

粒子群最適化による巡回セールスマン問題の解探索高速化

日大生産工 (学部) ○弘田 秋穂 日大生産工 山内 ゆかり

1 まえがき

数値最適化手法の一つに粒子群最適化 (Particle Swarm Optimization, 以下PSO)がある[1]。PSOは、複数の粒子と言う探索単位が自身のそれまでの探索で発見した最良解 (*pbest*)と群全体の中での最良解 (*gbest*)の情報を共有しながら探索を進める。*pbest* と *gbest* への引力が各粒子に働くため、これまでの良い解の近傍を探索する傾向がある。

一方、組み合わせ最適化問題の一つに巡回セールスマン問題 (Traveling Salesman Problem, 以下 TSP)がある。TSP は、訪問すべき全ての都市を一度ずつ通り、出発した都市に戻る経路を最も小さくする問題である。

これまでに、TSPに適応させたPSOが提案されてきた。PSOは、アルゴリズムが単純であり、解を実用的な時間内に計算されやすく、多くの大域的最適化問題へ応用される。しかし、PSOはTSPに適用することは難しく、解を得るのにとっても時間の掛かる非実用的な適用法が多かった。

対して本庄らは実用的な時間や計算空間量でTSPを解くPSOに基づいた手法の挿入操作 PSO戦略 (Insertion-based PSO strategy, 以下IPSO)を提案した[2]。IPSOは、解候補を巡回路で、探索単位を部分経路で表すことで、解更新を部分経路の挿入によって行い、高い解精度を得られた。

本研究では、従来研究の部分経路挿入時の計算量を軽量化し、解精度を下げずに高速化することを目的とする。

2 従来研究

2.1 PSO概要

先に記した様に、PSOは実数値最適化手法の1つである。PSOは、鳥や魚等の群で見られる群行動を応用した探索手法である。解空間内に複数の粒子が存在し、それらが作用しながら探索を進める。その内の1つの粒子である*i*は位置 x_i と速度 v_i の*n*次元の実数値ベクトルを持つ。ステップ*t*における粒子*i*の位置と速度の更新は以下の(1)式及び(2)式で定義される。

$$v_i(t) = wv_i(t-1) + c_1r_1(p_i - x_i(t)) + c_2r_2(g - x_i(t)) \quad (1)$$

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

c_1, c_2, w は定数で、 r_1, r_2 は、*i*に対して独立な乱数[0,1]である。また、 p_i は、その粒子*i*がこれまでに探索した中の最良の位置 *pbest*、*g*は群全体がこれまでに発見した最良の位置 *gbest*を表す。

2.1 IPSOにおける解表現

IPSOは、PSOで用いる群全体の中での最良解 *gbest*の代わりに、粒子*i*と近傍の粒子の最良解 *lbest*を用いる[3]。各粒子は *pbest*と *lbest*を用いて解探索を行う。粒子*i*について、その位置 x_i 、*pbest* を p_i 、*lbest* を l_i とする。

解表現について記す。都市0,1,2,3,4,5,6,7,8からなる9都市のTSPを例とする。解 $x_i = (0,1,2,3,4,5,6,7,8)$ は、0,1,2,3,4,5,6,7,8,0の順で都市を回ることを表す。また、TSPには始点都市と終点都市の区別をつけないため、例えば(4,5,6,7,8,0,1,2,3)等と記しても同じ巡回路である。また、部分経路とは巡回路内の連続した辺である。解 x_i の部分経路は、例えば(3,4,5)等がある。また例えば(0,3,4)は、解 x_i では0都市の一つ前後順の都市は1か8であるため、0,3もしくは0,4と順に行くこの例(0,3,4)は、 x_i の部分経路ではない。

IPSOに用いる部分経路の挿入は、巡回路 x_i から、二つの経部分経路 p_i' と l_i' を抽出し、残りの都市からなる閉路 x_i に挿入を行うことで解更新を行う。

2.2 部分経路挿入法アルゴリズム

巡回路 x_i と、その部分経路である p_i' と l_i' があるとき、部分経路挿入法アルゴリズムを以下に記す。

1. p_i' に、 l_i' に含まれる都市があるとき、その都市を p_i' から除き、新たな部分経路 p_i'' とする。 p_i' と l_i' に重複都市がない場合は p_i'' は p_i' と同じ部分経路となる。

Efficient Particle Swarm Optimization with Tour Representation for Solving Traveling Salesman Problems

Akiho HIROTA and Yukari YAMAUCHI

2. 手順1と同様に、 x_i から p_i'' と l_i' に含まれる都市を除いた閉路を x_i' とする。
3. x' のある辺の都市間に p_i'' を挿入し、 x'' を生成する。このとき挿入される辺は、他のどの辺に挿入された場合よりも x'' の総経路長が小さくなるものである。
4. 手順3と同様に、生成される x_i'' の総経路長が最小となる様に x_i'' の辺の間に l_i' を挿入する。また、この時選ばれる x_i'' の辺は、 p_i'' に含まれていない辺とする。

また、手順3と手順4において部分経路の挿入を行うとき、2通りの方法がでる。例えば、 $e(v_1, v_2)$ を両端に都市 v_1, v_2 を持つ挿入される辺とし、挿入する部分経路の両端の都市に v_3, v_4 があるとする。このとき生成される辺は、 $e(v_1, v_3)$, $e(v_4, v_2)$ の場合と $e(v_1, v_4)$, $e(v_3, v_2)$ となる場合がある。このうち生成される閉路の総経路長が小さくなる場合を選択する。

2.3 IPSOアルゴリズム

2.2の部分経路挿入法をPSOに導入させたIPSOのアルゴリズムを以下に示す。都市数を n 、 c_1, c_2 はそれぞれ $0 \leq c_1 \leq 1, 0 \leq c_2 \leq 1$ を満たす定数とする。また r_1, r_2 はそれぞれ $0 \leq r_1 < 1, 0 \leq r_2 < 1$ を満たす、 i に対して独立な乱数とする。 $C(x)$ は、巡回路 x の総経路長を返す関数とする。また $a = b$ と記述されたとき、巡回路 a を巡回路 b に置き換えることを表す。また $[d]$ は、実数 d の整数部分を表す。

1. 全ての粒子 i に対し、適当な実行可能解を x_i に設定する。また、 $p_i = x_i$ とする。粒子 i の近傍 A_i の $pbest$ の集合 A_i^p の中で最もその総経路長が小さいものを l_i とする。この数式を(3)式に示す。

$$l_i = \underset{x \in A_i^p}{\operatorname{argmin}} C(x) \quad (3)$$

2. 全ての粒子 i について、以下を行う。
 - a. 長さ $[c_1 r_1 (n + 1)]$ の連結成分を p_i からランダムに選択し、部分経路 p_i' とする。
 - b. 長さ $[c_2 r_2 (n + 1)]$ の連結成分を l_i からランダムに選択し、部分経路 l_i' とする。
 - c. 部分経路挿入法を用いて x_i に p_i' と l_i' を挿入し、新たな x_i とする。
3. 全ての粒子 i について、以下を行う。
 - a. $C(x_i) < C(p_i)$ の場合は $p_i = x_i$ とする。
 - b. (3)式通りに l_i を更新する。

4. 全ての粒子の $pbest$ の中で総経路長の最も小さいものを解 s にする。
5. 終了条件の判定を行う。条件を満たしていない場合は手順2へ戻る。
6. 解 s を得られた解とする。

1ステップを手順2、手順3を1度行うこととし、手順5における終了条件は、ステップの経過や、制限時間により設定される。

3 提案手法

本研究は、従来手法の高速化を目指す。提案手法の部分経路の挿入を行う過程において、探索中の辺で挿入された場合に生成された閉路の総経路長を求めながら行っていた。本研究では、経路挿入を、部分経路の端の都市から最も近い位置にある閉路内の辺に挿入を行う。挿入を行う辺の探索間は、総経路長の計算を行わないことで、計算量を小さくさせる。以下に、提案手法の部分経路挿入法のアルゴリズムを示す。また、提案手法と従来手法との違いは2.2に示したアルゴリズムの手順3、手順4に対応する箇所となる。

2.2同様に、巡回路 x_i と、その部分経路である p_i' と l_i' があるとき、以下に提案手法の部分経路挿入のアルゴリズムを示す。

1. p_i' に、 l_i' に含まれる都市があるとき、その都市を p_i' から除き、新たな部分経路 p_i'' とする。 p_i' と l_i' に重複都市がない場合は p_i'' は p_i' と同じ部分経路となる。
2. 手順1と同様に、 x_i から p_i'' と l_i' に含まれる都市を除いた閉路を x_i' とする。
3. それぞれ都市が重複しない、閉路 x' と2つの部分経路 p_i'', l_i' が存在するとき、 x' のある辺の都市間に p_i'' を挿入し、 x'' を生成する。部分経路 p_i'' の端の都市から最も近距離にある x' の都市を探索する。この時 p_i'' の端の都市は2つ存在するが、より近くの x' 内の都市との距離が小さい方を先に連結させ、 x'' を生成する。
4. 手順1と同様に l_i' の端都市と近距離の x'' 内の都市を探索し、 x'' に l_i' を挿入し、 x''' を生成する。

4 数値計算実験

4.1 実験条件

本実験で用いるパラメータの説明を行う。粒子数 m 、 $pbest$ の重み c_1 、 $lbest$ の重み c_2 、また、各粒子の近傍数 k がある。探索解の値は、これらのパラメータの設定に影響される。そのため、

本実験は、以下のパラメータの組み合わせごとに行う。

表1. 本実験のパラメータ設定

粒子数 m	32
$pbest$ の重み c_1	{0.2,0.7}
$lbest$ の重み c_2	{0.1,0.8}
近傍数 k	{2, $m - 1$ }

使用するTSPは[4]内に掲載される100都市のTSPであるkroA 100とする。また、20000ステップ経過を探索の終了条件とする。

次に、性能の評価を行うための特徴量である、誤差 e_s について説明する。誤差 e_s は、学習により得られた解 s の総経路長 $C(s)$ と、厳密解 s_e の総経路長 $C(s_e)$ との誤差を表し、(4)式で定義する。

$$e_s = \frac{C(s) - C(s_e)}{C(s_e)} \quad (4)$$

誤差 e_s は、解 s が厳密解 s_e と一致する場合0となり、 e_s が小さい程良い解が探索されたこととなる。

4.2 実験結果

パラメータの組ごとの従来手法と提案手法の比較を表2に示す。本研究の提案手法では、従来手法の高速化を目指すため、実行処理時間も記録する。

表2. 各実験における誤差 e_s とその処理時間

$m = 32, c_1 = 0.7, c_2 = 0.8, k = m - 1$		
	誤差 e_s	処理時間[s]
従来研究	0.034818	325
提案手法	0.025843	262
$m = 32, c_1 = 0.7, c_2 = 0.1, k = 2$		
	誤差 e_s	処理時間[s]
従来研究	0.025515	304
提案手法	0.019077	238

$m = 32, c_1 = 0.2, c_2 = 0.1, k = m - 1$		
	誤差 e_s	処理時間[s]
従来研究	0.000000	356
提案手法	0.637346	234
$m = 32, c_1 = 0.2, c_2 = 0.1, k = 2$		
	誤差 e_s	処理時間[s]
従来研究	0.000000	356
提案手法	0.637346	234

$m = 32, c_1 = 0.7, c_2 = 0.1, k = m - 1$		
	誤差 e_s	処理時間[s]
従来研究	0.034818	325
提案手法	0.025843	218
$m = 32, c_1 = 0.7, c_2 = 0.1, k = 2$		
	誤差 e_s	処理時間[s]
従来研究	0.025515	304
提案手法	0.019077	238

表2の実験結果より、どのパラメータの組での実験でも提案手法の方が短くなった。また誤差 e_s について、近傍数 $k = 2$ の場合と $k = m - 1$ の場合いずれも、($c_1 = 0.7, c_2 = 0.8$), ($c_1 = 0.7, c_2 = 0.1$)では、従来手法より提案手法の方が小さい値となった。提案手法において、従来手法からの解精度の保持を行えていると言える。

しかし($c_1 = 0.2, c_2 = 0.1$)の場合の e_s 値は、その組み合わせ内の提案手法と従来手法のみならず、他の実験での値からも差が大きく開いた。本研究の $pbest$ の重み及び $lbest$ の重みは、探索範囲の限定及び局所最適解に陥りにくくするバランスに影響させる。($c_1 = 0.2, c_2 = 0.1$)の場合での e_s 値も、 c_1, c_2 値の影響を受けた結果と考えられる。

この結果から、より良い解 s を探索するため $pbest$ の重み c_1 、 $lbest$ の重み c_2 の最適な値を見つける必要もある。また、本実験の試行回数が少ないことも、結果の差が目立つ要因であると考えられる。より十分な試行を行い、誤差 e_s ではなく、平均誤差 $\bar{e}_s$ として結果の比較を行う必要がある。

5 まとめ

本研究では、巡回セールスマン問題解くための粒子群最適化手法の1つである、IPSOの高速化の手法の提案を行った。IPSOは、粒子の位置を巡回路とし、その粒子の最良解 $pbest$ とリング型の近傍粒子の最良解 $lbest$ を部分経路挿入法により、巡回路に挿入することで、解更新を行う手法であり、厳密解に近い解を得られやすい手法である。

提案手法は、従来手法の高速化を目的とする。従来手法の部分経路挿入間において、ある辺に挿入された場合に生成される巡回路の総経路長を、挿入可能な辺全通りにおいて計算を行っていた箇所に軽量化の余地があった。この改善に、提案手法は、部分経路の挿入アルゴリズム

を、部分経路の端都市から近傍距離に位置する巡回路の都市から行うものとした。

また、従来手法と提案手法との比較に実験を行った結果、学習が終わるまでの処理時間を短くすることができた。しかし、*pbest*の重み c_1 や、*lbest*の重み c_2 等の設定されるパラメータの値とその組み合わせによっては、局所最適解に陥りやすくなり、解精度を維持できない結果となった。

今後は一定の学習回数間で同じ*pbest*が続く場合に行う処理を実装し、短時間でより良い探索解を得られることを目指す。また、実験の試行回数を増やし、解探索に影響するパラメータの組み合わせを増やし、最適なパラメータの値を見つける必要がある。

「参考文献」

- [1] J.Kennedy and R.C. Eberhart, Particle swarm optimization, Proceedings of IEEE International, Conference on Networks, 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, 知能と情報(日本知能情報ファジイ学会誌) Vol.28, No.4, pp.744-744, 2016
- [3] R.C Eberhart and J.Kennedy ,A new optimizer using particle swarm theory, Proceedings of IEEE International Symposium on MicroMachine and Human Science, 39-43, 1995
- [4] TSPLIB, <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/> (2017 年 10 月 19 日)