

# Cuckoo Search によるグラフ色塗り問題の解法

戸田敬太<sup>†1</sup> アランニャ・クラウド<sup>†2</sup> 狩野均<sup>†2</sup>

**概要**：本論文では、Cuckoo Search (CS) を用いた、グラフ色塗り問題 (GCP) の解法を提案する。CS は本来、連続値最適化問題の解法として提案された。そのため、CS を離散値最適化問題に直接適応することが難しい。CS の Lévy Flight (LF) をシグモイド関数によって離散化した研究があるが、LF の移動距離の概念を考慮していない。本研究では、LF の移動距離の概念を考慮した離散化を行った CS を提案する。加えて、離散値最適化問題における CS の改良として、托卵時に前世代の解との比較、更新を行わない手法を示す。比較実験として、ランダムに生成した GCP を用いて、本手法と GA との比較を行った。その結果、本手法の成功回数と探索速度において GA よりも優れていることを確認した。

**キーワード**：Cuckoo Search, 群知能, グラフ色塗り問題, Lévy flight

## 1. はじめに

Cuckoo Search (CS) は、カッコウの繁殖行動と Lévy flight をモデル化した多点探索手法であり、2009 年に Yang らによって提案された[1]。CS は連続値最適化問題の解法として提案されており、この問題において優れた性能を示している。近年では、組合せ最適化問題や制約充足問題などの離散値変数を扱う問題への適用が注目されている。

しかし、CS を離散値最適化問題に対して適用する研究は、まだ少数であり、応用範囲の広い手法は確立されていない。また、3 色グラフ色塗り問題 (GCP) への提案はされていない。ただし、類似した問題として、4 色 GCP への適用[2]と、最低彩色数を求める GCP への適用[3]が提案されている。4 色 GCP へ適用した Yongquan らは、CS を 4 色 GCP へ適用し、さらに他の近似解法とのハイブリッド化を行う手法を提案した。しかし、この手法は他の近似解法を用いることが前提の手法なため、CS 単体では性能が低い。また、Lévy flight の移動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る。最低彩色数を求める GCP へ適用した Halima らの研究は、Lévy flight をバイナリで表現している。しかし、コード化の際にナップサック問題用のコード化を用いているため、GCP には不向きである。またこちらも Lévy flight の移動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る。

3 色 GCP をほかの群知能で解いたものも提案されている[4]。青木らは、CS と同じ多点探索の連続値最適化手法である PSO にハミング距離を導入し、個体間の距離を考慮した離散化を提案した。この手法は PSO の距離の考慮しているため、今まで主流であったバイナリを用いた離散化に比べ、モデル化として優れており、また、高い性能を示している。しかし、この手法で用いているハミング距離は、2 つの個体間の差を表した距離であるため、個体同士の相互

作用がない CS への適用は困難である。

本研究では、Lévy flight を離散値変換した CS による 3 色 GCP における手法を提案する。また、離散値最適化問題における CS の手法自体の改良を示す。

以下では、まず研究分野の概要として通常の CS の動作原理、GCP、関連研究について述べる。次に提案手法について説明し、最後に GCP を対象として、遺伝的アルゴリズム (GA) との比較実験を行い、本手法の有用性を評価する。

## 2. Cuckoo Search

### 2.1 カッコウの繁殖行動

カッコウは「托卵」という特殊な繁殖を行っている。これは自身の卵をほかの種類の子に育てさせる繁殖行動である。カッコウの雛は比較的短期間で孵化し、巣の持ち主 (宿主) の雛より早く生まれることが多い。宿主の卵より早く孵化したカッコウの雛は、宿主の卵を巣から捨てることによって生存率を向上させる。もし、宿主の雛より遅く孵化してしまっても、カッコウの雛は餌を独占し、宿主の雛を餓死させる。こうして、カッコウの雛は宿主の育雛本能に依存して育てられ、成長して巣立っていく。

この特殊な繁殖行動のために、カッコウは托卵を見破られないようにする。たとえば、卵の色や斑紋、雛の鳴き声などを宿主の卵 (雛) に似せるなどがある。

### 2.2 Lévy flight

動物の飛行挙動や採餌行動が Lévy flight の特徴を有していることが様々な研究で示されている[5][6][7][8]。Lévy flight は短距離のランダムな移動である Random walk とは違い、Random walk 中にときおり長距離の移動が挟まれる挙動である (図 1)。Lévy flight に基づく探索方法は、Random walk に基づく探索よりも効果的であることが知られている[9][10]。

### 2.3 Cuckoo Search

Cuckoo Search は連続値最適化問題の解法であり、托卵と呼ばれるカッコウの繁殖行動と Lévy flight と呼ばれる確率分布によって構成される。特徴として、GA や PSO に比べ、パラメータ同士の相関が低く、また変更鈍感なため、パ

<sup>†1</sup> 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  
<sup>†2</sup> 筑波大学システム情報系情報工学科  
Division of Information Engineering, Faculty of Engineering, Information and  
Systems, University of Tsukuba

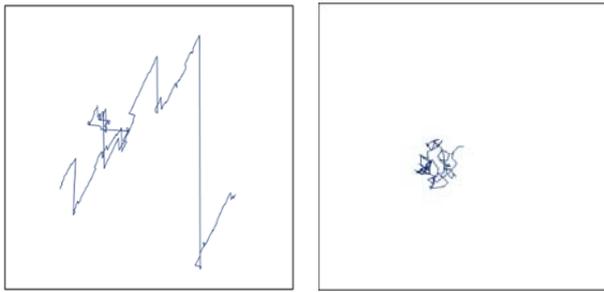


図 1 Lévy flight (左) と Random walk (右) の挙動

```

1  解候補の初期化
2  while (終了条件)
3      for (すべての解候補)
4          step1 : Lévy flight による新たな解候補の探索
5          if (新たな解候補が同等以上)
6              step2 : 解候補を更新
7          end if
8          探索回数 + 1
9      end for
10     for (すべての解候補)
11         step3 : if (托卵発見)
12             step4 : 一様乱数による新たな解候補の探索
13             if (新たな解候補が同等以上)
14                 step5 : 解候補を更新
15             end if
16             探索回数 + 1
17         end if
18     end for
19     最良解の更新
20 end while
    
```

図 2 Cuckoo Search の擬似コード

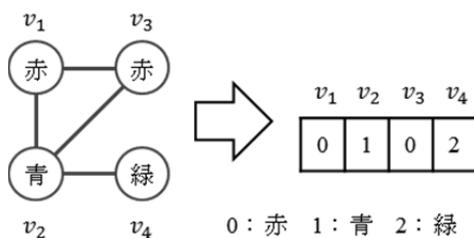


図 3 解のコード化の例

ラメータ設定が容易である。CS にはいくつかの基本ルールがある。

- 各カッコウは、1つの巣に対して1つの卵を托卵。
- 質の高い卵を持った巣は、次世代に継承。
- 巣の数は一定。
- カッコウの卵は一定確率  $P_a \in [0,1]$  で発見され、その場合、巣または卵が捨てられる。その後、新しい巣を新しい場所に構築。

以上のルールに基づいて、CS の基本的な流れは図 2 の擬似コードのように要約できる[11]。

ここで、新しい解  $x_i^{(t+1)}$  は、元の  $x_i^{(t)}$  に Lévy flight を実行することによって生成される[12][13]。

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \times \text{Lévy}(\beta) \quad (1)$$

$$\text{Lévy}(\beta) \sim s = \frac{p}{|q|^{1/\beta}}, \quad (0.3 \leq \beta \leq 1.99)$$

本論文中の式の  $\alpha$ ,  $\beta$  は定数である。 $\alpha (> 0)$  は、問題の大きさに関連するステップサイズであるが、多くの場合  $\alpha = 1$  を用いる。式(1)の  $p$  と  $q$  は次のような正規分布(2)によって導出される。

$$p \sim N(0, \sigma_p^2), \quad q \sim N(0, \sigma_q^2) \quad (2)$$

$$\sigma_p = \left\{ \frac{\Gamma(1 + \beta) \sin(\pi\beta/2)}{\Gamma[(1 + \beta)/2] \beta 2^{(\beta-1)/2}} \right\}^{1/\beta}, \quad \sigma_q = 1$$

これらの Lévy flight による探索、托卵の発見の繰り返しによって、探索空間の最適解を発見することを目指す。

## 2.4 グラフ色塗り問題

グラフ色塗り問題 (GCP) とは、無向グラフに対して、隣接する頂点同士が同じ色にならないように、グラフのすべての頂点を塗り分ける問題である。本研究では3色で塗り分けるグラフ色塗り問題を用いる。この問題は NP 完全問題であり、様々な制約充足アルゴリズムを評価するために用いられている。本研究では、Minton らの論文に示してある方法で3色 GCP を生成する[14]。また、本論文では辺の数  $m$  を頂点の数  $n$  で割ったものを制約密度  $d$  とし、この指標を用いて問題の難易度を分類する。これを用いることにより、手法がどのような問題に対して有効であるかを示すことができる。Hogg らは、GCP は  $d = 2.0 \sim 2.5$  付近で難しい問題になると示している[15]。また、この問題はグラフの構造に強く依存し、局所最適解が生じやすくなるため、探索が困難であるという特徴がある[16]。

ここでは、グラフを計算機で扱えるように解のコード化を行う。4頂点グラフの例を図3に示す。元のグラフを一次元配列に変換し、「赤:0」「青:1」「緑:2」と対応付けで変換する。

## 2.5 関連研究

Zhou らは CS を 4色 GCP に適用し、他の近似解法とのハイブリッドにより改良を行った[2]。カッコウの探索に Walking one strategy, 托卵に Swap and inversion strategy を適用し、組み合わせる近似解法として Greedy transform algorithm (Greedy 法) を用いている。Lévy flight の離散化にはシグモイド関数で求めたバイナリを用いている。この研究では、標準的な PSO と MPSPSO[17] との比較を行い、性能向上を示している。しかし、ICS の参考元の手法である MTPSPSO[18] に比べ、十分な性能を得られていない。また、Greedy 法の性能を上げるために CS を用いているため、CS 単体では性能が低い。さらに、Lévy flight の離散化がバイナリなため、Lévy flight の多様な移動距離の概念を考慮し

ておらず、モデル化としては不十分である。なお、この手法は上記の通り、他の近似解法である Greedy 法とのハイブリッドであり、純粋な CS ではないため、今回は比較対象としない。

HalimaらはCSを最低彩色数を求めるGCPに適用した[3]. Lévy flight の離散化にはシグモイド関数で求めたバイナリを用いている。しかし、コード化の際に参考元の手法であるナップサック問題用のコード化を用いているため、GCPには不向きである。また、ICSと同じく、他の近似解法(RLF法)を用いることを前提としているため、CS単体では性能が低い。また、ICSと同様に Lévy flight の移動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る。なお、この手法は上記の通り、他の近似解法である RLF 法とのハイブリッドであり、純粋な CS ではないため、今回は比較対象としない。

青木らは、3色GCPを解く手法として、CSと同じ連続値最適化手法の群知能である PSO にハミング距離を導入したものを提案した[4]. この手法は PSO の個体間の距離を考慮した離散化を行っているため、今まで主流であったバイナリを用いた離散化に比べ、モデル化として優れており、また、性能を高めることに成功した。しかし、この手法で用いているハミング距離は、2つの個体間の差を表した距離であるため、個体同士の相互作用がない CS への適用は困難である。

### 3. 提案手法

本手法ではCSの3色GCPへの適用を行った。また、Lévy flight の多様な移動距離を考慮した離散値への変換方法と、離散値最適化問題における手法の改良を明示した。

#### 3.1 Lévy flight の離散化

GCPにおいて、Lévy flight は解の要素の変更数によって表現した。各解において  $M$  個の要素数をランダムに変更する(提案手法1)。  $M$  の値は次式(3)に従い、反復ごとの各解によって  $M$  の値は異なる。

$$M = [Lévy] + 1 \quad (3)$$

式(3)の  $[ ]$  はガウス記号であり、Lévy の値は式(1)を用いて導出した。  $M$  の分布を図4に示す。  $\beta$  は表2の値を用いた。

この離散化により、Lévy flight の多様な移動距離を考慮できると考えられる。

#### 3.2 托卵発見時の挙動の改良

CSを離散値最適化問題に適用するにあたり、アルゴリズムの改良を1点行った。従来、図2の13行目のように托卵発見時にも新旧の解同士の比較を行っていたが、この比較を削除する。この理由を以下に説明する。CSは元々、連続値最適化問題に対する手法である。連続値最適化問題は解の値を変更すれば同じ適応度になることは少ない。しかし、離散値最適化問題であるGCPは解の値を変更しても同じ適応度になる場合が多い。そのため離散値最適化問題は

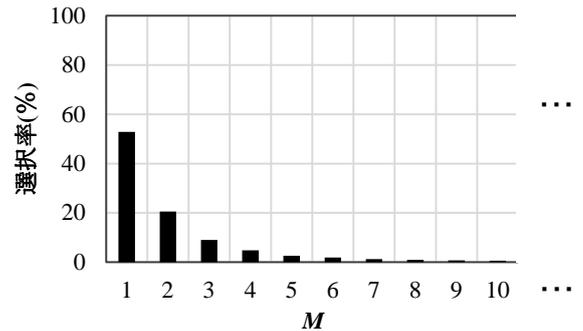


図4 Mの選択確率分布

連続値最適化問題に比べ、解が更新されにくくなっている。そこで、托卵発見の際、新旧の比較をせずに新しい解をそのまま次世代に残す手法を導入した(提案手法2)。ただし、この托卵発見を毎世代行くとランダム探索のような挙動となるため、本手法では托卵発見率を小さくし、数世代に1回発見されるようにした。

また、一様乱数のままではランダム探索のような挙動となるため、托卵発見の際、一様乱数の代わりに Lévy flight を組み合わせて効率的に探索を行う手法もあわせて提案する(提案手法3)。さらに、提案手法3の Lévy flight で用いている式(3)の  $M$  (変更する要素数)の値を固定(=E)した手法もあわせて提案する(提案手法4)。なお、この定数  $E$  はパラメータ設定により決定する。提案手法4種をまとめたものを表1に示す。

### 3.3 グラフ色塗り問題におけるCSのアルゴリズム

アルゴリズムの流れは図2の通りであり、本手法では図2の4行目「step1: Lévy flight による新たな解候補の探索」に、3.1のような離散化、図2の14行目「step5: 解候補を更新」に3.2節のような改良を加えた。変更される要素はランダムに選ばれ、変更する色は元の色と違う色が選ばれるとする。図2の12行目「step4: 一様乱数による新たな解候補の探索」では、 $M$  を一様乱数 ( $[1, \text{頂点数}]$ ) に置き換えて計算した解を生成している。

## 4. 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4種と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(成功回数)と発見までにかかった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、最適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

#### 4.1.1 実験方法

今回、提案手法で設定したパラメータは、 $\beta$ , 托卵発見率, 個体数, 最大探索回数(≠反復回数), 提案手法4の定数  $E$  (後述)である。設定方法として、 $n = 120$ , 制約密度  $d = 2.5$

表 1 提案手法における托卵発見時の相違点

変更点	手法	提案手法 1	提案手法 2	提案手法 3	提案手法 4
		1.CS-R-比有	2.CS-R-比無	3.CS-L-比無	4.CS-定数-比無
要素の変更数の決定方法		一様乱数	一様乱数	Lévy flight	定数 $E$
比較の有無		あり	なし	なし	なし

表 2 実験条件

	提案手法 1	提案手法 2	提案手法 3	提案手法 4	GA
個体数	200	10	10	10	40
最大探索回数 ( $n = 90$ )	$5.0 \times 10^6$				
最大探索回数 ( $n = 120, 150$ )	$1.0 \times 10^7$				
最大探索回数 ( $n = 180$ )	$2.0 \times 10^7$				
托卵発見率 (CS), 突然変異率 (GA)	20%	0.01%	0.1%	0.1%	1.1%
トーナメント数	N/A	N/A	N/A	N/A	2
$\beta$	1.5	1.5	1.5	1.5	N/A
$\alpha$	1	1	1	1	N/A

で生成した問題 100 問を解き、その成功回数の高いものを採用した。結果を表 2 に示す。なお、最大反復回数以外は、他の頂点数においても  $n = 120$  で求めたパラメータを用いる。提案手法 4 の定数  $E$  以外のパラメータは提案手法 3 で設定した値を用い、定数  $E$  の設定を 1~10 の範囲で行った。なお、定数  $E$  の最大値は頂点数である。

#### 4.1.2 実験結果

提案手法 4 の定数  $E$  のパラメータを変えた実験結果を図 5 に示す。なお、成功回数は 98~100 回の間に分布している。また、比較のために提案手法 3 の結果もあわせて示す (図 5 の Levy)。図 5 から以下のことがわかる。

- 定数  $E=3$  のとき、最適値となる。
- このとき、提案手法 3 よりも平均探索回数が少ない。

#### 4.1.3 考察

実験結果から定数  $E$  のパラメータの性質を考察する。定数  $E$  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探索になり、小さすぎると局所最適解に陥ってしまうからであると考えられる。

また実験により、提案手法 4 が提案手法 3 に比べ、高い探索速度を持つことがわかった。しかし、定数  $E$  の値によっては低い探索速度を持つ場合もある。ただし、提案手法 4 は他の提案手法に比べパラメータが 1 つ増えるため、CS の利点である「パラメータ設定の容易さ」が損なわれる。

### 4.2 評価実験

#### 4.2.1 実験方法

本研究では、頂点数 90~180、制約密度 1.5~9.0 の問題を各制約密度で 100 問ずつ解き、成功回数と平均探索回数を比較した。表 2 に実験条件を示す。比較手法として、表 1 の手法 4 種と GA を用いる。GA は一様交叉、トーナメント選択を採用した。

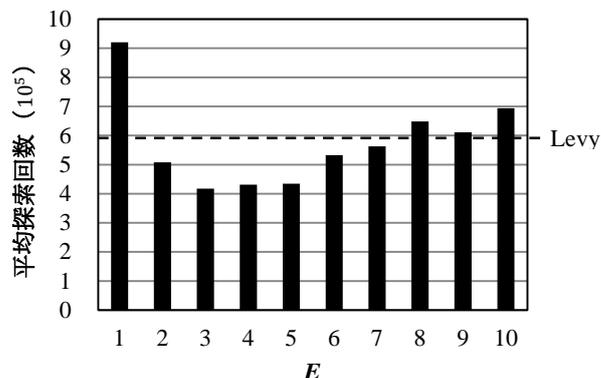


図 5 定数  $E$  のパラメータ比較

#### 4.2.2 実験結果

結果の詳細は、最も難しい  $d = 2.5$  のみ示す。

##### (1) 成功回数での比較

各手法の成功回数での比較を図 6 に示す。図 6 から以下のことがわかる。

- $n = 90$  では、提案手法 1, 2, 3, 4 は GA より成功回数が多い。
- $n = 120$  では、提案手法 2, 3, 4 は GA、提案手法 1 より成功回数が多い。提案手法 3, 4 は提案手法 2 より成功回数が多い。GA は提案手法 1 より成功回数が多い。
- $n = 150, 180$  では、提案手法 1, 2, 3, 4 は GA より成功回数が多い。提案手法 2, 3, 4 は提案手法 1 より成功回数が多い。提案手法 3, 4 は提案手法 2 より成功回数が多い。

##### (2) 平均探索回数での比較

各手法の平均探索回数の比較を図 7 に示す。図 7 から

以下のことがわかる。

- $n = 90,120$ では、提案手法 1, 2, 3, 4 は GA より平均探索回数が少ない。また、提案手法 2, 3, 4 は提案手法 1 より平均探索回数が少ない。
- $n = 150$ では、提案手法 2, 3, 4 は提案手法 1 より平均探索回数が少ない。
- 全頂点数で、提案手法 3, 4 は提案手法 2 より平均探索回数が少ない。また、提案手法 4 は提案手法 3 より平均探索回数が少ない。

#### 4.2.3 考察

実験により、提案手法 2, 3, 4 は GA や提案手法 1 に比べ、 $d = 2.5$ において高い成功回数、探索速度を持つことがわかった。成功回数、探索速度が向上した理由として、次のことが考えられる。

- $d = 2.5$ のような難しい問題では、解の多様性が重要になるため、提案手法 1 の托卵発見時の比較を削除することにより、提案手法 1 に比べ多様性が増加し、托卵発見を解の探索に活かした。
- 解同士の相互作用がないことで、GA に比べ多様性が保て、初期収束が起きにくくなっている。

また、提案手法 3, 4 は提案手法 2 に比べ、 $d = 2.5$ において高い成功回数、探索速度を持つことがわかった。成功回数、探索速度が向上した理由として、次のことが考えられる。

- 提案手法 3, 4 では提案手法 2 に比べランダム探索になりやすいため多様性が増加し、托卵発見を提案手法 2 より効率的に解の探索に活かした。

## 5. おわりに

本研究では、Lévy flight の離散値への適用方法と離散値最適化問題における托卵発見時のアルゴリズムの改良を提案した。提案手法において、Lévy flight の移動距離の再現に成功した。また、GA との比較実験を行った。実験結果により、 $d = 2.5$ のような難しい問題では、GA と比較して提案手法が高い成功回数、探索速度を有することを確認した。さらに、托卵発見時の比較を削除し、変更する要素数を Lévy flight もしくは定数  $E$  にすることにより、さらなる性能向上を示した。

今後は、CS のパラメータについてより詳しい考察を行う。また、アルゴリズムの汎用性を高め、様々な離散化問題に適用できるようにすることが重要と考える。

**謝辞** なお、本研究は JSPS 科研費 15K00296 の助成を受けたものである。

## 参考文献

- [1] Xin-She Yang, Suash Deb, Cuckoo Search via Lévy Flights, *Nature & Biologically Inspired Computing*, no.37, pp. 210-214 (2009).
- [2] Yongquan Zhou, Hongqing Zheng, Qifang Luo, Jinzhao Wu, An improved Cuckoo Search Algorithm for Solving Planar Graph Coloring Problem, *Applied Mathematics & Information Sciences*, Vol.7, No.2, pp.785-792(2013).
- [3] Halima Djelloul, Abdesslem Layeb, Salim Chikhi, A Binary Cuckoo Search Algorithm for Graph Coloring Problem, Vol.5, Issue.3, pp.42-56(2014).
- [4] Takuya Aoki, Claus Aranha, Hitoshi Kanoh, PSO Algorithm with Transition Probability Based on Hamming Distance for Graph Coloring Problem, 2015 IEEE International Conference On Systems, Man, And Cybernetics, pp.1956-1961(2015).
- [5] Brown C., Liebovitch L. S., Glendon R., Lévy flights in Dobe Ju/hoansi foraging patterns, *Human Ecol.*, 35, pp.129-138(2007).
- [6] Pavlyukevich I., Lévy flights, non-local search and simulated annealing, *J. Computational Physics*, 226, pp.1830-1844(2007).
- [7] Pavlyukevich I., Cooling down Lévy flights, *J. Phys. A:Math. Theor.*, 40, pp.12299-12313(2007).
- [8] Reynolds A. M. and Frye M. A., Free-flight odor tracking in *Drosophila* is consistent with an optimal intermittent scale-free search, *PLoS One*, 2, e354(2007).
- [9] G.M. Viswanathan, E.P. Raposo, M.G.E. da Luz, Lévy flights and super diffusion in the context of biological encounters and random searches, *Physics of Life Reviews*, Vol.5, Issue3, pp.133-150 (2008).
- [10] Ahmed Fouad Ali, A Hybrid Gravitational Search with Lévy Flight for Global Numerical Optimization, *Information Sciences Letters*, Inf. Sci. Lett.4, No.2, pp71-83(2015).
- [11] Xin-She Yang, Suash Deb, Engineering Optimisation by Cuckoo Search, *International Journal of Mathematical Modelling and Numerical Optimisation*, pp.330-343(2010).
- [12] Xin-She Yang, *Nature-Inspired Metaheuristic Algorithms Second Edition*, LUNIVER PRESS(2010).
- [13] Rosario Nunzio Mantegna, Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes, *Physical Review E*, Vol.49, no.5, pp.4677-4683(1994).
- [14] 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).
- [15] Tad Hogg, Colin Williams, The Hardest Constraint Problems: A Double Phase Transition, *Artificial Intelligence*, Vol.69, pp.359-377(1994).
- [16] 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).
- [17] 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).
- [18] Ling-Yuan Hsu, Shi-Jinn Horng, Pingzhi Fan, Muhammad Khurram Khan, Yuh-Rau Wang, Ray-Shine Run, Jui-Lin Lai, Rong-Jian Chen, MTPSO algorithm for solving planar graph coloring problem, *Expert Systems with Applications*, Vol.38, pp.5525-5531(2011).

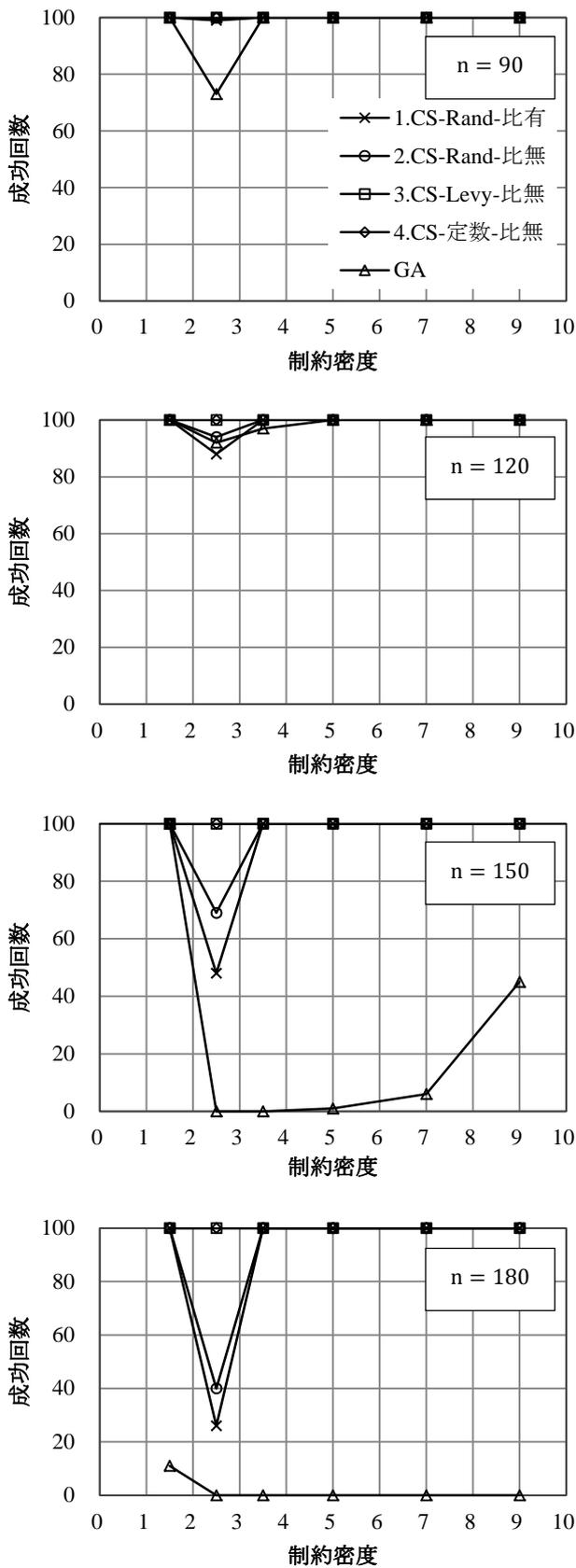


図 6 成功回数での比較 (n = 90,120,150,180)

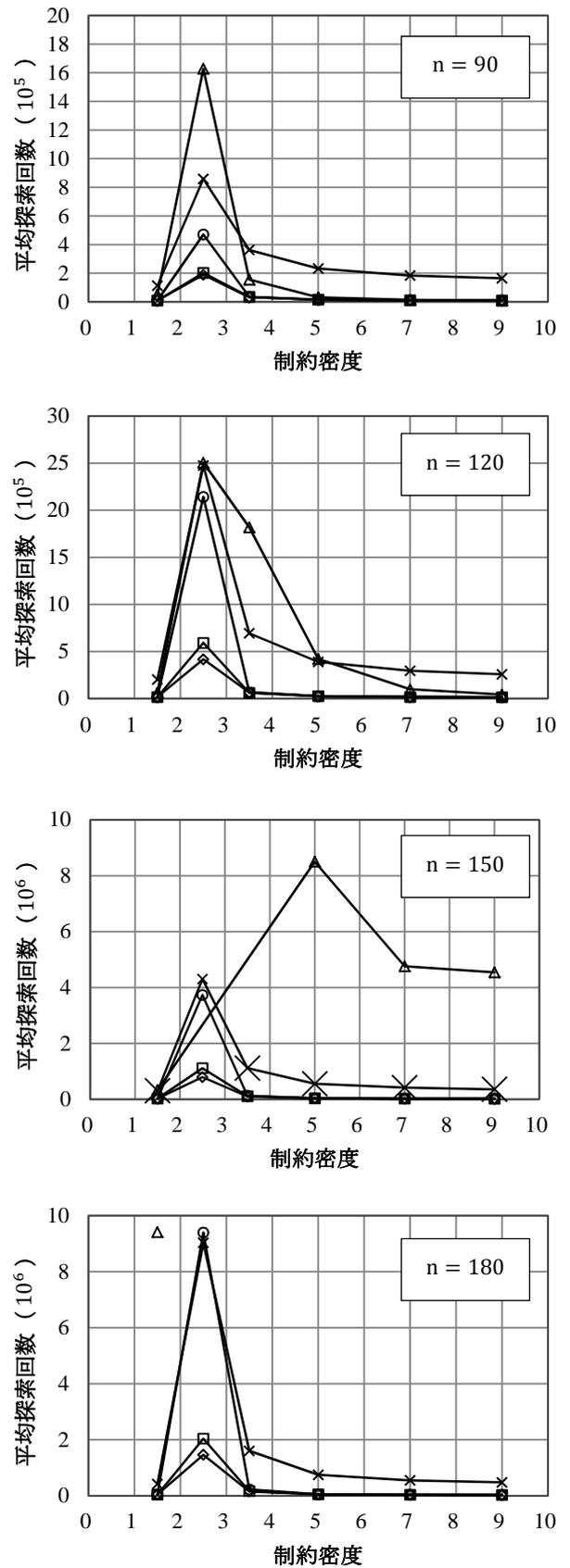


図 7 探索回数での比較 (n = 90,120,150,180)