

ハミング距離に基づく遷移確率を用いた PSOによるグラフ色塗り問題の解法

青木拓也^{†1} アランニャ・クラウス^{†2} 狩野均^{†2}

本論文では、Particle Swarm Optimization (PSO) を用いた、グラフ色塗り問題の解法を提案する。PSO は本来、連続値変数の最適化問題に対して提案された多点探索手法である。そのため、離散値変数の最適化問題に直接適用することは難しい。PSO の速度を遷移確率で表現した研究があるが、距離の概念を考慮していない。本手法は探索点（解候補）間の距離をハミング距離で表現し、その距離に比例した遷移確率を導入した。ランダムに作成したグラフ色塗り問題を用いて、本手法と GA および従来の遷移確率を用いた PSO との比較実験を行った。その結果、本手法は正解率と探索速度において従来手法よりも優れていることを確認した。

PSO Algorithm with Transition Probability Based on Hamming Distance for Solving Planar Graph Coloring Problem

TAKUYA AOKI^{†1} CLAUS ARANHA^{†2}
HITOSHI KANO^{†2}

In this paper, we propose a PSO algorithm with transition probability based on hamming distance for solving planar graph coloring problems. The first version of PSO was intended to handle only continuous optimization problems. Then, to apply PSO to discrete problems, the standard arithmetic operators of PSO are required to be redefined over discrete space. Chin et al (2010) introduced transition probability into PSO to settle this problem, but this model does not consider the concept of distance. In this work, we propose a new algorithm that uses transition probability based on hamming distance into PSO. The experimental results show that the new algorithm can get higher correction coloring rate and smaller average iterations than a Genetic Algorithm and the conventional PSO.

1. はじめに

Particle Swarm Optimization(PSO)は1995年にKennedyとEberhartによって提案された、鳥の群れの採餌行動をモデルとした多点探索手法である[1]。しかし、初期のPSOは連続値変数の関数最適化問題の解法として提案されたため、離散値変数を扱う問題には直接適用することはできなかった。しかし、離散値変数を扱う重要な問題も多いため、PSOを離散値問題に適応させる研究が多く行われてきた[2]。

代表的な手法として、KennedyとEberhartが提案したBinary PSO (BPSO)がある[3]。これはシグモイド関数(SF)を利用して0,1の二値の離散値に対応できるようにしたものである。以降、そのBPSOを拡張し、ナップサック問題に適応させる手法も多く提案されている[4][5][6]。

またPSOにSmallest Position ValueなどSF以外の方法で離散値を扱う手法も提案されており、オペレーション・スケジューリング問題や巡回セールスマン問題を解く手法も提案されている[7][8]。

BPSOを拡張し、制約充足問題の一つであるグラフ色塗り問題(GCP)に適応する手法も提案されている[9][10]。

陳らはPSOに遷移確率を新たに導入することで、離散値問題に適用する手法を提案した[11]。しかし、この手法はPSOの特徴の一つである、探索点（解候補）間の距離を考慮していないため、モデル化としては不十分であり、効率的な探索を妨げていると考えられる。

本研究は、陳らの手法を改良し、遷移確率を探索点間のハミング距離に基づいたものにし、探索性能を向上させた手法を提案する[12]。様々なPSOのアルゴリズムが提案されているが、他のメタヒューリスティクス手法と比較して、その性能はあまり高くない[13]。一方、PSOと他のヒューリスティクスをハイブリッド化したPSOアルゴリズムはよい性能が得られているが、これらは特定の問題のために設計されており、構造はより複雑で、汎用性が低いことが指摘されている[13]。提案手法は、性能と汎用性の両方に優れたモデル化を狙っている。

以下では、まず研究分野の概要として、通常のPSOの動作原理、グラフ色塗り問題、関連研究について述べる。次に提案手法を説明し、最後にランダムに生成したグラフ色塗り問題を対象として、従来の遷移確率を用いたPSO（従来手法）、進化的アルゴリズム(GA)と比較実験を行い、本手法の有効性を評価した。

^{†1} 筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻
Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba

^{†2} 筑波大学システム情報系情報工学域
Division of Information Engineering, Faculty of Engineering, Information and Systems, University of Tsukuba

2. 研究分野の概要

2.1 通常の PSO の動作原理

通常の連続変数を対象とした PSO は次に示す探索点の速度と位置の更新式で構造化されている。各探索点は群れ全体の最良解の位置情報を共有しているので、ある探索点が良い解に近づくと、他の探索点もその良い解に近づく。

・速度の更新式：

$$v_i^{k+1} = wv_i^k + c_1r_1(pb_{est}_i^k - x_i) + c_2r_2(gb_{est}^k - x_i)$$

・位置の更新式： $x_i^{k+1} = x_i^k + v_i^{k+1}$

i ：探索点の番号 k ：現在の探索回数

v_i^k ： k 回目の探索での i 番目の探索点の速度

x_i^k ： k 回目の探索での i 番目の探索点の位置

$pb_{est}_i^k$ ： k 回目の探索での i 番目の探索点自身の最良解

gb_{est}^k ： k 回目の探索での群れ全体の最良解

w ：慣性力パラメータ

c_1 ： $pb_{est}_i^k$ へ近づく力の大きさを表すパラメータ

c_2 ： gb_{est}^k へ近づく力の大きさを表すパラメータ

r_1, r_2 ： $[0,1]$ の一様乱数

2.2 グラフ色塗り問題

グラフ色塗り問題とは、与えられた無向グラフに対して、辺で繋がれた頂点が同じ色にならないようにすべての頂点に色を塗る問題である。この問題は NP 完全であり、制約充足アルゴリズムの性能を評価するために用いられている。また、この問題は変数間の依存性が強く、局所最適解が生じやすいため、探索が難しいという特徴がある[14]。

本研究では、Minton らの論文[15]に示してある方法で、3色グラフ色塗り問題を作成する。またここで辺の数 m を頂点の数 n で割ったものを制約密度 $d (= m / n)$ として定義する。これは問題の難易度を分類するためしばしば用いられる指標である。これにより、どのような問題に対して手法が有効であるのか明確になる。Hogg らの研究によれば GCP においては $d = 2.0 \sim 2.5$ 付近で難しい問題になるとされている[16]。

解のコード化として、本手法では解候補を次元配列として表現し、探索点とする。使用する色を赤、青、緑の三色とし、赤は 0、青は 1、緑は 2 に対応させ、各頂点はいずれかの値をとるものとする。例えば頂点数 4 のグラフの場合、解候補の一つは図 1 のように表される。

また、各探索点の適応度 $f(x)$ を以下の式で表す。

$$f(x) = 1 - (\text{その探索点の違反数} \div m)$$

なお、探索点の違反数は辺で繋がっている頂点の色が同じである場合一つにつき、違反数 1 とする。

2.3 関連研究

2.3.1 MTPSO

BPSO を GCP に拡張させた先行研究として Hsu らの Modified Turbulent PSO (MTPSO)がある[10]。この研究では Walking one strategy, Assessment strategy, Turbulent strategy

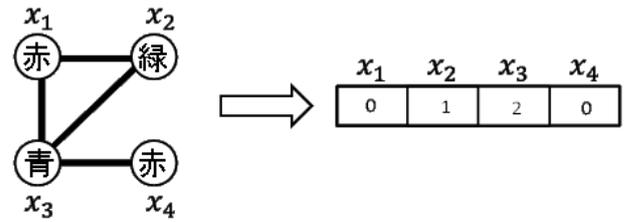


図 1 解のコード化の例

を導入することによって、通常の PSO よりも性能を高めることに成功した。

しかし、そもそもグラフ色塗り問題のような問題では、色などを数字にコード化するため、数値的な差は最適化に関係しない（例えば探索点のある成分を赤にすれば最適解になる場合、青を赤に変える場合、緑を赤に変える場合では数値としては 0 と 1, 0 と 2 の差があるが、本質的にはどちらも同じである）。そのため、グラフ色塗り問題においては精度が低下する原因となっている可能性がある。よって、今回は比較対象としない。

2.3.2 遷移確率を用いた PSO

BPSO を元にシグモイド関数で離散値と連続値の変換をする手法が主流であったが、陳らは別のアプローチとして遷移確率を提案した[11]。これは PSO における速度をそのまま確率にすることで離散値を直接扱えるようにした手法である。

この手法であれば、先に述べた色を数字にコード化することによる影響を受けないという利点がある。

以下に遷移確率 $P(D_l)$ を求める式を具体的に示す。

$$P(D_l) = \frac{wP_{set}(D_l) + c_1r_1P_{pbest}(D_l) + c_2r_2P_{gbest}(D_l)}{w + c_1r_1 + c_2r_2}$$

ただし、 D_1, D_2, D_3 は各色を表す。また、 $P_{set}(D_l)$ は

$$\sum_{l=1}^3 P_{set}(D_l) = 1$$

$$0 \leq P_{set}(D_l) \leq 1$$

を満たすパラメータであり、実際の実験では $P_{set}(D_l) = 1/3$ を採用している。また、

$$P_{pbest}(D_l) = \begin{cases} 1 & (pb_{est}_{ij}^k = D_l) \\ 0 & (otherwise) \end{cases}$$

$$P_{gbest}(D_l) = \begin{cases} 1 & (gb_{est}_j^k = D_l) \\ 0 & (otherwise) \end{cases}$$

と定義する。

すなわち、 $pb_{est}_i^k, gb_{est}^k$ の j 番目の成分が持つ色ほど、その色に遷移する確率が増えるよう設定されている。

しかし、この手法では探索点間の距離を考慮しておらず、

表 1 実験条件

	w 最適化	c_2 最適化	c_1 最適化
個体数	10	10	10
最大反復回数	1.0×10^5	1.0×10^5	1.0×10^5
w	変化	最適値	最適値
c_1	1.0 に固定	1.0 に固定	変化
c_2	1.0 に固定	変化	最適値
結果	図 2	図 3	図 4

PSO の利点を十分活かしかれていないとは言えない。また色ごとに遷移確率を設定しているため、探索点が持つ解の全体的な性質を探索に活かさないという問題がある。また、色数が増えるほど計算量が増えるという問題もある。

3. 提案手法

3.1 基本戦略

遷移確率というアイデアを活かしつつ、探索点間の距離を考慮できるように、遷移確率の導出式と遷移確率による操作を変更する。

まず各探索点間の距離としてハミング距離を導入する。これにより、数字によるコード化の影響を受けない探索が可能となる。さらに探索点の全体的な性質を探索に利用するため、従来手法[11]において色ごとに設定されていた遷移確率を探索点ごとに設定する。これにより計算量は探索点数のみに依存することになる。また探索点ごとに異なる遷移確率を持つため、探索の多様化も狙える。

これに伴い、遷移確率による操作を変更する。基本的なアイデアとしては、 $pbest_i^k$ とのハミング距離が小さいほど $pbest_i^k$ への遷移確率が大きくなる。また、 $gbest^k$ とのハミング距離が小さいほど $gbest^k$ への遷移確率が大きくなる。さらに以前の位置とのハミング距離が小さいほど、ランダムに動く遷移確率が大きくなるように設定する。

3.2 遷移確率

遷移確率 P_i^a を以下のように定義する。なお、 $H(x,y)$ は x,y 間のハミング距離を表すとする。

$$P_i^a = \frac{z_i^a}{\sum_{b=1}^3 z_i^b}$$

ただし、

$$z_i^1 = w(1 - \frac{H(x_i^k, x_i^{k-1})}{n})$$

$$z_i^2 = c_1 r_1 (1 - \frac{H(x_i^k, pbest_i^k)}{n})$$

$$z_i^3 = c_2 r_2 (1 - \frac{H(x_i^k, gbest^k)}{n})$$

とする。

そして、 k 回目の探索での i 番目の探索点の j 番目の成分

x_{ij}^k に対し、次の操作を行う。

$$\begin{cases} \text{確率 } P_i^1 \text{ で } x_{ij}^k \text{ の色をランダムな色置換} \\ \text{確率 } P_i^2 \text{ で } x_{ij}^k \text{ の色を } pbest_{ij}^k \text{ の色に置換} \\ \text{確率 } P_i^3 \text{ で } x_{ij}^k \text{ の色を } gbest_{ij}^k \text{ の色に置換} \end{cases}$$

3.3 提案手法のアルゴリズム

アルゴリズムの疑似コードを以下に示す。

```

探索点をランダムに初期化;
for(反復回数  $k=1$  to 最大反復回数){
    for(探索点  $i=1$  to 個体数){
        探索点の制約違反数と  $pbest_i^k$  を求める;
         $gbest^k$  を求める;
        if(適応度  $f(gbest^k) == 1$ ){break;}
        for(探索点  $i=1$  to 個体数){
            探索点  $i$  の遷移確率  $P_i^1 \sim P_i^3$  を求める;
            for(探索点  $i$  の成分  $j=1$  to  $n$ ){
                遷移確率  $P_i^1 \sim P_i^3$  に従い色を遷移;}
            }
        }
    }
    最良解を出力;

```

4. 評価実験

本章では提案手法の性能を調べるために、提案手法のパラメータを最適化する予備実験を示す。次に提案手法と、従来の遷移確率を用いた PSO (従来手法)、進化的アルゴリズム(GA)で比較実験を示す。ここでは最適解の発見回数(正解数)と最適解の発見までにかかった平均探索回数(平均探索回数)を指標として評価を行った。なお、平均探索回数の算出には、解を発見できなかった場合を除いた。

4.1 パラメータ設定

(1) 実験方法

頂点数 $n=90$ 、制約密度 $d=2.0$ で生成した問題を 100 問解き、その正解数を元に、パラメータ w 、 c_1 、 c_2 の最適化を順次行った。まずパラメータ w を決定するためにパラメータ c_1 、 c_2 を固定して w を変化させ実験した。正解数が最大となった w を最適値とした。

次に最適値の w を使い、以下同様に各パラメータを最適化した。実験の条件を表 1 に示す。なお、パラメータ c_2 から先に設定したのは、後に示すように c_2 の最適値が予想よりも小さかったため、先にパラメータ c_1 を最適化しても局所最適解に収束し、十分な探索性能が得られなかったためである。

(2) 実験結果

パラメータを変えて実験結果をそれぞれ図 2~4 に示す。図 2~4 から以下のことがわかる。

- $w=0.05$ で正解数が最大になる。

- $w = 0.05$ を最大として $w = 0$, $w = 0.1$ に近づくにつれ正解数が低下していく傾向がある.
- $c_2 = 0.03$ で正解数が最大になる.
- $c_2 = 0.2$ に近づくにつれ正解数が低下していく傾向がある.
- $c_1 = 7.0$ で正解数が最大になる.
- $c_1 = 1.0$ から $c_1 = 0$ に近づくにつれ正解数が低下していく傾向がある.

(3) 考察

実験結果から各パラメータの性質を考察する.

パラメータ w は大きすぎても小さすぎても探索性能を低下させる. 大きすぎると単なるランダムサーチになり, 小さすぎると局所最適解に陥ってしまうからだと考えられる.

パラメータ c_2 は大きすぎると探索性能を低下させる. $c_2 = 0$ でも探索性能が低下していないことから, $gbest^k$ が探索に及ぼす良い影響は比較的軽微であり, また大きすぎると悪影響を及ぼすことがわかる. この理由としては, GCP の特性が関係していると考えられる. GCP の良い解は辺で繋がった頂点間の関係で成り立っているが, 提案手法ではその関係を考慮することができない. また, 解の色を循環的に入れ替えても本質的には同じ解であり, 6 つのパターンが考えられるが, 提案手法ではそれを考慮することができない. つまり, $gbest^k$ と探索点のパターンが不一致であれば, 探索に悪影響を与えてしまうということである. よって, 上記の二点を考慮すればさらなる性能の向上が予想できる.

パラメータ c_1 は小さすぎると探索性能を低下させる. これは探索点の付近が探索できず, 付近にあるより良い解の発見が困難になるからだと考えられる.

4.2 探索挙動

提案手法の探索挙動を図 5 に示す. 実験条件は表 1 と同様であり, $n = 90$, $d = 2.5$ のときの試行 100 回の平均値である. 図 5 から探索は 10 万回でほぼ収束していることがわかる.

4.3 実験方法

本手法, 従来手法, GA でグラフの頂点数 n を 90, 120, 150, 制約密度 d を 1.5~10 で変化させ, それぞれ 100 問ずつグラフ色塗り問題を解いた. そして正解数と平均探索回数(個体数×世代数 or 反復回数)を比較した. 表 2 に実験条件を示す.

なお従来手法のパラメータは文献 [11] の値を用いた. また GA のパラメータは, 提案手法と同様の予備実験により決定した. また, GA はトーナメント選択, エリート保存あり, 一様交叉, 線形スケールを採用した.

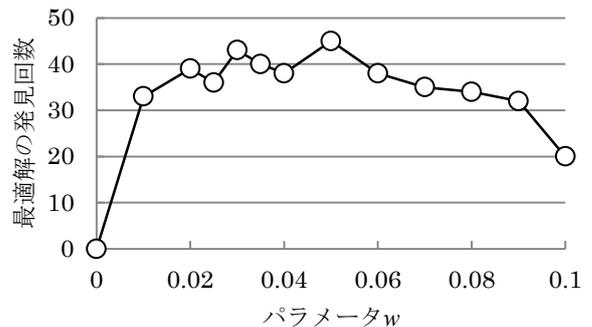


図 2 パラメータ w の設定

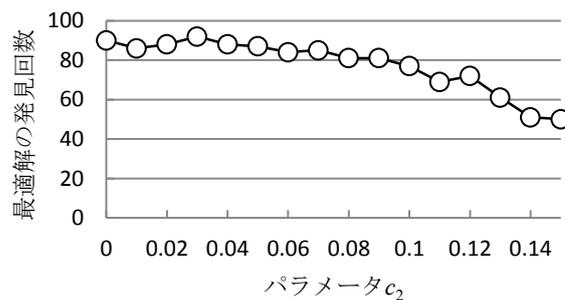


図 3 パラメータ c_2 の設定

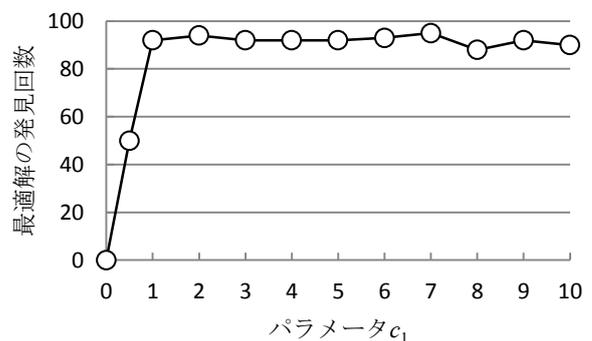


図 4 パラメータ c_1 の設定

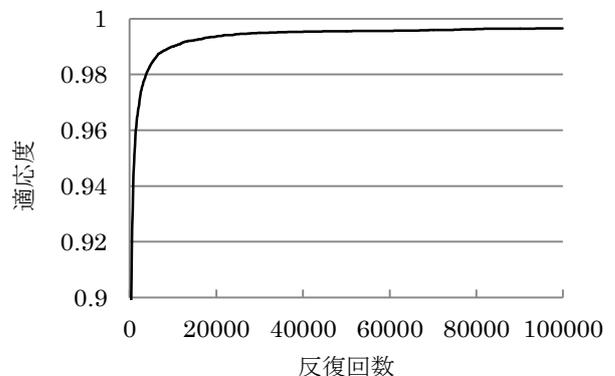


図 5 提案手法の探索挙動($n = 90$, $d = 2.5$)

表 2 実験条件

	本手法	従来手法	GA
個体数	10	10	50
最大世代数 or 反復回数	1.0×10^5	1.0×10^5	2.0×10^4
w	0.05	0.02	N/A
c_1	7.0	2.5	N/A
c_2	0.03	1.0	N/A
突然変異率	N/A	N/A	1.4%
トーナメント	N/A	N/A	2
エリート保存	N/A	N/A	1

4.4 実験結果

(1) 正解数での比較

正解数で比較した実験結果をそれぞれ図 6~8 に示す。図 6~8 から以下のことがわかる。

- $n = 90$ では、 $d = 2.0 \sim 5.0$ の範囲で他手法よりも提案手法の正解数が多い。
- $n = 120$ では、 $d = 1.5 \sim 4.0$ の範囲で他手法よりも提案手法の正解数が多い。
- $n = 150$ では、 $d = 1.5 \sim 6.0$ の範囲で他手法よりも提案手法の正解数が多い。
- $n = 90 \sim 150$ のどの範囲でも、 $d = 7.0$ 以上の範囲では、どの手法もほぼ同程度の高い正解数だった。
- 最も難しい $d = 2.5$ の問題において、提案手法は他手法よりも、 $n = 90$ では約三倍、 $n = 120$ では約二倍、正解数が多かった。 $n = 150$ では他手法と同程度の正解数だった。

(2) 平均探索回数での比較

平均探索回数で比較した実験結果をそれぞれ図 9~11 に示す。図 9~11 から以下のことがわかる。

- $n = 90 \sim 150$ では、 $d = 1.5 \sim 4.0$ の範囲で提案手法のほうが、GA よりも少ない平均探索回数で最適解を発見している。
- $n = 90, 120$ では、 $d = 6.0 \sim 10$ の範囲で、GA のほうが、提案手法よりも少ない平均探索回数で最適解を発見している。
- $n = 90$ では、 $d = 6.0 \sim 10$ の範囲で、従来手法のほうが、提案手法よりも少ない平均探索回数で最適解を発見している。
- $n = 120$ では、 $d = 2.5 \sim 3.0, 5.0 \sim 10$ の範囲で、従来手法のほうが、提案手法よりも少ない平均探索回数で最適解を発見している。
- $n = 150$ では、 $d = 2.0, 3.0, 6.0 \sim 10$ の範囲で、従来手法のほうが、提案手法よりも少ない平均探索回数で最適解を発見している。

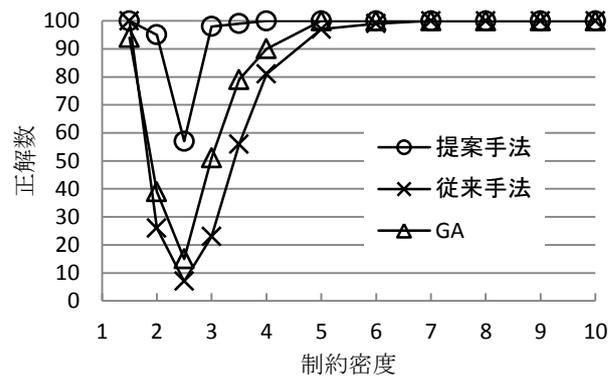


図 6 正解数での比較($n = 90$)

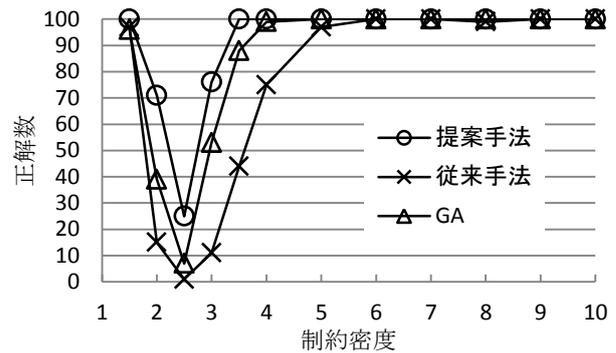


図 7 正解数での比較($n = 120$)

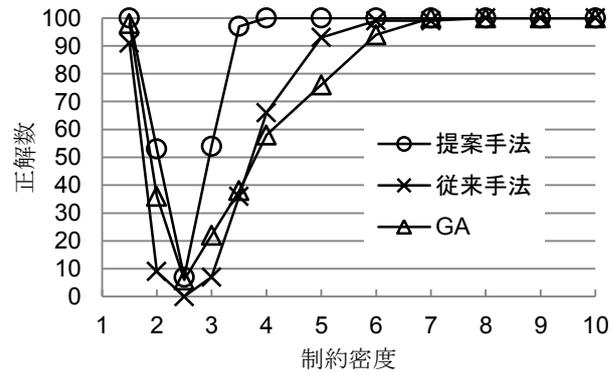


図 8 正解数での比較($n = 150$)

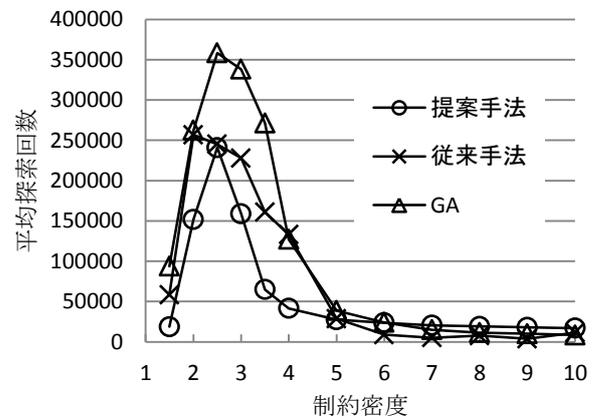


図 9 探索回数での比較($n = 90$)

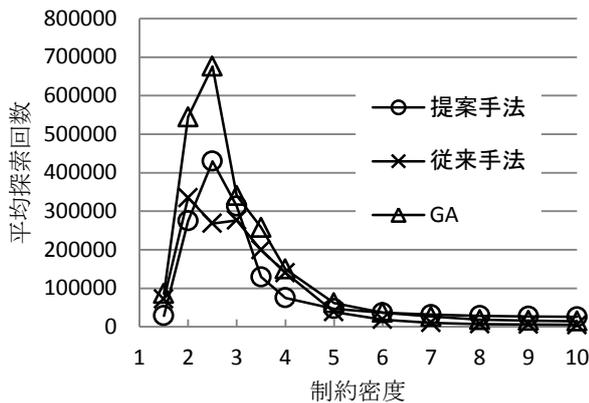


図 10 探索回数での比較($n = 120$)

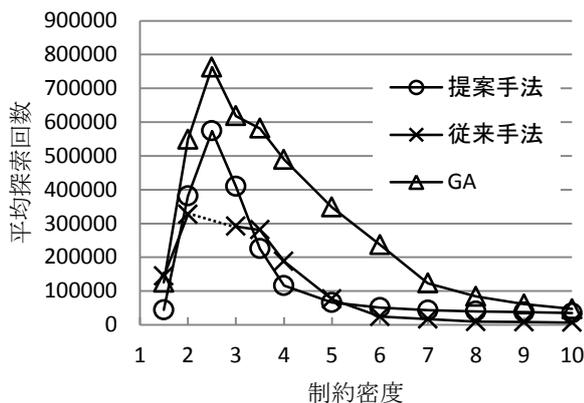


図 11 探索回数での比較($n = 150$) (※従来手法では $d = 2.5$ で最適解を一度も発見できなかったため、その部分のグラフを点線で示している)

4.5 考察

実験により、提案手法は、難しい問題範囲において、従来手法、GA よりも高い正解率を持つこと、 $d = 4.0$ 以下の範囲で GA よりも提案手法のほうが高い探索速度を持つことがわかった。正解率、探索速度が向上した理由として、次のことが考えられる。

- ・ 従来手法に距離の概念をハミング距離として導入することで、従来手法より効率的に $pbest$, $gbest$ を解の探索に活かした。
- ・ 探索点ごとの最良解を保持することで、GA よりも多くの最良解付近を重点的に探索できた。

5. おわりに

PSO にハミング距離に基づいた遷移確率を導入し、グラフ色塗り問題を対象に他手法との比較を行った。実験結果により、 $n = 90 \sim 150$, $d = 2.0 \sim 4.0$ の範囲で、従来手法、GA と比較して提案手法によって正解率が改善することを確認した。特に $n = 90$, $n = 120$ では、最も難しい問題である $d = 2.5$ の問題において大きな改善が見られた。また、 $n = 90 \sim 150$, $d = 1.5 \sim 4.0$ の範囲では、GA よりも提案手法のほうが高い探索速度を持つことを確認した。

今後は、 $gbest$ をよりよく探索に反映できるようにする

など、提案した手法のアルゴリズムを改良し、スケジューリング問題などの組合せ最適化問題へ応用する予定である。

参考文献

- 1) Kennedy, J., Eberhart, R.: Particle swarm optimization, In Proceedings of the IEEE international conference on neural networks, Vol. 4, pp. 1942-1948, (1995)
- 2) Jonas Krause, Jelson Cordeiro I, Rafael Stubs Parpinelli, Heitor Silve'rio Lopes.: A Survey of Swarm Algorithms Applied to Discrete Optimization Problems, Swarm Intelligence and Bio-Inspired Computation, pp.169-191, (2013).
- 3) Kennedy, J., Eberhart R.C.: A discrete binary version of the particle swarm algorithm, In IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Vol. 5, pp. 4104-4108, (1997).
- 4) Deep, K., Bansal, J.C. : A socio-cognitive particle swarm optimization for multidimensional knapsack problem. In: First International Conference on Emerging Trends in Engineering and Technology, pp. 355-360, (2008).
- 5) Hembeker, F., Lopes, H.S., Godoy Jr., W. : Particle swarm optimization for the multidimensional knapsack problem. Proceedings of the Eighth International Conference on Adaptive and Natural Computing Algorithms, Part I. Vol. 4431 of Lecture Notes in Computer Science. Springer-Verlag, Heidelberg, pp. 358-365, (2007).
- 6) Shen, X., Li, Y., Chen, C., Yang, J., Zhang, D. : Greedy continuous particle swarm optimisation algorithm for the knapsack problems. Int. J. Comput. Appl. Technol. 44 (2), pp.137-144, (2012).
- 7) Ucar, H., Tasgetiren, M.F. : A particle swarm optimization algorithm for permutation flow shop sequencing problem with the number of tardy jobs criterion. In: Fifth International Symposium on Intelligent Manufacturing Systems, pp. 237-250, (2006).
- 8) Rosendo, M., Pozo, A. : Applying a discrete particle swarm optimization algorithm to combinatorial problems. Brazilian Symposium on Neural Networks. IEEE Computer Society, Los Alamitos, CA, pp. 235-240, (2010).
- 9) Guangzhao Cui, Limin Qin, Sha Liu, Yanfeng Wang, Xuncai Zhang, Xianghong Cao: Modified PSO algorithm for solving planar graph coloring problem, Progress in Natural Science, Vol.18, pp.353-357, (2008).
- 10) Ling-Yuan Hsu, Shi-Jinn Horng, Pingzhi Fan, Muhammad Khurram, Yuh-Rau Wang, Ray-Shine Run, Jui-Lin Lai, Rong-Jian Chan: MTPSO algorithm for solving planar graph coloring problem, Expert Systems with Application, Vol.38, pp.5525-5531, (2011)
- 11) 陳聡志, 狩野 均 : 遷移確率を用いた PSO によるグラフ色塗り問題の解法, 情報処理学会, 第 72 回全国大会 6V-1, (2010).
- 12) 青木 拓也, 狩野 均 : ハミング距離に基づく遷移確率を用いた PSO によるグラフ色塗り問題の解法, 情報処理学会 第 76 回全国大会 3T-6, (2014).
- 13) Wei-Neng Chen, Jun Zhang, Henry S. H. Chung, Wen-Liang Zhong, Wei-Gang Wu, Yu-hui Shi : A Novel Set-Based Particle Swarm Optimization Method for Discrete Optimization Problems, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 14, NO. 2, APRIL, pp.278-300, (2010).
- 14) 水野, 西原 : 確率的制約充足アルゴリズムにおける局所最適構造, 人工知能学会論文誌, Vol. 16, No. 1, pp. 38-45, (2001).
- 15) Steven Minton, Mark D. Johnston, Andrew B. Philips, Philip Laird: Minimizing Conflicts: A Heuristic Repair Method for Constraint-Satisfaction and Scheduling Problems, Artificial Intelligence, Vol.58, pp.161-205 (1992).
- 16) Tad Hogg, Colin Williams: The Hardest Constraint Problems: A Double Phase Transition, Artificial Intelligence, Vol.69, pp.359-377 (1994).