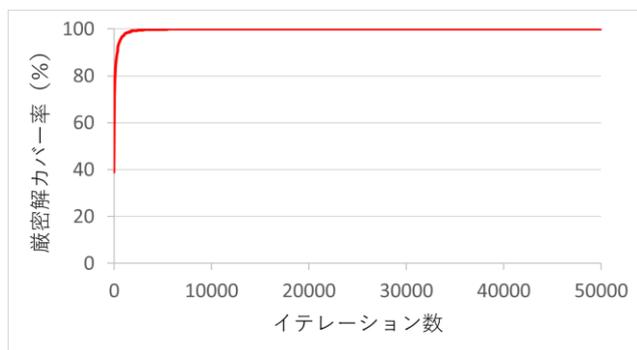


図 1: 提案手法における pr299 の厳密解カバー率

図 1 より, 2000 イテレーションほどで厳密解カバー率が 100%になり, その後も 100%を維持していることが分かる. 厳密解の経路を全て保持しているにも関わらず, 厳密解に収束しないということは, 解の探索範囲は拡大し, 全粒子が多様な経路を保持しているものの, その組み合わせが効率よく行われていないことが考えられる. そこで, 巡回路を重ね合わせて経路を選択する際に, 距離が短い経路を選択しやすくする, といった効率のよい解の組み合わせ方法を考案し, 更なる大規模問題に挑戦していきたいと考えて

いる.

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).