

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出来るものが多い。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 に高速に良い解を発見出来るということが知られている粒子群最適化(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その有効性を確認した。

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出来る。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、実数値最適化問題に用いられるヒューリスティクスの一つ、粒子群最適化(Particle Swarm Optimization, PSO) [1]に注目した。粒子群最適化とは群知能の一種であり、解空間上に配置された複数の粒子が互いに情報共有を行いながら解空間の探索を行うアルゴリズムであり、短時間で良い解に到達するという特徴がある。この特徴に着目した本庄らは PSO 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、PSO のアルゴリズムの特徴を基に、新たなアルゴリズムを構築した。この提案手法は、PSO で用いられている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、解の更新を行うよう設計した。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を比較することで、その有効性を確認した。

2. 既存手法

2.1 巡回セールスマン問題

巡回セールスマン問題(Traveling Salesman Problem, TSP)とは、与えられた都市を全て巡る最短の巡回回路を求める問題である。 n 都市問題における TSP の解候補数 S_n は以下の式で表せる。

$$S_n = \frac{(n-1)!}{2} \quad (1)$$

ここで、 n は都市数である。(1)式で表される様に、都市数が増加すると、解候補数は爆発的に増加する。例えば、50 都市問題の TSP は解候補数が約 10 の 62 乗となり、全ての解を探索する列挙法では現実的な時間内で厳密解を求めることが出来ない。そのため、TSP の様な、列挙法では厳密解を求めることが難しい組み合わせ最適化問題に対し、解の精度は保証されないが、短い時間で解を導くことが出来る遺伝的アルゴリズム(Genetic Algorithm, GA) [3][4]や、蟻コロニー最適化(Ant Colony Optimization, ACO) [5][6]、粒子群最適化(Particle Swarm Optimization, PSO)[1]などのヒューリスティクスの研究が盛んに行われている。

2.2 粒子群最適化

粒子群最適化(Particle Swarm Optimization, PSO)とは、1995 年に Kennedy らが提案した、魚や鳥などに見られる群れ行動を探索手法に応用した、実数値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

^{†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

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、全粒子の中で最も適応度の高い粒子の情報を全粒子で共有し、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。

$$x_i(t) = x_i(t-1) + v_i(t-1) \quad (2)$$

$$v_i(t) = wv_i(t-1) + c_1r_1(pb_{best_i} - x_i(t)) + c_2r_2(g_{best} - x_i(t)) \quad (3)$$

ここで w, c_1, c_2 は [0,1] のパラメータ, r_1, 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, g_{best} は全粒子中の最良解である。アルゴリズムの詳細な流れ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 の最良解 pb_{best_i} を現在位置に設定する。次に、全粒子の中で適応度が最も高い解を g_{best} と設定する。

②位置の更新

(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。

④ pb_{best}, g_{best} の更新

全粒子の pb_{best} と g_{best} を更新する。

⑤速度の更新

(3)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繰り返すことで解空間を探索する。

また、粒子間の情報共有を制限する PSO[7]も提案されている。この PSO において、各粒子は近傍にいる粒子とのみ情報共有をすることが出来る。この情報共有を制限した PSO における t イテレーション目における位置 $x_i(t)$ と速度 $v_i(t)$ の更新式は以下のようになる。

$$x_i(t) = x_i(t-1) + v_i(t-1) \quad (4)$$

$$v_i(t) = wv_i(t-1) + c_1r_1(pb_{best_i} - x_i(t)) + c_2r_2(lb_{best_i} - x_i(t)) \quad (5)$$

ここで w, c_1, c_2 は [0,1] のパラメータ, r_1, 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 の近傍内の最良解である。

2.3 挿入操作 PSO 戦略

PSO は実数値最適化手法の一つであるため、TSP に適用させるためには様々な工夫が必要である。例えば Swap Operator という交換子を用いる PSO[8][9]や、Fuzzy 行列を用いる PSO[10]などがある。その中で、本庄らは各粒子が持つ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使い、解更新を行う挿入操作 PSO 戦略(Insertion-based PSO strategy, IPSO)を提案した。解空間上に配置された複数の粒子は、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。IPSO のアルゴリズムの詳細な流れ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、各粒子の最良解 pb_{best_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。

$$d_{ij} = \frac{1}{S_{ij}} \quad (6)$$

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

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 の異なる経路が多くなるほど長くなる。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 lb_{best_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 g_{best} と設定する。

②解の更新

解 x は pb_{best} の部分経路である pb_{best}' と lb_{best} の部分経路である lb_{best}' を最も総経路長が短くなるように挿入することで更新される。粒子 i の部分経路の挿入の詳細は以下の通りである。解更新は以下の I~V をもって行われる。図 1 に 9 都市の場合の粒子 i が解更新に用いる巡回路の一例を示す。以下、図 1 の経路を用いて解更新の解説をしていく。

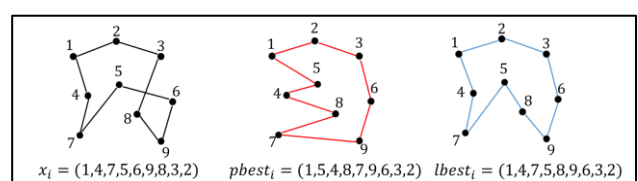


図1 ある粒子 i が保持している解を表した図

図1の黒点は都市を、数字は都市番号を表しており、黒点を結ぶ実線はその都市を結ぶ経路を表している。また、黒線で結ばれた巡回路は現在の解 x_i 、赤線で結ばれた巡回路は $pbest_i$ (これまでの最良解)、水色で結ばれた巡回路は $lbest_i$ (近傍内のそれまでの最良解) を表している。また、 $x_i = (1,4,7,5,6,9,8,3,2)$ は都市1→都市4→都市7→…→都市3→都市2と行って、最後に都市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。

$$p = [c_1 r_1 (n + 1)] \quad (7)$$

$$l = [c_2 r_2 (n + 1)] \quad (8)$$

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2の例では $pbest'_i = (5,4,8,7)$ と $lbest'_i = (8,9,6)$ を抜き出している。

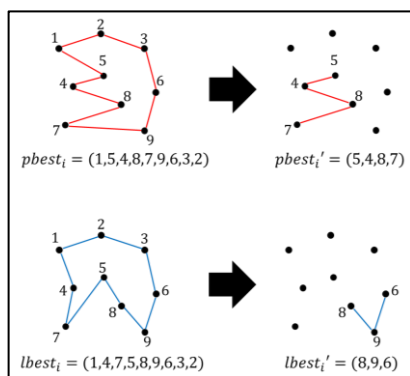


図2 部分経路の作成

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3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 $pbest''_i$ に再形成している。

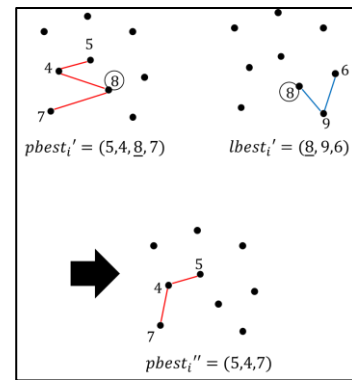


図3 $pbest'_i$ の共通都市の削除

III x_i の再形成

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4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 を $x'_i = (1,2,3)$ に再形成している。

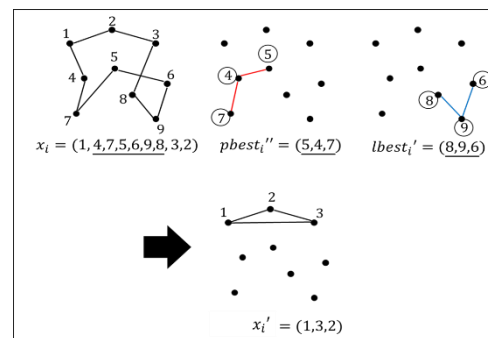


図4 x'_i の再形成

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5の例では都市1と都市3の間に $pbest''_i$ を挿入している。

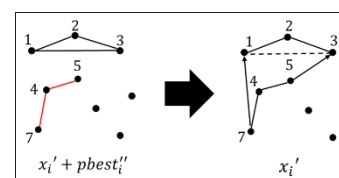


図5 $pbest''_i$ の挿入

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6の例では都市5と都市3の間に $lbest'_i$ を挿入している。

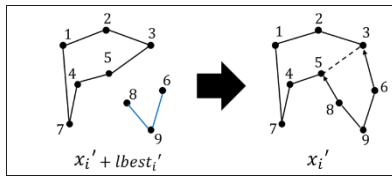


図6 $lbest_i^f$ の挿入

以上の 1 ~ V の操作を全粒子で行う。

③総経路長の計算

各粒子が持つ巡回路の総経路長を求める。

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

全粒子の $pbest, lbest$ と $gbest$ を更新する。

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繰り返すことでTSPの解空間を探索する。

2.4 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 $pbest_i$ と近傍内の最良解である $lbest_i$ の部分経路を挿入することで解 x_i の更新を行っている。しかし、 $pbest_i$ と $lbest_i$ による部分経路挿入は、それぞれの粒子が待つ解 x_i と似たような巡回路の部分経路挿入を繰り返すだけで、解探索が終了してしまう。これでは初期設定時の離れた粒子の解が持つ経路の組み合わせを試せないまま収束してしまい、結果として探索が十分に行われないうちに局所解に陥ってしまう。

3. 提案手法

本提案手法では、既存手法において部分経路挿入で行っていた『②解の更新』を、各粒子 i のそれまでの最良解 $pbest_i$ 、近傍の粒子の解である $lbest_i$ 、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、それぞれの解の経路に重み $c_1 \sim c_3$ をかけて作成した経路集合 G_i を用いて行う。まず、ある都市 a をランダムに選択する。そして G_i に含まれる経路から、次式で定義される確率 P_{ab} で次の経路 ab を選択する。

$$P_{ab} = \frac{w_{ab}}{\sum_{l \in N'} w_{al}} \quad (l \in N') \quad (9)$$

$$w_{ab} = \frac{G_i^{ab}}{(d_{ab})^D}$$

$$G_i^{ab} = x_i + c_1 pbest_i + c_2 lbest_i + c_3 x_i^f$$

ここで、 N' は未訪問都市、 $D, c_1 \sim c_3$ はパラメータを表している。(9)式は距離が短く、 G_i の重みが大い経路を選択しやすくなるように設定している。また、この経路集合

G_i のパラメータは、それぞれの粒子が自分の解の周辺を探索するため、 $1 \geq c_1 \geq c_2 \geq c_3$ を満たすように設定している。 G_i に選択出来る経路が存在しない場合、未訪問都市の経路候補の中から距離の逆数を用いたルーレット選択を用いて経路を選択する。これらの操作を繰り返すことで巡回路を構築していく。

4. 結果

提案手法の有効性を確認するため、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。

IPSO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、提案手法のパラメータもまた、事前実験で最も結果が良かった $c_1=0.6$ 、 $c_2=0.3$ 、 $c_3=0.1$ を使用した。終了条件についてはrd100とkroA150を30000イテレーション、pr299を500000イテレーションと設定した。各問題50試行平均の結果を表1~表3に示す。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

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 7 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。

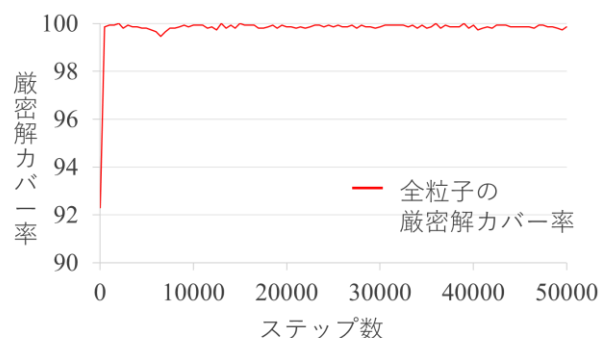


図 7 提案手法における pr299 の厳密解カバー率

図 7 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、厳密解に到達しないということは、解の探索範囲は拡大し、全粒子を参照すると多様な経路を保持しているものの、各粒子における経路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、様々な粒子の情報を組み合わせる仕組みを考案することにより、効率よく解空間を探索するアルゴリズムを構築し、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uzzy学会誌, vol.28, no.4, pp.744-755 (2016).
- [3] J. H. Holland, Adaption in Natural and ArtiJicial Systems, Ann Arbor:Univ. Michigan Press (1975).
- [4] 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).
- [5] M. Dorigo, V. Maniezzo, and A. Colormi, "Ant system: optimization by a colony of cooperating agents," IEEE Trans. Syst. Man. Cybern., B, Cybern., vol.26, no.1, pp.29-41 (1996)

- [6] T. Stutzle and H.H. Hoos, "MAX-MIN ant system," Future Generation Computer Systems, vol.16, no.9, pp.889-914 (2000).
- [7] Russ C Eberhart, James Kennedy, et al. A new optimizer using particle swarmtheory. In *Proceedings of the sixth international symposium on micro machineand human science*, Vol. 1, pp. 39-43, New York, NY (1995).
- [8] K. Wang, L. Huang, C. Zhou, and W. Pang. Particle swarm optimization for traveling salesman problem. In *Proceedings of International Conference on Ma-chine Learning and Cybernetics*, pp. 1583-1585 (2003).
- [9] Xiaohu H Shi, Yanchun Chun Liang, Heow Pueh Lee, C Lu, and QX Wang. Particle swarm optimization-based algorithms for tsp and generalized tsp. *Information Processing Letters*, Vol. 103, No. 5, pp. 169-176 (2007).
- [10] Wei Pang, Kang Ping Wang, Chun Guang Zhou, and Long Jiang Dong. Fuzzy discrete particle swarm optimization for solving traveling salesman problem. In *Proceedings of the fourth International Conference on Computer and Information Technology*, 2004. CIT'04., pp. 796-800, IEEE (2004).