The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

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 Solving the Traveling Salesman Problem using Particle Swarm Optimization

山田 悠希[*1]　　穴田 一[*1]
Yuki Yamada　　Hajime Anada

[*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 #1

Many economic and industrial problems lead to combinatorial optimization problems. Of these combinatorial optimization problems, the traveling salesman problem (TSP) is one of the most important problem in the field of technology and science. And particle swarm optimization (PSO) is a population based stochastic optimization technique inspired by social behavior of bird flocking or fish schooling. PSO has been applied to various combinatorial optimization problems belonging to nondeterministic polynomial-time hard (NP-hard) combinational problems. But applying PSO to TSP is difficult. Therefore, we construct a new algorithm which is based on PSO. We confirmed the effectiveness of our algorithm using several benchmark problems taken from the TSPLIB, which is a library of traveling salesman problem.

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) 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) [Kennedy 95]を TSP 向けに改良した挿入操作 PSO 戦略（Insertion-based PSO strategy, IPSO) [本庄 16]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．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. 既存研究

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．

$$x_i(t) = x_i(t-1) + v_i(t-1) \tag{1}$$

$$v_i(t) = wv_i(t-1) + c_1r_1\left(pbest_i - x_i(t)\right) + c_2r_2\left(lbest_i - x_i(t)\right) \tag{2}$$

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，$pbest_i$は粒子 i のそれまでの最良解，$lbest_i$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$pbest_i$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 の近傍内で適応度が最も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), (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$pbest_i$と$gbest$の更新
全粒子の $pbest, lbest$ と $gbest$を更新する．
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．

連絡先：山田悠希，東京都市大学，〒158-8557
東京都世田谷区玉堤 1-28-1

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．

$$d_{ij} = \frac{1}{S_{ij}} \tag{3}$$

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

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 $pbest'$ 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，$pbest_i' = (5,4,8,7)$ から都市 8 を削除し，$pbest_i' = (5,4,7)$を再形成している．

Ⅲ x_i' の形成

x_i から $pbest_i', lbest_i'$ と共通する都市を削除し，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，$x_i' = (1,3,2)$としている．

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．

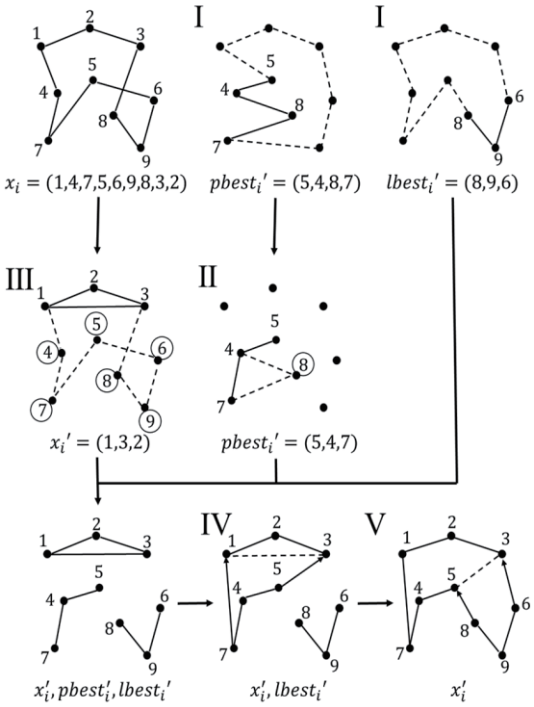

図 1：解の更新の例

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

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．

2.3 既存手法の問題点

既存手法において，解 x の更新は各粒子の今までの最良解 $pbest$と近傍の最良解 $lbest$ を用いて行われる．近傍数が 2 である場合，近くにいる 2 粒子のうち良い粒子の解が$lbest$となるため，それぞれの粒子が参照できる粒子の情報が少ない．これでは，離れた粒子との組み合わせを試さないうちに探索が終了してしまい，全粒子の初期解の周辺を探索する多点探索とは言えない．

3. 提案手法

提案手法における解の更新は，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 に重ね合わせた経路集合 G を用いて行わ

れる．まず，ある都市 i をランダムに選択する．そして G に含まれる経路から，次式で定義される確率P_{ij}で次の経路 ij を選択する．

$$P_{ij} = \frac{w_{ij}}{\sum_{l=1}^{n} w_{il}} \tag{6}$$

$$w_{ij} = \frac{c}{(a_{ij})^{D}}$$

$$C = C_1 + C_2 + C_3 + C_4$$

$$C_1 = \begin{cases} c_1 & x_i\text{に含まれる場合} \\ 0 & otherwise \end{cases}$$

$$C_2 = \begin{cases} c_2 & pbest_i\text{に含まれる場合} \\ 0 & otherwise \end{cases}$$

$$C_3 = \begin{cases} c_3 & lbest_i\text{に含まれる場合} \\ 0 & otherwise \end{cases}$$

$$C_4 = \begin{cases} c_4 & x_i^f\text{に含まれる場合} \\ 0 & otherwise \end{cases}$$

ここで，n は都市数，D, c_1～c_4 はパラメータを表している．(6)式は G に含まれる経路のうち，距離が短く，複数の解に含まれる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 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，c_1が最も高くなるように設定している．G に選択できる経路が存在しない場合，未訪問都市の経路の中から距離情報を用いたルーレット選択を用いて経路を選択する．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．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，c_4もまた，事前実験で最も結果が良かった$c_1 = 0.6$，$c_2 = 0.2$，$c_3 = 0.1$，$c_4 = 0.1$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．提案手法を用いて pr299 の厳密解カバー率の計算を 50 試行行い，その平均の推移を図 2 に示す．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．

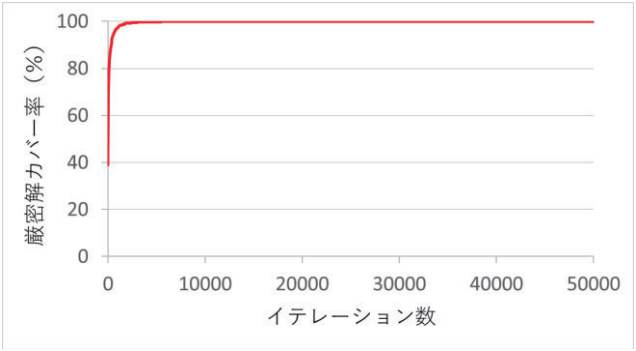

図 2：提案手法における pr299 の厳密解カバー率

図 2 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，様々な粒子の情報を参照するような経路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

[Kennedy 95] J.Kennedy , R.C.Eberhart , : ”Particle swarm optimization”IEEE International Conf. on Neural Networks, pp.1942-1948 (1995)

[本庄 16] 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)