



HOKKAIDO UNIVERSITY

Title	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
Author(s)	本庄, 将也; 飯塚, 博幸; 山本, 雅人 他
Citation	知能と情報, 28(4), 744-755 https://doi.org/10.3156/jsoft.28.744
Issue Date	2016-08
Doc URL	https://hdl.handle.net/2115/64574
Type	journal article
File Information	28_744.pdf



原著論文

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[†]

本庄 将也^{*1}・飯塚 博幸^{*1}・山本 雅人^{*1}・古川 正志^{*2}

近年注目されている実数値最適化手法の一つに粒子群最適化 (Particle Swarm Optimization, PSO) がある。PSO は群知能の一種であり、複数の探索単位 (粒子) が互いに情報共有を行いながら解の探索を行う。多点探索を行うメタヒューリスティクスとしては遺伝的アルゴリズム (Genetic Algorithm, GA) が有名であるが、多くの実数値最適化問題において PSO のほうが GA に比べて高速に良い解を発見できることが知られている。本研究では、組合せ最適化問題の一種である巡回セールスマン問題 (Traveling Salesman Problem, TSP) に対して短時間で良い解を得ることを目的として、PSO を基にしたアルゴリズムである挿入操作 PSO 戦略を提案する。提案手法では、粒子の解候補は実数値ベクトルではなく巡回路として表現され、粒子間の相互作用は部分経路挿入によって行われる。本論文では、挿入操作 PSO 戦略について説明し、数値計算実験からパラメータと得られる解の良さと必要な時間の関係について調査し、パラメータ調整の指針を示す。また、各ベンチマーク問題に対して提案手法と GA などの代表的なメタヒューリスティクスを適用し、提案手法がこれらの手法より短時間で良い解を求められることを示す。

キーワード：粒子群最適化、巡回セールスマン問題

1. はじめに

巡回セールスマン問題 (Traveling Salesman Problem, 以下 TSP) は古くから研究されている組合せ最適化問題の一つであり、訪問すべき都市群と都市間の距離が与えられたとき、すべての都市を一度ずつ訪問して出発した都市に戻る経路 (巡回路) の中で総経路長が最も小さい巡回路を見つける問題である。TSP の定義は単純であるが、複雑性クラスは NP 困難に属しており [1]、実時間内に厳密解を得ることが一般的に困難である。TSP は運搬経路計画、遺伝子地図、X 線結晶構造解析、VLSI 設計など広い分野で応用されており [2]、総経路長の小さい解を短時間で求める手法が古くから盛んに研究されている [3]。

一方、近年注目されている実数値最適化手法の一つに粒子群最適化 (Particle Swarm Optimization, 以下 PSO) [4] [5] がある。PSO では、解空間を動き回る複数の探索単位 (粒子) 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。このとき、各粒子には *pbest*, *gbest* への引力が働き、これまでに発見された良い解の近傍を探索する傾向がある。PSO は多くの多峰性の大域的最適化問題に対して、最適解もしくは準最適解を実用的な計算時間内に求めることが可能なことが示されている [6]。また、そのアルゴリズムが非常に単純であることも特徴である。しかし、PSO は組合せ最適化問題である TSP にそのまま適用することは難しい。これまで、いくつかの TSP 向けに改良された PSO [7] [8] が提案されているが、総経路長が小さな解を得るためにはとても長い時間が必要なことや、アルゴリズムの計算空間量が大きいことなどから、非実用的である。

本論文では、PSO に基づいて実用的な時間や空間で TSP を解く手法を提案することを目的とする。具体的には、解空間上に複数存在する粒子が互いの最良解や全体の最良解から影響を受けて解空間上を移動することで解探索を進める近似解法を PSO 戦略と定義し、最良解の部分経路の挿入に基づいて TSP を解く挿入操作 PSO 戦略を提案する。提案手法では、部分経路の挿入によって最良解に近づく操作を実現しており、従来手法と比較して、最良解の情報をより有効に利用している点、および、解の更新にかかる操作の計算時間の点で優位性がある。また、提案手法といくつかの代表的なメタヒューリスティクスを比較しつつ、計算時間や各パラメータの性質について議論する。実

[†] Particle Swarm Optimization with Tour Representation for Solving Traveling Salesman Problems

Masaya HONJO, Hiroyuki IIZUKA, Masahito YAMAMOTO and Masashi FURUKAWA

^{*1} 北海道大学大学院情報科学研究科

Graduate School of Information Science and Technology, Hokkaido University

^{*2} 北海道情報大学

Hokkaido Information University

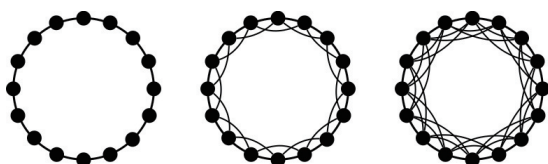


図1 粒子間の通信を制限したPSOで用いられる近傍粒子グラフ。黒点は粒子，黒点を結ぶ線は近傍関係を示す。左から $k=2$, $k=4$, $k=6$

験結果から，提案手法の有効性を示すとともに今後の並列化の見通しについても検討する。

2. 粒子群最適化 (PSO)

PSO [4] は，1995 年に Kennedy と Eberhart によって提案された実数値最適化手法の一つであり，鳥や魚などの群 (swarm) の採餌行動などに見られる群行動を探索手法に応用したものである。複数の粒子 (particle) が解空間内に散りばめられ，それらが相互作用しながら動き回ることによって探索が進む。具体的には，ある粒子 i は n 次元の実数値ベクトルで表現される位置 $\mathbf{x}_i$ と速度 $\mathbf{v}_i$ の情報を保持しており，位置 $\mathbf{x}_i$ が問題の解を表現する。ステップ t における粒子 i の位置 $\mathbf{x}_i(t)$ と速度 $\mathbf{v}_i(t)$ は，(1)，(2) 式に従い更新される。

$$\mathbf{v}_i(t) = w\mathbf{v}_i(t-1) + c_1r_1(\mathbf{p}_i - \mathbf{x}_i(t)) + c_2r_2(\mathbf{g} - \mathbf{x}_i(t)) \quad (1)$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t) \quad (2)$$

ここで， w ， c_1 ， c_2 は定数， r_1 ， r_2 は i に対して独立な $[0, 1]$ の乱数， $\mathbf{p}_i$ はその粒子 i がこれまでに発見した最良な位置 ($\mathbf{pbest}$)， $\mathbf{g}$ は群全体としてこれまでに発見した最良な位置 ($\mathbf{gbest}$) である。 $\mathbf{p}_i$ と $\mathbf{g}$ は粒子 i の更新時に適宜更新する。これらの操作を一定回数繰り返す，より良い $\mathbf{g}$ を探索する。以後，単に $\mathbf{x}_i$ ， $\mathbf{v}_i$ と表記したときは， $\mathbf{x}_i(t)$ ， $\mathbf{v}_i(t)$ を意味する。PSO は，遺伝的アルゴリズム (Genetic Algorithm, 以下 GA) [9] と比較して高速に良い解を発見でき，次元数の増加や多峰性が強い問題に対しても GA より有効であることがわかっている [10]。

また，粒子間の通信を制限した PSO [11] も提案されている。この PSO では，各粒子は近傍と呼ばれるその粒子と通信可能な複数の粒子が設定されている。そして，粒子 i の速度 $\mathbf{v}_i$ の更新には， $\mathbf{gbest}$ の代わりに， i と近傍内の粒子が探索した中での最良解 ($\mathbf{lbest}$) を用いる。具体的な速度 $\mathbf{v}_i$ の更新式を (3) 式に示す。

$$\mathbf{v}_i(t) = w\mathbf{v}_i(t-1) + c_1r_1(\mathbf{p}_i - \mathbf{x}_i(t)) + c_2r_2(\mathbf{l}_i - \mathbf{x}_i(t)) \quad (3)$$

ここで， $\mathbf{l}_i$ は粒子 i の $\mathbf{lbest}$ である。各粒子 i の近傍 A_i はすべての粒子を 1 次元格子状に辺で接続したときに d 個隣りまでの粒子の集合となる。粒子間の近傍関係により作られるグラフを図 1 に示す。各頂点の次数 k は $k=2d$ であり，粒子数を m とすると $2 \leq k < m$ が成り立つ。特別な例として，すべての粒子を近傍と設定する場合， $\mathbf{gbest}$ を用いる PSO と同じ動作となる。このとき，すべての粒子は次数 $k=m-1$ となる。これ以降， $\mathbf{gbest}$ を利用した PSO は， $k=m-1$ を用いる $\mathbf{lbest}$ を利用した PSO として扱う。この拡張された PSO では， k が小さな値のときに局所最適解に陥りにくい特徴がある。

PSO を TSP 向けに改良した手法として，Swap Operator を用いる PSO [7] や Fuzzy 行列を用いる PSO [8] が提案されている。

文献 [7] では Swap Operator (SO) と呼ばれる操作単位を導入することによって PSO を TSP 向けに拡張している。SO(i, j) はある巡回路 $\mathbf{s} = (v_1, v_2, \dots, v_n)$ において，都市 v_i と都市 v_j を入れ替える操作であり，巡回路 $\mathbf{x}_i$ は SO の順列 $\mathbf{v}_i$ の要素を先頭から順に適用することにより更新され， $\mathbf{v}_i$ は $\mathbf{x}_i$ を $\mathbf{pbest}$ と $\mathbf{gbest}$ に変更する 2 種類の SO の順列によって更新される。しかし，この手法では更新された $\mathbf{x}_i$ は結果的に $\mathbf{pbest}$ や $\mathbf{gbest}$ と大きく異なる巡回路となる場合が多く，探索途中で得られた良い巡回路の情報を効率よく利用できていない。そのため，現実的な時間で良い解を得ることができない。

文献 [8] では Fuzzy 行列と呼ばれる解表現を定義し，TSP を PSO で探索可能にしている。具体的には，Fuzzy 行列 W の各要素 w_{ij} は，TSP に含まれる都市の集合 N を $N = \{1, 2, \dots, n\}$ とすると， i 番目に都市 j を訪問する優先度であり，次に示すアルゴリズムに従って都市の訪問順リスト T に一意に変換される。

- (1) T を空のリスト， $i=1$ として初期化する。
- (2) T に含まれていない都市の集合 $N_s = N - T$ を求める。
- (3) 集合 $\{w_{ij} | j \in N_s\}$ の中で最も値の大きな要素 w_{ik} を求め， k を T の末尾に追加する。
- (4) $i \leftarrow i+1$ とし， $i \leq n$ ならば (2) に戻る。

完成した訪問順リスト T はリストの末尾の都市から先頭の都市へ遷移するとみなせば巡回路となる。しかし，Fuzzy 行列を用いる PSO は探索に $O(n^2)$ の

メモリが必要になり、10万都市規模のTSPになるとメモリが確保できずに実行不可能に陥る問題点がある。

3. 提案手法：挿入操作 PSO 戦略

3.1 挿入操作 PSO 戦略のアプローチ

本論文の提案手法である、挿入操作 PSO 戦略 (Insertion-based PSO strategy, 以下 IPSO) のアプローチを説明する。IPSO は PSO を基にしており、各粒子は $pbest$ と $lbest$ の影響を受けて探索を進める。各粒子の近傍粒子は文献 [11] と同様にリング型の近傍を用いる。粒子の近傍構造をリング型にしたときの特徴として、粒子の近傍関係をネットワークとすると、そのネットワークの平均頂点間距離 L を近傍数 k によって大きく変更可能である点が挙げられる。粒子数を m とすると、 $k = m - 1$ のときに $L = 1$ となり、 $k = 2$ のときに少なくとも $L = (m + 1) / 4$ となる。平均頂点間距離 L が大きいということは、ある粒子の解がほかの粒子に伝搬するのに時間がかかることを意味している。これにより、 L が小さいときと比べて各粒子の解のバリエーションが多くなり、局所最適解に陥りにくくなることが期待できる。粒子の更新方法は PSO の更新方法を抽象化し、それに基づいた TSP 向けの更新方法を提案する。PSO では、粒子の位置 x_i 、 $pbest$ p_i 、 $lbest$ l_i はそれぞれ実数値ベクトルであり、 x_i から p_i と l_i へ向かうベクトルが速度 v_i に反映され、位置 x_i が更新される。これは、 x_i を p_i と l_i に近づける操作といえる。TSP で巡回路 a を巡回路 b に近づけたい場合、最も簡単な方法は巡回路 a の経路の一部を巡回路 b の部分経路で置き換えることである。IPSO では、 x_i 、 p_i 、 l_i を巡回路として保持し、 x_i に p_i と l_i の部分経路を挿入することにより x_i を更新する。

3.2 部分経路の挿入

部分経路の挿入方法について、図 2 の例を用いて説明する。ここでは、図 2 のように都市 1, 2, 3, 4, 5, 6, 7, 8, 9 から成る 9 都市の対称 TSP を解いており、図の左上の解 x_i に対して、同じ図中の p_i と l_i の部分経路を挿入することを考える。

まず、提案手法で用いる解表現について説明する。解 $x_i = (1, 2, 3, 4, 5, 6, 7, 8, 9)$ は 1, 2, 3, 4, 5, 6, 7, 8, 9, 1 の順に都市を巡る実行可能解である。TSP では始点や終点となる都市の区別がないので、 $(2, 3, 4, 5, 6, 7, 8, 9, 1)$ としても同じ巡回路を表す。さらに、この例では対称 TSP と呼ばれる、都市 a から都市 b までの距離と都市 b から都市 a までの距離が等しい TSP について考えているので、 x_i の逆順である $(9, 8, 7, 6, 5, 4,$

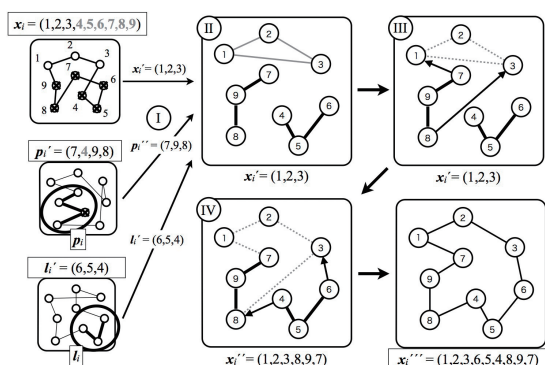


図 2 部分経路挿入法の例

3, 2, 1) も同じ巡回路である。また、 $e(a, b)$ を都市 a から都市 b までの他の都市を経由しない最短道 (辺) とすると、 $x_i^e = \{e(1, 2), e(2, 3), e(3, 4), e(4, 5), e(5, 6), e(6, 7), e(7, 8), e(8, 9), e(9, 1)\}$ のように辺の集合としても表現できる。

次に、部分経路の定義について説明する。部分経路とは巡回路中の連続した辺である。連続な辺とは、 x_i を例に説明すると、 $\{e(1, 2), e(2, 3), e(3, 4)\}$ のような、部分辺集合内の都市が辺を介して互いに接続されている元の巡回路の辺集合 x_i^e の部分集合である。例えば、 $\{e(1, 2), e(4, 5), e(7, 8)\}$ は連続な辺ではないので部分経路ではない。これ以降、部分経路 $\{e(1, 2), e(2, 3), e(3, 4)\}$ は $(1, 2, 3, 4)$ のように訪問する都市の順序で表記する。この場合、巡回路と違い、辺 $e(4, 1)$ は含まれない。

次に、選択された部分集合 p_i' と l_i' の挿入方法について説明する。巡回路 a に部分経路 b' を挿入するとき、 a と b' の両方に含まれる都市が存在する場合、その都市を 2 度訪れることとなり、挿入後の巡回路は実行可能解とはならない。そこで、部分経路 b' に含まれる都市を巡回路 a から除いた閉路 a' に部分経路 b' を挿入することとする。例えば、巡回路 $a = (1, 2, 3, 4, 5)$ 、部分経路 $b' = (2, 4)$ の場合、閉路 $a' = (1, 3, 5)$ となる。辺で考えると、 $e(1, 2), e(2, 3), e(3, 4), e(4, 5)$ が除去され、 $e(1, 3), e(3, 5)$ が追加される。提案手法では、現在の粒子 i の解 x_i に、部分経路 p_i' と部分経路 l_i' を挿入する。2 つの部分経路を挿入するが、 l_i' に含まれる都市が p_i' にもある場合、 p_i' から該当の都市を除去する。図 2 の例では、 $p_i' = (7, 4, 9, 8)$ 、 $l_i' = (6, 5, 4)$ であり都市 4 が共通して含まれているため、 p_i' から都市 4 を除去し、 $p_i'' = (7, 9, 8)$ としている。

最後に、部分経路挿入法のアルゴリズムを以下に示す。

- (1) p_i' から l_i' に含まれる都市を除いた部分経路 p_i'' を生成する。
- (2) x_i から p_i'' または l_i' に含まれる都市を除いた閉路 x_i' を生成する。
- (3) x_i' のすべての辺の中で、その辺の両端の都市の間に p_i'' を挿入したときに最も総経路長が短くなるような辺を選択し、その辺に対して p_i'' を挿入する。この操作により生成された閉路を x_i'' とする。
- (4) x_i'' の辺の中で p_i'' に含まれていない辺に対して、その辺の両端の都市の間に l_i' を挿入したときに最も総経路長が短くなるような辺を選択し、その辺に対して l_i' を挿入する。

手順(3)と(4)で、部分経路を挿入するときに接続の仕方は2通りある。具体的には、選択された辺を $e(v_1, v_2)$ 、挿入する部分経路の両端の都市 v_3, v_4 とすると、 $e(v_1, v_3), e(v_4, v_2)$ と接続する場合と、 $e(v_1, v_4), e(v_3, v_2)$ と接続する場合である。本手法では、これら2通りのうち、総経路長が短くなる方のみを考慮している。

3.3 IPSO のアルゴリズム

部分経路の挿入を導入した PSO 戦略アルゴリズムを以下に示す。各粒子 i は巡回路(解) x_i と、粒子 i の $pbest$ p_i 、粒子 i の $lbest$ l_i をもつ。 n は都市数、 c_1 および 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、 r_1, r_2 はそれぞれ $0 \leq r_1 < 1, 0 \leq r_2 < 1$ の i に対して独立な乱数である。 $[a]$ は実数 a の整数部を返す記号である。 $C(x)$ は巡回路 x の総経路長を返す関数である。また、巡回路 a を巡回路 b に置き換える場合は代入記号を用いて $a = b$ と記述する。

- (1) すべての粒子 i の解 x_i に適当な実行可能解を設定し、 $p_i = x_i$ とする。 l_i は p_i および粒子 i の近傍 A_i の $pbest$ の集合 A_i^p の中で最も総経路長が小さい巡回路とする。数式では(4)式のように定義される。

$$l_i = \operatorname{argmin}_{x \in A_i^p} C(x) \quad (4)$$

- (2) すべての粒子 i について、次を適用する。
 - (a) p_i から長さ $[c_1 r_1 (n+1)]$ の連結成分をランダムに抽出し、部分経路 p_i' とする。
 - (b) l_i から長さ $[c_2 r_2 (n+1)]$ の連結成分をランダムに抽出し、部分経路 l_i' とする。
 - (c) x_i に p_i' と l_i を部分経路挿入法により挿

入し、次の x_i とする。

- (3) すべての粒子 i について、次を適用する。
 - (a) $C(x_i) < C(p_i)$ ならば、 $p_i = x_i$ とする。
 - (b) (4)式に従い、 l_i を更新する。
- (4) すべての粒子の $pbest$ の中で最も総経路長が小さい巡回路を解 s とする。
- (5) 終了条件を満たしていなければ(2)へ戻る。
- (6) 解 s を最終的に得られた解として返す。

アルゴリズム中で、(2)、(3)を1度実行することを1ステップという。IPSOの N ステップ目とは、すべての x_i, p_i, l_i の更新を $N-1$ 回試みた状態を指す。アルゴリズムの終了条件は、 N ステップ経過や制限時間経過などを設定する。

4. 数値計算実験：IPSOのパラメータによる性能変化

提案手法 IPSO には、粒子数 m 、 $pbest$ の重み c_1 、 $lbest$ の重み c_2 、各粒子の近傍数 k 、終了条件の5つのパラメータがある。本実験では、良い解を得るために設定すべき m, c_1, c_2, k の値について調査する。本章では、まず、解の更新が起こらなくなるまで十分に探索する場合に有効なパラメータの考察を行い、続いて、十分に探索が行えないような制限時間が設定されている場合に有効なパラメータの考察を行う。

ここで、IPSOの振る舞いを議論するために2つの特徴量、誤差 e_s と $pbest$ の密度 D_p を定義する。誤差 e_s は解 s の総経路長と厳密解 s_0 の総経路長の誤差率であり、(5)式のように定義される。

$$e_s = \frac{C(s) - C(s_0)}{C(s_0)} \quad (5)$$

e_s は解 s が厳密解 s_0 である場合に0となる。本論文では、複数回試行の平均誤差 $\bar{e}_s$ が小さくなるパラメータを良いパラメータとする。 $pbest$ の密度 D_p は各粒子の $pbest$ p_i の類似度の平均値であり、(6)式のように定義される。

$$D_p = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^m S(p_i, p_j) \quad (6)$$

ここで、 $S(a, b)$ は巡回路 a と巡回路 b の辺の一致率を求める関数であり、(7)式により定義される。

$$S(a, b) = \frac{|E_a \cap E_b|}{n} \quad (7)$$

E_x は巡回路 x の辺集合を表す. $S(a, b)$ は a と b が同じ巡回路の場合に 1 となり, a と b が一つも同じ辺を含まない場合に 0 となる. よって, $pbest$ の密度 D_p は $0 \leq D_p \leq 1$ を満たす. D_p が高い値の場合, $pbest$ と $lbest$ に共通な部分経路が多くなる. この状態では, 解 x_i は p_i や l_i と似た経路に移動しやすくなるため, p_i や l_i の周辺の局所最適解に陥りやすくなる. さらに, $D_p = 1$ の場合では, すべての粒子が同一の巡回路の部分経路の挿入を行うので, すべての粒子が少ないステップでその解に収束する. つまり, $D_p = 1$ の状態が長いステップ続いている場合, これ以上良い解が発見される可能性はほぼないといえる.

4.1 制限時間が設定されていない場合

まず, 解の更新が起こらなくなるまで十分に探索する場合に, 得られる解の誤差を最小にするためのパラメータについて調査する.

4.1.1 実験条件

本実験に用いる IPSO のパラメータ (m, c_1, c_2, k) を表 1 に示す. パラメータの組ごとに 100 回試行し,

表 1 提案手法 IPSO のパラメータ

粒子数 m	{16, 32, 64}
$pbest$ の重み c_1	{0, 0.1, ..., 1.0}
$lbest$ の重み c_2	{0, 0.1, ..., 1.0}
近傍数 k	{2, 4, 6, $m-1$ }

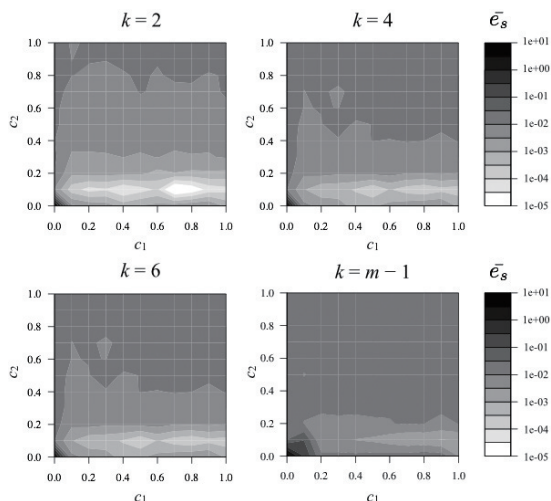


図 3 $m = 32$ の場合の平均誤差 $\bar{e}_s$. 左上: 近傍数 $k = 2$, 右上: $k = 4$, 左下: $k = 6$, 右下: $k = m - 1$

平均誤差 $\bar{e}_s$ と $pbest$ の平均密度 $\bar{D}_p$ をそれぞれ比較する. 実験で使用する TSP は, TSPLIB [12][13] に掲載されている kroA 100 とする. この問題は 100 都市の TSP である. 探索の終了条件は 20000 ステップ経過とする. この終了条件は予備実験を行うことで実験的に求めた.

4.1.2 実験結果

まず, 粒子数 m を 32 に固定し, (c_1, c_2, k) のすべての組合せに対して実験を行う. 得られた平均誤差 $\bar{e}_s$ を図 3 に示す. それぞれ k を 2, 4, 6, $m-1$ と設定したときのグラフであり, 横軸が c_1 , 縦軸が c_2 , 濃淡は平均誤差 $\bar{e}_s$ の高低を表す. 座標点が黒いほど $\bar{e}_s$ が大きく, 白いほど $\bar{e}_s$ が小さい. 濃淡のスケールは結果の見やすさの観点から対数にしている. この条件で最も $\bar{e}_s$ が小さくなるパラメータの組合せは $k = 2$, $c_1 = 0.7$, $c_2 = 0.1$ である. また, すべての k について, $0.4 \leq c_1 \leq 1.0$, $c_2 = 0.1$ の範囲内で $\bar{e}_s$ が小さくなることがわかる.

次に, 上記の実験の中から, k が $\{2, m-1\}$ であり, (c_1, c_2) が $\{(0.7, 0.1), (0.2, 0.1), (0.7, 0.8)\}$ である, 計 6 種のパラメータの組の結果に着目し, それらの平均誤差 $\bar{e}_s$ と $pbest$ の平均密度 $\bar{D}_p$ の推移を図 4 に示す. 横軸はステップ数, 縦軸はそれぞれ平均誤差 $\bar{e}_s$ と $pbest$ の平均密度 $\bar{D}_p$ を表す. $\bar{e}_s$ と $\bar{D}_p$ は 1000 ステップごとに記録している. s はそのステップが終了したのちに更新されたものを使用する. $\bar{e}_s$ のグラフの縦軸は結果の見やすさの観点から対数軸表示している. まず, 平均誤差 $\bar{e}_s$ に着目する. $k = 2$ の場合, $c_1 = 0.7$, $c_2 = 0.1$ のときに最も $\bar{e}_s$ が小さくなり, その次に $c_1 = 0.2$, $c_2 = 0.1$ のときの $\bar{e}_s$ が小さく, $c_1 = 0.7$, $c_2 = 0.8$ のときに最も $\bar{e}_s$ が大きい. 解が収束するまでのステップ数は $\bar{e}_s$ が大きいパラメータの組ほど小さい. $k = m-1$ の場合についても上記と同様の結果である. また, c_1 と c_2 を固定した場合, $k = m-1$ より $k = 2$ の $\bar{e}_s$ が小さくなることがわかる. 次に, $pbest$ の平均密度 $\bar{D}_p$ に着目する. 20000 ステップにおいてはすべてのパラメータの組でほぼ 1 に収束するが, はじめの数千ステップにおいて差がみられる. $c_1 = 0.7$, $c_2 = 0.1$ の場合に最も収束が遅く, その次に $c_1 = 0.2$, $c_2 = 0.1$ の場合の収束が遅く, $c_1 = 0.7$, $c_2 = 0.8$ の場合に最も収束が早い. この傾向は $k = 2$ の場合にも $k = m-1$ の場合にも成り立つ.

次に, $(c_1, c_2) = (0.7, 0.1)$ と設定し, k をそれぞれ 2, 4, 6, $m-1$ に設定したときの平均誤差 $\bar{e}_s$ と $pbest$ の平均密度 $\bar{D}_p$ の推移を図 5 に示す. 図 5 より, k が小さいほど, 最終的に $\bar{e}_s$ が小さくなることと, $\bar{D}_p$ が大きくなりやすいことがわかる. ただし, 1000 ステップ

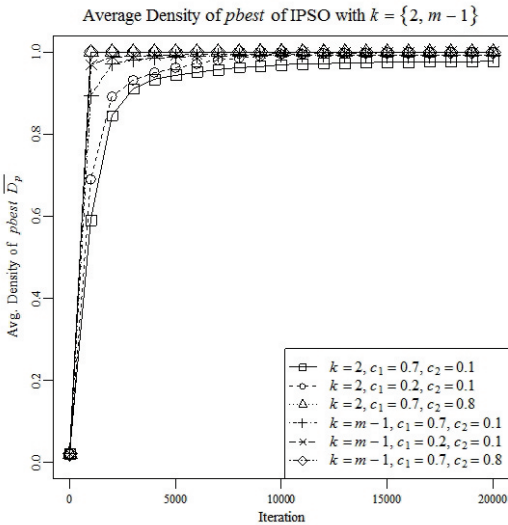
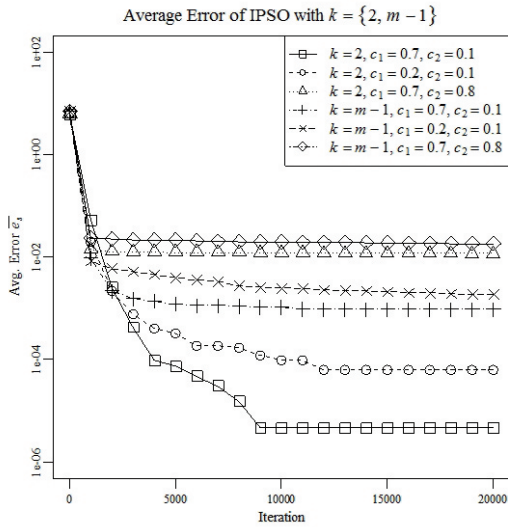


図4 $m=32$, $k=\{2, m-1\}$ のときの平均誤差 $\bar{e}_s$ の推移 (上図) と $pbest$ の平均密度 $\bar{D}_p$ の推移 (下図)

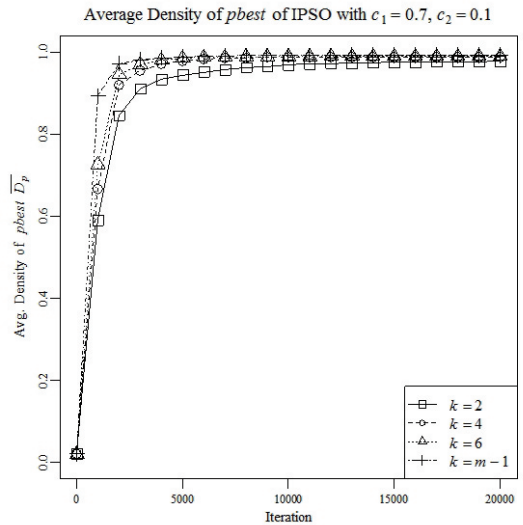
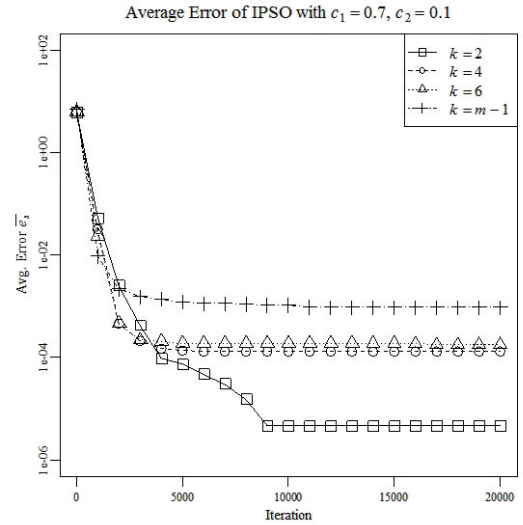


図5 $m=32$, $c_1=0.7$, $c_2=0.1$ のときの平均誤差 $\bar{e}_s$ の推移 (上図) と $pbest$ の平均密度 $\bar{D}_p$ の推移 (下図)

ブ時点では k が大きいほど誤差が小さくなっている。

最後に、 $(c_1, c_2) = (0.7, 0.1)$ とし、 $k \in \{2, m-1\}$ と m の範囲のすべての組合せに対して実験を行う。この実験で得られた平均誤差 $\bar{e}_s$ と $pbest$ の平均密度 $\bar{D}_p$ の推移を図6に示す。 $k=2$, $m=64$ について6000ステップ以降のデータが表示されていないが、これは6000ステップ以降で $\bar{e}_s=0$ であることを表している。図6から、 $k=2$, $k=m-1$ のいずれの場合でも、粒子数 m が大きいほど $\bar{e}_s$ が小さくなることがわかる、 $\bar{D}_p$ は、これまでの実験結果と比較して、 m による変化は微小である。

4.1.3 実験考察

まず、図3, 4の結果から、 $0.4 \leq c_1 \leq 1.0$, $c_2=0.1$ と設定すると誤差 e_s の小さい解を得ることができ、特に $c_1=0.7$, $c_2=0.1$ のときに最も平均誤差 $\bar{e}_s$ が小さくなることが示された。 $c_1=0.7$, $c_2=0.1$ の場合、挿入操作で使用する部分経路 p'_i , l'_i の最大長がそれぞれ $0.7n$, $0.1n$ となる。つまり、 x_i には $pbest$ の比較的長い部分経路と $lbest$ の短い部分経路が挿入される。これより、各粒子は自身の $pbest$ の周辺を重点的に探索し、それぞれの粒子が近づきすぎないようにすれば、最終的に得られる解の誤差 e_s を小さくできると考えられる。これは探索範囲の限定と局所最適解か

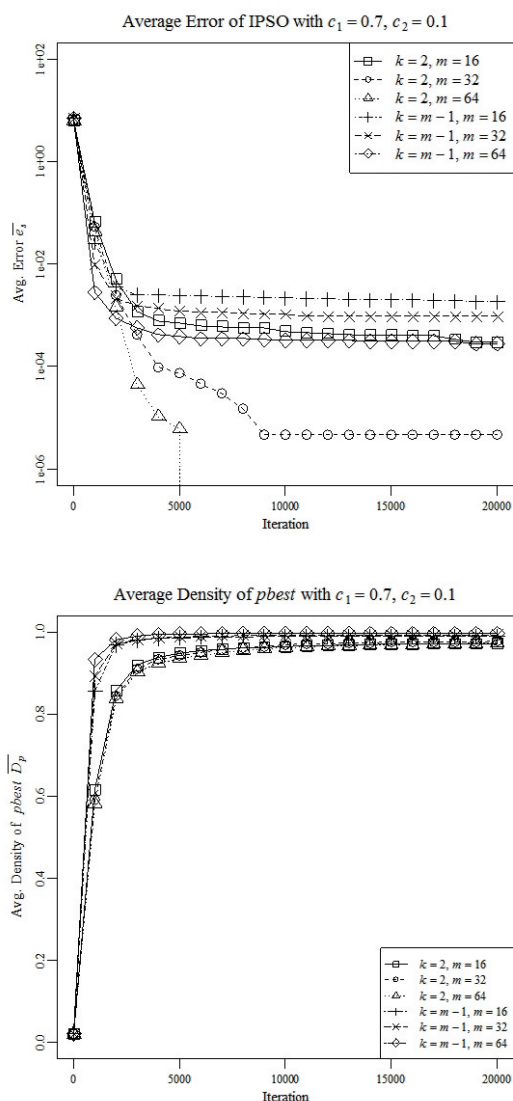


図6 $c_1=0.7$, $c_2=0.1$ のときの平均誤差 $\bar{e}_s$ の推移 (上図) と $pbest$ の平均密度 $\bar{D}_p$ の推移 (下図)

らの脱出のしやすさのバランスをうまくとっている。

次に、図4から、平均誤差 $\bar{e}_s$ が減少するにつれて $pbest$ の平均密度 $\bar{D}_p$ が上昇することがわかる。これは $lbest$ の経路情報の拡散が主な要因である。探索の初期では部分経路の局所最適挿入により移動した x_i は高い頻度で p_i を更新するが、やがて、それだけではなかなか p_i は更新されなくなる。この状態では、 x_i に $lbest$ の部分経路 l_i' を挿入したときに総経路長が減少する可能性が最も高く、 p_i は徐々に l_i に近づいていく。このようにして、最も良い $lbest$ が近傍関係にある粒子を通じてすべての粒子に拡散していく。また、 $pbest$ の密度 D_p の定義でも触れたが、 D_p が高く

なると解は更新されにくくなる。解の更新がしばらく起こらないと、 $lbest$ が拡散して D_p が増加し、さらに解の更新が難しくなるというサイクルが生じている。

次に、図5から、 $k=2$ のときに最も $pbest$ の密度 D_p の増加が緩やかで、最終的に平均誤差 e_s が小さくなることが示された。この理由は、 $k=2$ のときに粒子の近傍関係ネットワークの平均頂点間距離が最も長くなるためである。複雑ネットワークにおいて、距離とは頂点間の最短経路の辺数であり、平均頂点間距離とはすべての頂点对の距離の平均である。この場合、 $lbest$ の拡散にかかる時間が最も長く、 $pbest$ の多様性が比較的長く保たれ、局所最適解に陥りにくい。一方、探索初期においては $k=m-1$ の場合に e_s が最も小さい。 $k=m-1$ のときには $gbest$ を用いる PSO となり、すべての粒子が $gbest$ の周辺を集中的に探索する。このため、少ないステップで e_s が減少しやすいが、 D_p も増加しやすく、局所最適解に陥りやすい。

4.2 制限時間が設定されている場合

次に、時間制限が設定されていて十分に探索が行えない場合に有効なパラメータの考察を行う。

4.1の実験において、十分に時間をかける場合、最適なパラメータセット P_B は $P_B(k, c_1, c_2) = (2, 0.7, 0.1)$ であることを示した。しかし、探索の初期においては P_B 以外のパラメータを用いた方が小さな誤差となる場合がある。そのとき、同時に $pbest$ の平均密度も高くなる。これより、短時間で誤差の小さな解を得るためには、粒子全体の探索範囲を狭めて集中的に探索したほうが良いという仮説が導ける。具体的に探索範囲を狭めるパラメータ調整に以下の方法があげられる。

- 近傍数 k を大きくし、同じ $lbest$ を利用する粒子を増やす。
- $lbest$ の重み c_2 を大きくし、解 x_i を $lbest$ に近づける。
- $pbest$ の重み c_1 を大きくし、解 x_i を $pbest$ に近づける。
- 粒子数 m を小さくすることにより、 $pbest$ や $lbest$ の種類が減り、生成される解の多様性が小さくなる。また、単位時間での探索なので、 m を小さくすると、各粒子の更新回数が増える。

ただし、 c_1 と c_2 の両方を大きくしてしまうと、解は $pbest$ と $lbest$ が入り混じった巡回路になる場合が多く、ある特定の解に落ち着きにくい。つまり、探索範囲を狭めるのならば c_1 と c_2 のどちらか一方だけを大き

表2 本実験で使用するパラメータ

パラメータセット名	m	k	c_1	c_2
P1	32	2	0.7	0.1
P2	32	$m-1$	0.7	0.1
P3	32	2	0.1	0.8
P4	4	2	0.7	0.1
P5	4	$m-1$	0.1	0.8

くすべきである。上記に基づくパラメータ調整の有効性を確かめるために数値計算実験を行い検証する。

4.2.1 実験条件

上記を基に比較するパラメータセットを表2に示す。P1は粒子数 m を32として、 k 、 c_1 、 c_2 は十分に時間をかける場合に最適なパラメータである。P2、P3、P4はP1からそれぞれ、 k 、 (c_1, c_2) 、 m を調整したパラメータセット、P5はP1からすべてのパラメータに調整を加えたパラメータセットである。

実験には575都市問題であるrat575を使用する。このTSPに対して、パラメータをP1～P5に設定した5種類のIPSOを制限時間60秒で200試行、制限時間3600秒で30試行ずつ行い、平均誤差 $\bar{e}_s$ と $pbest$ の平均密度 $\bar{D}_p$ を比較する。 $\bar{e}_s$ と $\bar{D}_p$ は、制限時間が60秒の実験では3秒ごとに、制限時間が3600秒の実験では180秒ごとに記録する。IPSOの実装はScala 2.10.4で行い、実験機のCPUはIntel Xeon E 5520 (2.26 GHz)、OSはFedora 20、Javaのバージョンは1.7.0を使用する。

4.2.2 実験結果

制限時間を60秒に設定したときの平均誤差 $\bar{e}_s$ と $pbest$ の平均密度 $\bar{D}_p$ を図7に、制限時間を3600秒に設定したときの平均誤差 $\bar{e}_s$ と $pbest$ の平均密度 $\bar{D}_p$ を図8に示す。これらの結果より、P1は900秒以降については最も $\bar{e}_s$ が小さく、P1の一部のパラメータに変更を加えたP2～P4の $\bar{e}_s$ は900秒以前において一定の期間P1より小さくなることが示された。つまり、時間制限が設定されている場合、P1以外のパラメータが最適である可能性がある。また、P2～P4の結果を比較すると、最も序盤に $\bar{e}_s$ が小さくなるのはP4で、その次にP3であった。よって、できるだけ短時間で $\bar{e}_s$ を小さくしたい場合、 m を小さくすることが最も効果的で、次に c_1 を小さくしつつ c_2 を大きくすることが効果的である。 k を増やすことは m や (c_1, c_2) を調整するより効果が薄い。また、P5は60秒まではP1～P4より $\bar{e}_s$ が小さくなったので、より短時間で $\bar{e}_s$ を小さくしたい場合はこれらのパラメータを同時に調整すると効果的であることが示された。

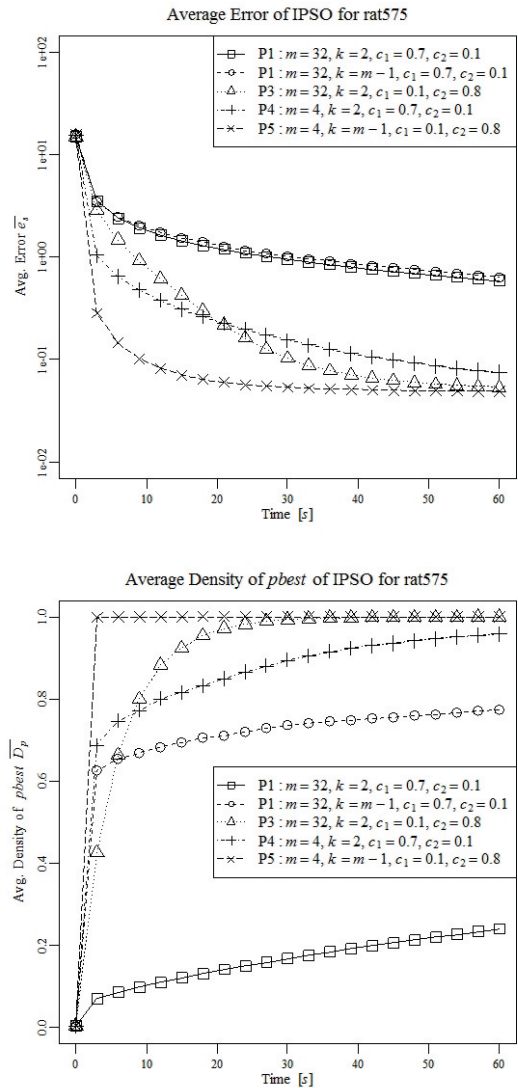


図7 制限時間が60秒の平均誤差 $\bar{e}_s$ の推移（上図）と $pbest$ の平均密度 $\bar{D}_p$ の推移（下図）

また、初期に $\bar{e}_s$ が小さくなるパラメータセットほど、短時間で $\bar{D}_p$ がほぼ1となった。つまり、この実験から上記で示した仮説が成立していることが示された。

5. 数値計算実験：IPSOと既存手法の性能比較

提案手法IPSOの有効性を示すために、いくつかの既存手法との性能比較を行う。本実験では、焼きなまし法(Simulated Annealing, 以下SA) [14] [15]とGA [9]を比較手法として用いる。それぞれの方法で解の生成方法や評価方法がかなり異なるため、解の評価回数などで比較することは適切ではないと考え、実行時

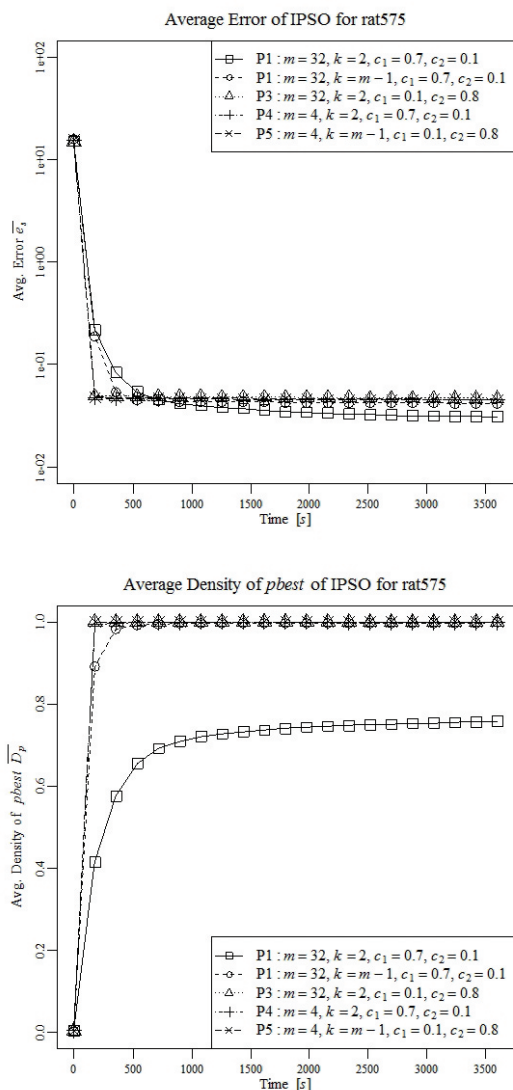


図8 制限時間が3600秒の平均誤差 $\bar{\epsilon}_s$ の推移 (上図) と $pbest$ の平均密度 $\bar{D}_p$ の推移 (下図)

間に基づいた比較を行うこととする。ただし、それぞれの方法でデータ構造や実装上の工夫をすることによって実行時間がかからないよう工夫し、各パラメータも十分にチューニングすることとした。実験では、数種類のベンチマーク問題に対して制限時間 T_c [秒] 内で得られた解の誤差を比較する。

5.1 実験条件

以下に本実験で用いる SA と GA の設定について説明する。

5.1.1 SA

初期解 s_0 はランダムな実行可能解とする。近傍解は現在の解 s からランダムに2辺を選択してそれら

を繋ぎ変えた巡回路とする。近傍解 s' が改悪解であった場合の受理率 p_w は(8)式により決定する。

$$p_w = \exp\left(-\frac{C(s') - C(s)}{T}\right) \quad (8)$$

C は総経路長を求める関数、 T は温度パラメータとする。 T は探索が進むにつれて減少させる。ステップ t における温度 $T(t)$ は(9)式に従う。

$$T(t) = T_0 \alpha^{\left[\frac{t}{R}\right]} \quad (9)$$

T_0 は初期温度、 α は減少率、 R は各温度での反復回数、 $[a]$ は実数 a の整数部を返す記号である。パラメータ T_0 , α , R は問題ごとに予備実験を行い決定する。SA のパラメータ調整については、初期温度 T_0 については探索開始時の受理率が0.5から0.8程度になる値に、減少率は0.9から0.9999などの1に非常に近い値に設定するのが一般的である [16]。初期温度 T_0 , 減少率 α についてはこれらの範囲内で調整し、各温度での反復回数 R は制限時間 T_c を考慮しつつ決定する。

5.1.2 GA

m 個の個体に対し、選択、交叉、突然変異を終了条件が満たされるまで繰り返し適用する。選択にはトーナメント選択を使用する。トーナメント選択は、個体群中の2個体の総経路長を比較し、優秀な個体のみを次世代に残す手法である。選択後の個体数を m にするために、各個体は2度ずつ自身以外の別々の個体と比較される。交叉には提案方法で使用した挿入法を使用する。個体Aと個体Bが交叉のペアとして選ばれた場合、個体Aに個体Bの部分経路を、個体Bに個体Aの部分経路を挿入し、次世代の個体 A' , B' とする。部分経路長 l_p は $2 \leq l_p \leq n$ の範囲でその都度ランダムに決定する。一世代ごとにランダムに選択された m_c 個、 $m_c/2$ 組の個体に交叉を適用する。突然変異は、選択された個体の解 $s = (v_0, v_1, \dots, v_{n-1})$ からランダムに都市をいくつか選択し、それを別の都市間に挿入する操作とする。移動する都市数は1つか2つとし、その都度ランダムに決定する。一世代ごとにランダムに選択された m_m 個の個体に適用する。これ以外に、エリート保存を適用する。エリート保存とは、個体群内で総経路長が小さいいくつかの個体を次世代に残す戦略である。本実験に用いるGAでは、個体群内で総経路長が最も小さい個体を交叉および突然変異の対象から除外する。また、トーナメント選択を

採用しているため、この個体は次世代にも確実に残される。パラメータ m , m_c , m_m は問題ごとに予備実験を行い決定する。

比較には、TSPLIBにある、eil51, kroA100, kroA200, tsp225, pr299, pr439, rat575, d657, u724, pr1002の10個のベンチマーク問題を使用する。問題の都市数は51から1002である。各手法はScala 2.10.4を用いて実装する。また、IPSO, SA, GAは与えるパラメータによって探索性能が大きく変わるため、実装したIPSO, SA, GAを用いて事前実験を行い、各ベンチマーク問題に対して制限時間60秒で得られる解の誤差が最小になるパラメータセットを求め、それを用いる。使用した実験機のCPUはIntel Xeon E 5520 (2.26 GHz)であり、OSはFedora 20, Javaのバージョンは1.7.0を使用する。制限時間 T_e は60秒とする。各手法を各問題にそれぞれ200回適用し、得られた誤差 e_s の最小値, 平均値, 最大値を比較する。

5.2 実験結果

本実験で得られた結果を表3に示す。誤差 e_s の最小値, 平均値, 最大値はそれぞれ百分率で小数点第3位まで表示する。 e_s の標本標準偏差は指数表記で有効桁数3桁で示す。表3より、すべてのベンチマーク問題において、最小値, 平均値, 最大値のいずれもIPSOが最も小さいことがわかる。IPSOとSAについて、それぞれのベンチマーク問題に対して2群の平均値の差の検定をWelchのt検定により行った結果、すべての問題で1%水準で有意差が認められる。また、IPSOとGAについても同様に検定を行った結果、すべての問題で1%水準で有意差が認められる。これらのことから、すべての手法に同一の制限時間を設定する場合、IPSOはSAやGAより誤差 e_s が小さい

解を得られる可能性が高いといえる。

5.3 実験考察

本実験では各ベンチマーク問題に対してIPSOのパラメータ調整を行った。その結果、問題ごとに最適なパラメータは異なり、その傾向は4.2において示したものに一致した。まず、都市数が小さいeil51, kroA100, kroA200に対しては $m=64$, $k=2$, $0.4 \leq c_1 \leq 0.8$, $c_2=0.1$ が最適であった。これらの問題は都市数も小さく、60秒で解の更新が起こらなくなるまで十分に探索できたと考えられる。問題ごとに最適な c_1 の値が異なったが、4.1において c_1 は $[0.4, 1.0]$ の範囲であれば小さな誤差になることを示しており、その範囲内に収まっているので適当である。次にtsp225では $m=32$, pr299では $m=16$, pr439では $m=8$, rat575以上では $m=4$ となり、徐々に使用する粒子数 m は小さくなった。4.2において m を小さくすると他のパラメータを調節したときより早い時間においての誤差が小さくなることを示しており、問題の都市数が大きくなることで急激に探索時間が不足してくることがわかる。また、eil51からpr439に対しては $k=2$, $0.4 \leq c_1 \leq 0.8$, $c_2=0.1$ が最適であったが、rat575からpr1002に関しては $c_1=0.1$, $0.9 \leq c_2 \leq 1.0$ となり、u724以上の問題では $k=m-1$ となった。 m を小さくするだけではさらに探索時間が足りなくなったために、 (k, c_1, c_2) に関しても調整する必要があることが示されている。

以上の結果から、制限時間を設定する場合のIPSOのパラメータ調整について次の指針が導ける。まずはじめに、 $0.4 \leq c_1 \leq 0.1$, $c_2=0.1$, $k=2$ として、粒子数 m をできるだけ大きな値に設定する。このパラメータで解の収束曲線を記録し、制限時間内に収束してい

表3 各ベンチマーク問題に対する結果

TSP	IPSO				SA				GA			
	Error (%)				Error (%)				Error (%)			
	Min	Average	Max	S.D.	Min	Average	Max	S.D.	Min	Average	Max	S.D.
eil51	0.000	0.000	0.000	0.00E+00	0.000	0.134	0.469	1.30E-03	0.000	1.687	3.756	8.33E-03
kroA100	0.000	0.000	0.000	0.00E+00	0.000	0.527	1.917	4.11E-03	0.000	1.817	6.390	1.50E-02
kroA200	0.000	0.595	1.372	2.88E-03	0.143	1.647	4.450	7.66E-03	0.678	3.176	6.453	1.26E-02
tsp225	0.689	1.891	3.013	4.60E-03	0.715	2.911	4.750	8.02E-03	1.175	3.717	6.793	1.07E-02
pr299	0.168	1.218	3.399	5.86E-03	0.488	2.206	7.557	1.01E-02	0.726	3.683	7.454	1.33E-02
pr439	1.010	3.704	7.656	1.40E-02	1.243	4.506	8.317	1.51E-02	1.523	5.042	8.575	1.43E-02
rat575	3.411	4.930	6.629	5.95E-03	5.832	7.724	10.320	7.89E-03	4.370	6.252	8.460	7.04E-03
d657	2.905	4.610	7.655	8.42E-03	5.457	7.882	10.744	1.02E-02	4.807	7.190	10.145	1.05E-02
u724	3.436	5.201	6.991	6.91E-03	6.998	9.514	15.614	1.06E-02	7.559	10.251	12.971	9.26E-03
pr1002	6.351	8.917	11.401	8.26E-03	10.254	12.870	16.705	1.33E-02	11.305	29.299	34.692	3.75E-02

なければ, m を減少させる. m を小さくしても収束しないようであれば, 次は $c_1=0.1$, $0.9 \leq c_2 \leq 1.0$ と設定し, 最後に $k=m-1$ を試みる.

6. おわりに

本論文では, 粒子群最適化 (PSO) を巡回セールスマン問題 (TSP) に適用した. このために, 提案手法 IPSO では粒子の位置を巡回路表現で保持し, これらの粒子の位置を部分経路挿入法を用いて, その粒子の最良解と近傍粒子の最良解のそれぞれの部分経路を現在の位置である巡回路に挿入することで更新するようにアルゴリズムを変更した. そして, IPSO のパラメータ m , c_1 , c_2 , k と得られる解の関係を数値計算実験により調査し, 十分に探索に時間をかける場合に最適なパラメータを示した. それは各粒子の位置である巡回路ができるだけ重複しないようにするパラメータであった. 一方, 探索時間が限られている場合は, 最適なパラメータは一意に決まらず, 制限時間に応じて適切に変更する必要があることを示した. また, 代表的なメタヒューリスティクスである, 焼きなまし法 (SA) や遺伝的アルゴリズム (GA) と比較して, 同じ制限時間では, IPSO の方が高い精度の解を発見可能なことを示した. さらに, この実験結果から IPSO のパラメータ調整の指針も合わせて示した.

今後の課題には提案手法の高速化が挙げられる. その実現のためには提案手法 IPSO の並列計算と TSP の解空間の限定がある. もととの PSO も並列計算が容易なアルゴリズムであることが知られており [17], 筆者らは IPSO の各粒子の更新処理を並列化し, *lbtest* の更新時に同期するアルゴリズムを開発し, 並列化によって計算時間を短縮できることを文献 [18] において示している. TSP の解空間の限定には, 経路長が一定以下の辺のみを使用する方法などがあるが, 各粒子に異なる制限を課すことで厳密解に到達不可能な解空間の限定に陥らないような工夫が必要である.

参考文献

- [1] 山本, 久保, 巡回セールスマン問題への招待, 朝倉書店, 1997.
- [2] W.J. Cook, In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation, Princeton University Press, 2012, 松浦俊輔 (訳): 驚きの数学巡回セールスマン問題, 青土社, 2013.
- [3] 柳浦, 茨城, 組合せ最適化問題に対するメタ戦略について, 電子情報通信学会論文誌. D-I, 情報・システム, I-情報処理, J 83-D-I-1, 3-25, 2000.
- [4] J. Kennedy and R.C. Eberhart, Particle swarm optimization, Proceedings of IEEE International Conference on Neural Networks, 1942-1948, 1995.
- [5] R. Poli, J. Kennedy and T. Blackwell, Particle swarm optimization, Swarm Intelligence, 1-1, 33-57, 2007.
- [6] 相吉, 安田, メタヒューリスティクスと応用, 電気学会, 2007.
- [7] K.Wang, L. Huang, C. Zhou and W. Pang, Particle swarm optimization for traveling salesman problem, Proceedings of International Conference on Machine Learning and Cybernetics, 1583-1585, 2003.
- [8] W. Pang, K. Wang, C. Zhou and L. Dong, Fuzzy discrete particle swarm optimization for solving traveling salesman problem, Proceedings of International Conference on Computer and Information Technology, 796-800, 2004.
- [9] J.H. Holland, Adaptation in natural and artificial systems, U Michigan Press, 1975.
- [10] J. Kennedy and W.M. Spears, Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator, Proceedings of IEEE International Conference on Evolutionary Computation, 78-83, 1998.
- [11] 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.
- [12] G. Reinelt, TSPLIB—A Traveling Salesman Problem Library, ORSA Journal on Computing, 3-4, 376-384, 1991.
- [13] TSPLIB, <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>
- [14] S. Kirkpatrick, C.D. Galatt Jr. and M.P. Vecchi, Optimization by Simulated Annealing, Science, 220-4598, 671-680, 1983.
- [15] V. Cerny, Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, Journal of Optimization Theory and Applications, 45, 193-226, 1985.
- [16] 古川, 川上, 渡辺, 木下, 山本, 鈴木, メタヒューリスティクスとナチュラールコンピューティング, コロナ社, 2012.
- [17] 斎藤利通, 粒子群最適化と非線形システム, 電子情報通信学会基礎・境界ソサイエティ Fundamentals Review, 5-2, 155-161, 2011.
- [18] 本庄, 山本, 巡回セールスマン問題向け並列粒子群最適化の提案, 第 14 回公益社団法人計測自動制御学会システムインテグレーション部門講演会 (SI 2013) 講演論文集, 914-919, 2013.

(2015 年 9 月 10 日 受付)
(2016 年 2 月 29 日 採録)

問い合わせ先

〒060-0814 北海道札幌市北区北 14 条西 9 丁目
北海道大学 大学院情報科学研究科 情報理工学専攻
自律系工学研究室
本庄 将也
TEL : 011-706-6445
E-mail : honjyo@complex.ist.hokudai.ac.jp

著者紹介

ほんじょう まさや
本庄 将也 [非会員]



2010年北海道大学工学部情報エレクトロニクス学科卒業。2012年北海道大学大学院情報科学研究科修士課程修了。同年同大学院博士後期課程入学。現在に至る。複雑ネットワーク、オペレーションズ・リサーチの研究に従事。情報処理学会、精密工学会、各会員。

いひづか ひろゆき
飯塚 博幸 [非会員]



2004年東京大学大学院総合文化研究科博士課程修了。博士(学術)。2005年日本学術振興会特別研究員(PD, はこだて未来大学), イギリスサセックス大学客員研究員, 2008年大阪大学大学院情報科学研究科 助教。2013年北海道大学大学院情報科学研究科 准教授。専門は人工生命, 複雑系科学, 人間情報工学。情報処理学会員。

やまもと まさひと
山本 雅人 [正会員]



1996年北海道大学大学院工学研究科システム情報工学専攻博士後期課程修了。同年日本学術振興会特別研究員(PD)。1997年北海道大学大学院工学研究科助手。2000年同大学院工学研究科助教授。同大学院情報科学研究科助教授, を経て2012年同大学院教授。この間, 科学技術振興機構さがけ研究員, デューク大学客員研究員, 博士(工学)。ゲームプログラミング, 複雑ネットワークの研究に従事。情報処理学会, 人工知能学会など, 各会員。

ふるかわ まさし
古川 正志 [非会員]



1973年北海道大学大学院工学研究科修士課程修了。同年旭川工業高等専門学校助手。その後, 助教授, 教授を経て, 2006年北海道大学大学院情報科学研究科教授。2013年より北海道情報大学教授, 現在に至る。1976-1977年コーネル大学NSF研究員。1981-1982年イーストアングリア大学客員教授。人工生命とアニメーション, メタヒューリスティクス, 複雑ネットワークの研究に従事(工学博士)。精密工学会, 日本機械学会(フェロー), 情報処理学会, 日本ロボット学会等の各会員。

Particle Swarm Optimization with Tour Representation for Solving Traveling Salesman Problems

by

Masaya HONJO, Hiroyuki IIZUKA, Masahito YAMAMOTO and Masashi FURUKAWA

Abstract:

In this paper, we propose a new Traveling Salesman Problem (TSP) solver based on Particle Swarm Optimization (PSO) Algorithm. PSO is one of the optimization methods classified into swarm intelligence, and consists of particles. Particles interact and move through solution space to find a better solution. PSO can find a good solution in a short time compared with Genetic Algorithm (GA) in many real-valued optimization problems. Because TSP is a combinatorial optimization problem, we change two points of the original PSO for solving TSP. The first point is that the position of each particle is represented by a tour instead of a vector. The second is that particles move by using Insertion method. Insertion method is an operation to combine a tour and sub-paths of other two tour. We analyze relation between parameters and length of an obtained tour, and indicate a guide to adjust parameters. The performance comparison result shows that the proposed method can find a better solution than Simulated Annealing (SA) and GA.

Keywords: Particle Swarm Optimization, Traveling Salesman Problem

Contact Address: **Masaya HONJO**

*Laboratory of Autonomous Systems Engineering, Division of Computer Science and Information Technology, Graduate School of Information Science and Technology, Hokkaido University
Kita 14, Nishi 9, Kita-ku, Sapporo, Hokkaido, 060-0814, Japan
TEL: 011-706-6445
E-mail: honjyo@complex.ist.hokudai.ac.jp*