頂点彩色問題を解く
多点探索型近似解法における
初期解生成

学生番号　2016493
氏名　矢嶋海土
提出年度　令和元年度

目次

１．はじめに

本研究ではグラフに対する頂点彩色問題を扱う.グラフとは頂点とそれを結ぶ辺の集合からなる図形であり,辺で結ばれた頂点同士は互いに隣接しているという.頂点彩色問題とは与えられたグラフ G に対して合法な k-彩色が存在するような最小の k を求める問題である.この最小の k を彩色数という.k-彩色とは G の全ての頂点を 1～k(k∈ℕ)までの数字で表した色に割り当てたものであり,k-彩色が合法であるとは,隣接した頂点同士が異なる色に割り当てられていることをいう.

頂点彩色問題は 4 色問題など離散数学の分野で古くから知られる問題だが[1],オペレーションズ・リサーチやスケジューリングにおける重要な基本問題である.一方で NP 困難と呼ばれる難しい計算クラスに属し[2],グラフサイズの多項式時間で彩色数を求めるのは絶望的であると考えられている.総当たりに基づいた指数時間アルゴリズムも考えられるが,グラフのサイズがある程度大きくなると,実用的な時間内で彩色数を求めるのは困難である.そのため短い時間で良質な解を求めるための近似解法の研究が進められてきた.

頂点彩色問題の解法としては繰返し戦略がよく用いられる.繰返し戦略とは k-彩色可能性問題を,k を減らしていきながら繰り返し解く手法である.k-彩色可能性問題とは与えられたグラフ G および自然数 k に対して,G に合法な k-彩色が存在するかを問う問題である.

近似解法における合法解の探索は,k-彩色の集合である空間の全体において行われる.短い時間で合法解を見つけるには,「良い」初期解から探索を開始することが重要であると考えられる.そこで本研究では,頂点彩色問題に対する多点探索型近似解法の初期解生成に,リサイクル法[3]を適用する.従来初期解生成には大きな注意が払われず,単純なランダム法や欲張り法が用いられることがほとんどであった.内田[3]は k-彩色可能性問題における合法解探索の初期解を,すでに得られた合法な(k+1)-彩色をベースに生成することを提案した.この手法によって初期解を生成すると,従来の手法を用いるよりも k の改善が早く進むことが,TabuCol[4]や PartialCol[5]などの局所探索法(一点探索型近似解法)に関して確認されている.ところが最先端の近似解法は,複数の解を保持して計算を進める多点探索型のアルゴリズムで,局所探索法はそのサブルーチンとして用いられるにすぎない.

本研究ではその事例研究として,最先端の多点探索型近似解法の 1 つである HEAD[6]の初期解生成にリサイクル法を適用する.運用方法をいくつか検討し,計算実験に基づいた比較を行う.その結果,従来の単純な初期解生成法に対しての優位性が確認された.

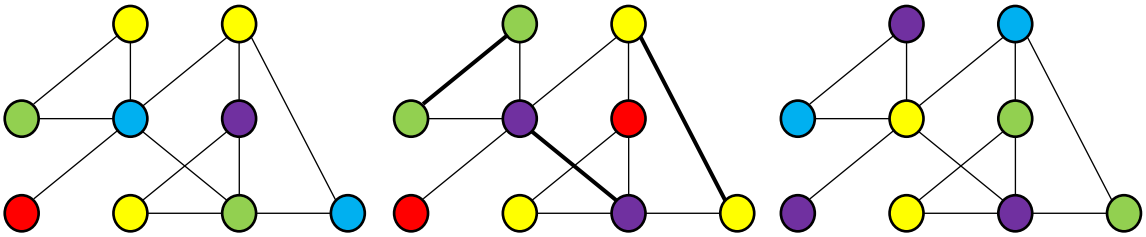
以下, 2 章では頂点彩色問題の定義と応用,及びリサイクル法について説明する.3 章ではリサイクル法の近似解法への適用について解説し,本論文で取り上げる HEAD を紹介する.次の 4 章では計算実験の結果を示し考察を行い,5 章でまとめを行う.

2. 頂点彩色問題

2.1. 定義

頂点彩色問題とは, 頂点集合を V, 辺集合を E とするグラフ G に対して, **合法**な **k-彩色**が存在する**最小の k** を求める問題である. この最小の k を(グラフ G の)**彩色数**という. k-彩色とは G の全ての頂点を 1～k(k∈ℕ)までの数字で表した色に割り当てたものである. k-彩色が合法であるというのは G に含まれるすべての隣接する頂点同士が異なる色に割り当てられていることを示す. これに反して両端の頂点に同じ色が割り当てられている辺を**違反辺**と呼ぶ.

また, 同一の色が割り当てられた頂点の集合を**カラークラス**と呼ぶ. 特に最大位数のカラークラスを**最大カラークラス**, 最小位数のカラークラスを**最小カラークラス**と呼ぶ.



a. 合法な 5-彩色　　b. 非合法　　c. 合法な 4-彩色

図 1　頂点彩色問題の解の例

図 1 において, b は隣接する頂点のうち３組が同じ色になっている(＝違反辺がある)ため合法でない. また, a・c はともに合法であるが, a が「赤・黄・緑・青・紫」の５色で塗られているのに対して, c は「黄・緑・青・紫」の４色なので, c のほうがより良い解であるといえる. 頂点彩色問題を解く際には繰返し戦略という手法がよく用いられる.

2.2.繰返し戦略

繰返し戦略とは頂点彩色問題を解くための手法の 1 つで,合法な k-彩色が存在することが保証された k から開始し,適当な終了条件が満たされるまで,k を 1 減らしては k-彩色可能性問題を解くことを繰返す,というものである.繰返し戦略の概要を以下に示す.

頂点彩色問題に対する繰返し戦略

入力:グラフ G=(V,E).
出力:グラフ G の彩色数の上界.
1: 適当な構築型解法を用いて,合法な k-彩色が存在するような k を決定;
2: while 終了条件が満たされない do
3: 　　k←k-1;
4: 　　k-彩色可能性問題を解く（もし k-彩色が存在しないと結論づけられた場合は while ループを抜ける）
5: end while;
6: while ループにおいて求められた合法な k-彩色の色数 k のうち,最小のものを出力

---

k-彩色可能性問題とは,与えられたグラフ G 及び自然数 k に対して,G に合法な k-彩色が存在するかどうかを問う問題である.k-彩色可能性問題は **NP 完全問題**[2]であり,実用的な時間内で厳密解を求めることは絶望的であるため,**近似解**を求めるために**近似解法**を用いることが現実的な手法となる.多くの近似解法では,適当な探索空間と,探索空間上の解が合法な k-彩色からどれくらい離れているかを評価するためのペナルティ関数 $\rho(c)$ を定め,その最小化を試みるように探索空間の探索を行う.従来最もよく用いられてきた方策は,すべての k-彩色 c の集合を探索空間とし,ペナルティ関数の値として違反辺の本数を用いる,というものである.この場合,図１b におけるペナルティ値 $\rho(c)=3$ となる.明らかに,ペナルティ値 $\rho(c)$ が 0 のとき,かつそのときに限り c は合法である.

k-彩色可能性問題において合法な k-彩色が存在する場合,それを短時間で求めるには,なるべく良い初期解(＝ペナルティ値の低い初期解)から探索を開始することが望ましいと考えられる.ところが従来研究では,局所探索法にせよ多点探索型解法にせよ,初期解はランダムもしくは単純な欲張り法によって生成されることがほとんどであった.この場合,k が小さくなっていくほど,ランダムに生成される k-彩色解のペナルティ値が高くなることが予想され,合法解の探索にかかる時間も増加していくと考えられる.そこで内田は局所探索型の近似解法に対する初期解生成アルゴリズムとしてリサイクル法を提案した[3].

2.3. リサイクル法

内田が提案した**リサイクル法**とは,既知の解を再利用して探索の初期解を生成するものである.その再利用する解とは,繰返し戦略において既に求められている,合法な(k+1)-彩色である.従来のアルゴリズムにおいて k-彩色可能性問題を解く際,繰り返し戦略で得られた合法な(k+1)-彩色は一切反映されない.リサイクル法では合法な(k+1)-彩色をもとに k 色の初期解を生成することで,初期解のペナルティ値を低減し,探索を効率化する狙いがある.リサイクル法の手順を以下に示す.

リサイクル法

入力:グラフ G=(V,E),G における合法な(k+1)-彩色.

出力:k-彩色.

1. 1 つ以上のカラークラスの集合 K⊆{1,…,k+1}とその中の 1 色 j∈K を選ぶ

2. K に含まれるすべての頂点を k+1 色のうち j 以外の色にランダムに変更する

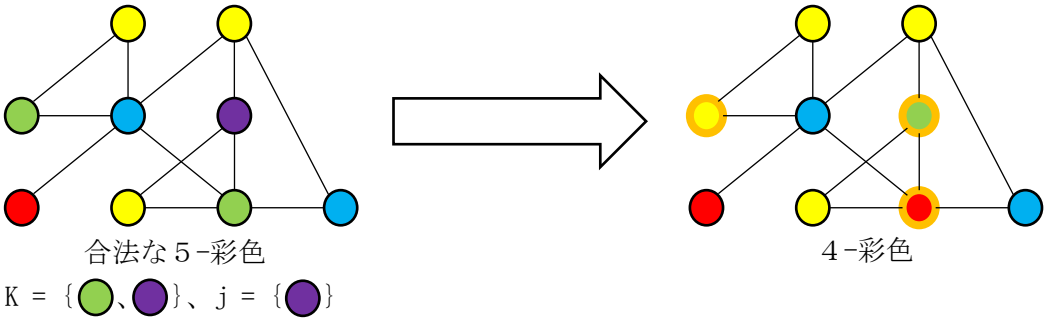


図 2　リサイクル法の手順

リサイクル法によって生成された k-彩色における k は,グラフに使用される色数が高々k 色であることを示すものであり,実際に使用されている色数は k よりも少なくなることがある.

図 2 の例では,合法な 5-彩色から緑と紫の頂点を選び K とする.K に含まれる頂点の色のうち紫を選び j とし,K に含まれるすべての頂点を j 以外の色にランダムに変更する.こうして 4-彩色が生成される.

内田の研究においては,リサイクル法に基づいて最小カラークラスの頂点のみを他色に変更する初期解生成法が提案された.同時に,この手法で得られた初期解のペナルティ値が,従来法に基づいて生成された初期解のペナルティ値を大幅に下回ることが明らかになっている.

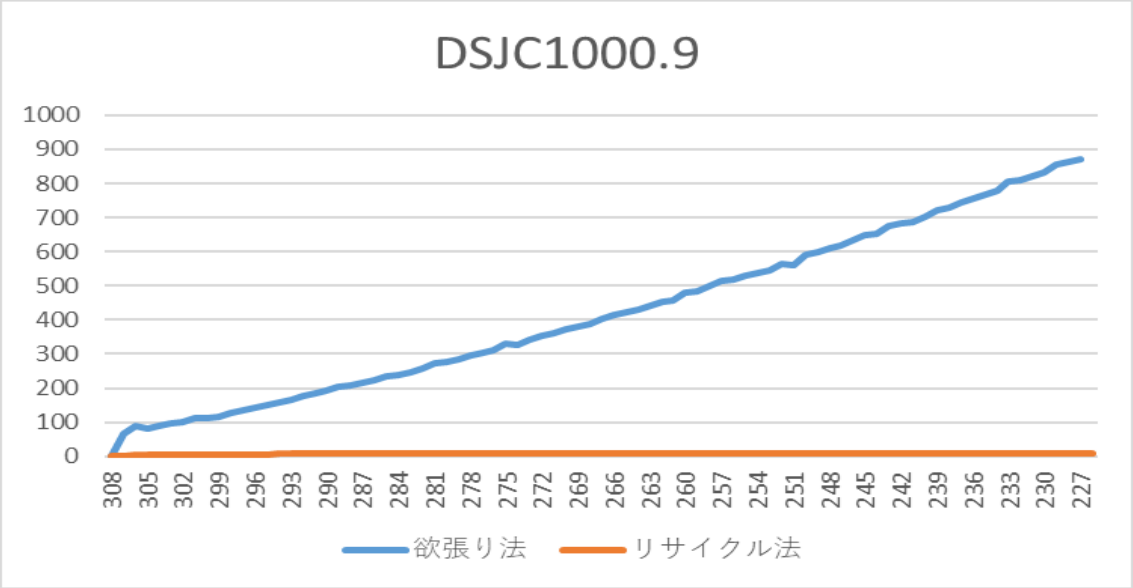


図 3 色数に対するペナルティ値の変化[3]

このグラフは内田の研究で得られた、欲張り法とリサイクル法を使用して k-彩色可能性問題の初期解を生成した場合の色数とペナルティ値の変化をグラフで表したものである. このグラフは The Second DIMACS Implementation Challenge のベンチマーク集[7]の DSJC1000.9 という問題例に対する結果で,縦軸はペナルティ値を,横軸は色数を示している. 繰返し戦略の計算の進行と合わせるため,横軸は降順に値を並べている. 青の欲張り法では色数の減少に伴ってペナルティ値が大きく増加しているのに対し,オレンジのリサイクル法ではペナルティ値は 0 ではないが十分に小さい値であり,色数の減少に伴う増加はほとんどない. このグラフからペナルティ値の低い初期解を生成することに関して,リサイクル法が優れていることがわかる.

3.多点探索型近似解法へのリサイクル法の適用

内田の研究においてリサイクル法は**TabuCol** および**PartialCol** という 2 つの局所探索型のアルゴリズムに適用された.本研究ではリサイクル法を多点探索型の近似解法である**HEAD** に適用する方法を提案する.

k-彩色可能性問題を解く近似解法は,そのアルゴリズムにおける解の探索方法に基づき**一点探索型**か**多点探索型**かに分類される.局所探索法とも呼ばれる一点探索型の近似解法は,ただ 1 つの解を保持し,現在の解にわずかな変更を加えることで得られる近傍の中に改善解があれば移動し,近傍内に改善解がなくなるまで繰り返す手法である.タブー探索に基づいた TabuCol[4]や PartialCol[5]などが例として挙げられる.

多点探索型の近似解法は複数の解を保持し,それらの解を組み合わせたり,ランダムな変形を加えるなどしてより良い解を探索していく.本研究で用いる HEAD の他,進化計算に基づいた HEA[8]や量子物理にインスパイアされた QCOL[9]などがある.計算時間や計算回数などの探索条件を一定以上の水準にした場合,こちらの方が局所探索型の解法よりも良質な解を求めることが可能なため,最先端の近似解法はすべて多点探索型である.

局所探索型	多点探索型
・TabuCol 実行不能解許容モデルのアルゴリズム. HEAD においてもサブルーチンとして使用される.	・HEA 進化計算に基づくアルゴリズム. この解法をベースとして HEAD が考案された.
・PartialCol 不完全解許容モデルのアルゴリズム.	・QCOL 量子物理に基づくアルゴリズム

3.1 HEAD

HEAD(Hybrid Evolutionary Algorithm in Duet)[6]は2018年に提案された,進化計算に基づく多点探索型の近似解法であり,同じく多点探索型の近似解法である**HEA**がベースとなっている.HEA との違いは HEA が 10 個ほどの初期解をもとに探索を進めていくのに対し,HEADは2つの初期解から探索を進めていくことである.HEADのアルゴリズムを以下に示す.

HEAD

入力:グラフ G=(V,E), 自然数 k.
出力:グラフ G における k-彩色.
1: p1,p2,e1,e2,b ←ランダムな k-彩色;
2: t ← 0;
3: while ρ(b) > 0 かつ p1 ≠ p2 do
4: c1 ←GPX(p1,p2);
5: c2 ←GPX(p2,p1);
6: p1 ←TabuCol(c1, τT);
7: p2 ←TabuCol(c2, τT);
8: e1 ← p1,p2,e1 のうちペナルティ値最小の解;
9: b ← e1,b のうちペナルティ値最小の解;
10: if t は τC の倍数 then
11: p1 ← e2;
12: e2 ← e1;
13: e1 ←ランダムな k-彩色
14: end if;
15: t ← t + 1
16: end while;
17: b を出力

1 において p1,p2,e1,e2,b の５つの k-彩色がランダムに生成される.**p1** と **p2** は実際のいわゆる進化の対象となる解であり,**e1** と **e2** は直近のエリート解を保持するための解,**b** は探索全期間中における最良解である.

4,5 では p1 と p2 に対して「GPX」(Greedy Partition Crossover.遺伝アルゴリズムでいう交叉に相当する操作)を行い c1 と c2 の２つの解を生成する.生成された 2 つの解それぞれを局所探索型の解法である TabuCol にかける.e1 と b を適宜更新して探索を続ける.

10～14 では探索が一定の期間を経過するごとに p1,e2,e1 を変更して,再び交叉・探索を行う.計算が終了すれば最良解 b を出力する.これが HEAD のアルゴリズムである.

本研究では上記の 1 の部分における初期解 p1 および p2 のうち 1 つ以上をリサイクル法を用いて生成する.

3.2 提案手法

HEAD にリサイクル法を適用する方法として,次の２通りを検討する.

(1)一方の解をリサイクル法に基づいて生成する方法(もう一方はランダムに生成)

→ **R-min**, **R-max**

(2)双方の解をリサイクル法に基づいて生成する方法

→ **R-minmin**, **R-maxmax**, **R-minmax**

それぞれの手法について説明する.

一方の解を生成

R-min:最小カラークラスの頂点を他色に変更(図 4)

R-max:最大カラークラスの頂点を他色に変更

双方の解を生成

R-minmin:R-min を 2 回行う

R-maxmax:R-max を 2 回行う(図 5)

R-minmax:R-min と R-max を行う

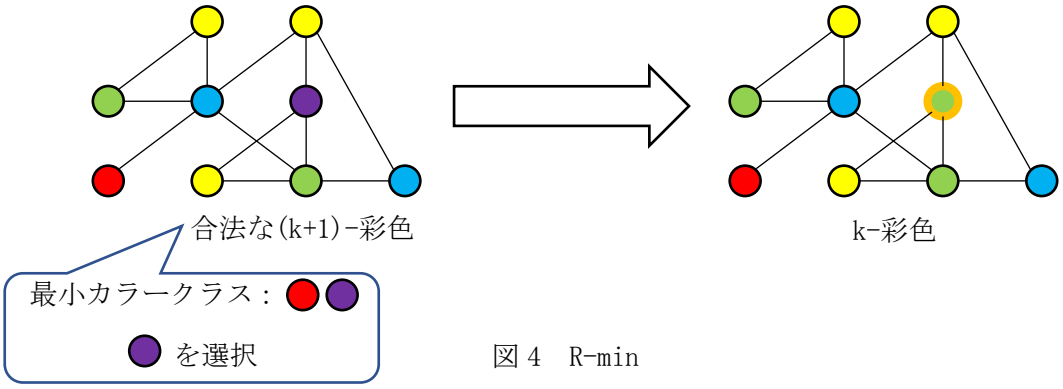


図 4　R-min

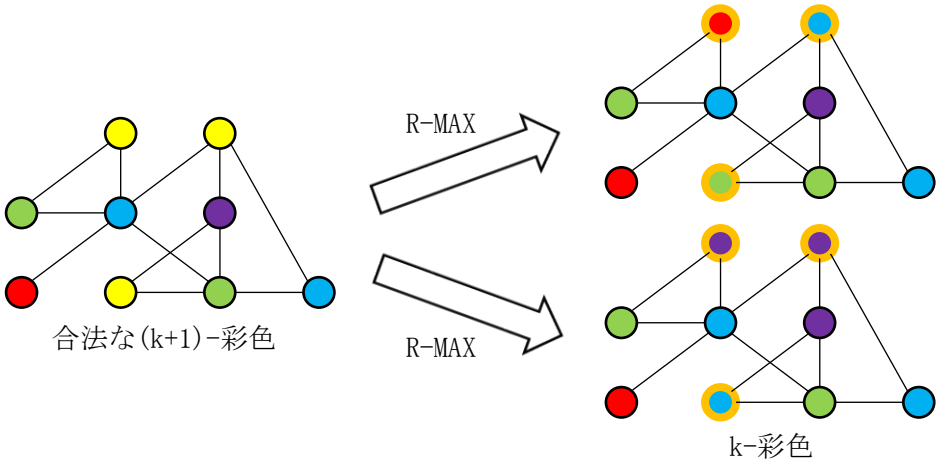


図 5　R-maxmax

内田の研究[3]では,TabuCol,PartialCol の両解法において本研究での R-min にあたる手法が適用された.図 3 からも分かるように,この手法で生成された初期解のペナルティ値が非常に小さい値になることは同研究で証明されている.

内田の研究と本研究におけるもっとも大きな違いは,探索に使用する初期解の個数,およびその使用方法である.内田の研究において用いられた解法は,生成された初期解を直接探索に使用するものである.一方,本研究で用いる HEAD,およびそのベースとなった HEA では生成された複数の初期解を掛け合わせて新たな解を生成し,その解に対して探索が行われる.

HEAD における 2 つの解をともに内田の研究における手法で生成(=R-minmin)すると,それらの解を掛け合わせて生成される解が酷似することとなってしまい,複数の解を保持する利点が薄くなってしまう.そこで,最大カラークラスの頂点の色を変更する(R-max 他)ことで,特に R-maxmax において 2 つの解の類似性を軽減し,多点探索型解法の利点を保つ狙いがある.

4. 実験

リサイクル法に基づく初期解生成法を使用することによる探索効率の変化を測るための実験を行う.

4.1 概要

HEAD の実装には,論文[7]の著者がウェブ上で公開している C++のソースコードを使用した.また,プログラム内の調整可能なパラメータはすべてデフォルト値を用いた.なお HEAD のアルゴリズムにおけるパラメータ τT, τC のデフォルト値として $\tau T = 3\times10^4$, $\tau C = 10$ が定められている.リサイクル法に基づき,合法な(k+1)-彩色を元に初期解を生成するためのプログラムを Python で作成した.また生成された初期解を取り扱えるようにするなど,実験に必要な仕組みを組み込むために HEAD のソースコードを改変した.

繰返し戦略に関して,最初の k の値は予備実験の結果に基づいて人為的に定めた(表 2).また終了条件として計算時間の上限を 10 分とした.ベンチマークに関して,DIMACS 2nd Challenge のうち特に難しいとされる 20 の問題例[7]を用いた.個々の問題例は乱数の種を変えて 10 回解いた.この実験を行ったワークステーションのクロック周波数は最大 3.90GHz,メインメモリの容量は 8GB である.

本実験では,6 通りの初期解生成法(デフォルトのランダム生成,およびリサイクル法に基づいた 5 通りの手法)について HEAD を 10 回走らせるので,各問題例を 60 回解くことになる.この 60 回の試行の中で達成された色数の最良値を k*とする.各初期解生成法が k*を達成した試行回数の,20 個の問題例に関してとった合計を表 1 に示す.2 つの解 p1,p2 のうち一方のみをリサイクル法に基づいて生成する R-min,R-max が,ランダム生成よりも高い頻度で k*を達成している.一方,p1,p2 の両方をリサイクル法に基づいて生成する R-minmin, R-maxmax,R-minmax は,k*を達成する頻度においてランダム生成に及ばない.この原因として,2 つの解の両方を同じ(k+1)-彩色から生成すると,両者が「近い」解となってしまい,結果として探索の多様性を損ねてしまうからではないかと考えられる.

初期解生成法	回数
ランダム	96
R-min	103
R-max	104
R-minmin	94
R-maxmax	91
R-minmax	89

表 1:最良値 k*を達成した試行回数の合計

それぞれの初期解生成法の優劣を測る基準として以下を使用する.

2つの初期解生成法R,R’において,
・RがR’よりも小さい最良値を達成している
・RとR’の達成した最良値は同じであるが,Rの方が高い頻度でその最良値を達成している

この基準を使用する理由として,頂点彩色問題は「より少ない色で頂点を塗り分ける」ことが目的であるため,より少ない最良値を達成した手法が優れた手法であると考えられるからである.

また,近似解法においては,より良い解に到達することに加えて,より早くその解に到達することが求められる.よって,それぞれの手法で実験において適当と思われる色数に到達するのにかかった時間についても比較していく.

問題例	色数
DSJC500.1	20
DSJC500.5	70
DSJC500.9	165
DSJC1000.1	30
DSJC1000.5	120
DSJC1000.9	305
DSJR500.1c	90
DSJR500.5	135
flat300_28_0	45
flat1000_50_0	120
flat1000_60_0	120
flat1000_76_0	120
le450_15c	25
le450_15d	25
le450_25c	30
le450_25d	30
R250.1c	70
R250.5	70
R1000.1c	110
R1000.5	255

表2:各問題例の探索開始時における色数

4.2 結果

まず,一方の解のみをリサイクル法に基づいて生成する R-min,R-max とランダム法(従来のアルゴリズム)において達成した最良値 k と,その達成回数 N(最大:10)を比較する.

問題例	ランダム		R-min		R-max	
	k	N	K	N	k	N
DSJC500.1	12	10	12	10	12	10
DSJC500.5	48	10	48	10	48	10
DSJC500.9	126	4	126	3	126	5
DSJC1000.1	21	10	21	10	21	10
DSJC1000.5	83	3	83	6	83	7
DSJC1000.9	224	8	223	1	223	2
DSJR500.1c	90	3	90	3	89	1
DSJR500.5	124	2	124	4	124	1
flat300_28_0	31	10	30	1	30	1
flat1000_50_0	50	10	50	10	50	10
flat1000_60_0	60	9	60	10	60	10
flat1000_76_0	82	5	82	6	82	7
le450_15c	16	9	16	9	16	9
le450_15d	16	10	16	7	16	8
le450_25c	26	10	26	10	26	10
le450_25d	26	10	26	10	26	10
R250.1c	67	3	65	2	66	3
R250.5	66	3	66	4	66	4
R1000.1c	103	1	104	1	103	1
R1000.5	249	2	248	1	248	1

表 3:ランダム法,R-min,R-max の達成した最良値とその達成回数

実験を通して達成した色数について,R1000.1c において R-min が達成した最良値は,ランダム法が達成した最良値に劣っているが,その他は全てリサイクル法を使用した手法がランダム法に対して同等かより良い数値となっている.また,達成した色数の最良値が同じである 14 個の問題例について,その達成回数を比較すると,R-min,R-max ともに 5 つの問題例でランダム法より多く,2 つの問題例で少ない回数となっている.

次にランダム法と R-min,R-max での探索時間について比較する.各手法において,それぞれの問題例における最良の k に到達した時間の平均(秒)を以下に示す.それぞれの手法が達成した最良の k が異なる場合は,それらのうち最も大きいものを使用する.

問題例	k	ランダム	R-min	R-max
DSJC500.1	12	22.63	16.31	19.64
DSJC500.5	48	48.06	35.54	29.34
DSJC500.9	126	76.98	57.08	105.15
DSJC1000.1	21	0.68	0.59	0.68
DSJC1000.5	83	532.15	345.01	458.10
DSJC1000.9	224	188.17	163.37	128.59
DSJR500.1c	90	0.11	0.11	0.12
DSJR500.5	124	424.27	276.20	256.13
flat300_28_0	31	2.16	1.99	2.56
flat1000_50_0	50	223.51	10.37	16.45
flat1000_60_0	60	436.06	43.18	60.08
flat1000_76_0	82	464.35	470.73	479.93
le450_15c	16	0.50	0.24	0.29
le450_15d	16	0.47	0.27	0.28
le450_25c	26	0.28	0.32	0.34
le450_25d	26	0.33	0.25	0.37
R250.1c	67	0.24	0.15	0.14
R250.5	66	347.43	178.42	310.74
R1000.1c	104	1.92	0.85	1.18
R1000.5	249	437.60	394.68	411.45

表 4:ランダム法,R-min,R-max の達成した最良値とその到達時間の平均

多くの問題例においてリサイクル法(特に R-min)を使用した方が k への到達が早いという傾向がみられる.リサイクル法を用いたことで,flat1000_50_0 や flat1000_60_0 では大幅な改善がなされている一方で,リサイクル法を使用したことで改悪されたと考えられるもの,すなわちリサイクル法を使用した両手法でランダム法よりも遅くなったものは DSJR500.1c,flat1000_76_0,le450_25c の３題である.達成した色数の最良値とその達成回数(表 3)において DSJR500.1c と flat1000_76_0 についてのリサイクル法の優位性が示されており,他方,表 3 でランダム法がより優れていると考えられる le450_15d や R1000.1c については表 4 でリサイクル法の優位性が示されている.

表 1 から従来法よりも劣っていると考えられる 2 つの初期解を生成する手法についても,達成した最良値 k と,その達成回数 N(最大:10)について比較する.

	ランダム		R-minmin		R-maxmax		R-minmax	
グラフ	k	N	k	N	k	N	k	N
DSJC500.1	12	10	12	10	12	10	12	10
DSJC500.5	48	10	48	10	48	10	48	10
DSJC500.9	126	4	126	3	126	1	126	3
DSJC1000.1	21	10	21	10	21	10	21	10
DSJC1000.5	83	3	83	2	83	1	84	7
DSJC1000.9	224	8	223	1	224	6	223	1
DSJR500.1c	90	3	89	2	88	1	89	1
DSJR500.5	124	2	124	5	124	4	124	1
flat300_28_0	31	10	28	1	31	10	31	10
flat1000_50_0	50	10	50	10	50	10	50	10
flat1000_60_0	60	9	60	9	60	8	60	10
flat1000_76_0	82	5	82	4	82	3	82	1
le450_15c	16	9	16	8	15	1	16	5
le450_15d	16	10	16	4	16	7	16	6
le450_25c	26	10	26	10	26	10	26	10
le450_25d	26	10	26	10	26	10	26	10
R250.1c	67	3	68	5	67	4	67	1
R250.5	66	3	66	5	66	3	66	6
R1000.1c	103	1	100	2	98	1	98	1
R1000.5	249	2	248	3	247	1	248	3

表 5:ランダム法,R-minmin,R-maxmax,R-minmax の達成した最良値とその達成回数

グラフによってはリサイクル法を使用した方がよい結果となっているものもあるが,一方の解のみを生成する手法と比較すると,その数は多いとは言えない.

以上の結果より,リサイクル法を使用した初期解の生成が,HEAD において片方の解のみを生成した場合に有効であることがわかった.以降では,双方の初期解を生成する手法がアルゴリズムの改善につながらなかった理由が 2 つの解の距離にあるとして検証実験を行う.

4.3 2つの解の距離

表1から,2つの初期解を生成する手法がHEADにおいて有効でない可能性と,その理由として2つの解の距離が近くなってしまうことと予想した.本節では2つの解の距離という概念と,そのリサイクル法への適用を考案する.

グラフGの2つのk-彩色c_1,c_2(いずれも,合法でなくともよい)において,c_1,c_2でともに同じ色に割り当てられている頂点の個数wをc_1,c_2の**重み**と呼ぶ.また,Gの頂点の個数Vからwを引いたものを, c_1,c_2の**距離**と呼ぶ.c_1,c_2において同じ色に割り当てられた頂点数が多いほど,c_1とc_2の重みが増加し,c_1とc_2の距離が小さく(近く)なる.

頂点彩色問題において多点探索型の近似解法が(局所探索型の近似解法に対して)優れている点として,複数の解を保持して探索を行うことから,局所的最適解からの脱出が容易であることがあげられる.特にただ2つの解を保持して探索を行うHEADにおいて,2つの解の距離が近くなってしまうと,局所探索法と同様に局所的最適解からの脱出が困難になってしまうと考えられる.そのため,解の距離は生成される初期解の「質」において重要であると考えられる.

2つの解の距離をもとに初期解を決定するプログラムをpythonで作成し,簡単な比較実験を行った.実験の概要は以下の通り.

・HEADについては先の実験と同様

・問題例は先の20題の中からDSJC1000.5,flat1000_76_0,R1000.5を使用

・**R-dist**,R-min,R-max,ランダム法の4つの手法について,乱数の種を変更し5回ずつ計算を行う.

R-dist:[R-min(1),R-max(1)]で作成した2つの解と,[R-min(2),R-max(2),ランダム]で作成した3つの解から,**2つの解の距離が最大**となる組み合わせを選び,2つの初期解とする手法.出力されうる初期解の組み合わせは[R-min(1),R-min(2)],[R-min(1),R-max(2)],[R-min(1),ランダム],[R-max(1),R-min(2)],[R-max(1),R-max(2)],[R-max(1),ランダム]の6通り.

・計算時間は5分

・実験に使用したCPUは「Intel core i5-6200U」

実験によって達成した最小の色数 k とその達成回数 N(最大:5)を表 6 に示す.

グラフ	ランダム		R-min		R-max		R-dist	
	k	N	k	N	k	N	k	N
DSJC1000.5	85	1	86	5	85	1	85	5
flat1000_76_0	85	5	84	1	84	2	84	5
R1000.5	253	1	253	4	252	3	252	4

表 6:R-dist の達成した最良値 k とその到達回数 N

実験の結果,達成した最良値とその到達回数の 2 点において,他の全ての手法と比較しても R-dist の優位性が示されている.

次に,R-dist が出力した解の組み合わせとそれぞれの解のペナルティ値を表 7 に示す.縦は出力された初期解の色数を表しており,同一の色数の初期解が 2 つ出力される.横は 1 回目から 5 回目の試行回数を表す.ランダムは RDM とする.1 つの表が 1 つのグラフに対応しており,表の見方は計算回数ごとに縦に見て,

DSJC1000.5 の 1 回目,89 色が「R-max(77)」「RDM(2753)」であれば,「DSJC1000.5 の 1 回目の計算で,89 色の初期解として,R-max で生成されたペナルティ値 77 の解とランダムに生成されたペナルティ値 2753 の解が出力された」という意味である.

色数	1	2	3	4	5
89	R-max(77)	R-max(82)	R-max(75)	R-min(31)	R-min(44)
	RDM(2753)	RDM(2783)	RDM(2795)	RDM(2745)	RDM(2836)
88	R-min(36)	R-max(76)	R-min(39)	R-min(49)	R-min(41)
	RDM(2784)	RDM(2780)	RDM(2877)	RDM(2815)	RDM(2847)
87	R-max(92)	R-max(85)	R-min(38)	R-min(36)	R-min(48)
	RDM(2983)	RDM(2896)	RDM(2881)	RDM(2875)	RDM(2850)
86	R-min(41)	R-max(83)	R-max(97)	R-min(37)	R-min(43)
	RDM(2962)	RDM(2843)	RDM(2927)	RDM(2901)	RDM(3013)
85	R-min(39)	R-min(37)	R-min(43)	R-min(40)	R-min(39)
	RDM(2888)	RDM(2973)	RDM(2931)	RDM(2929)	RDM(2977)

表 7:DSJC1000.5 において R-dist が出力した解とそのペナルティ値

色数	1	2	3	4	5
99	R-min(28)	R-min(47)	R-min(38)	R-min(37)	R-max(66)
	RDM(2512)	RDM(2477)	RDM(2424)	RDM(2486)	RDM(2471)
98	R-min(42)	R-min(39)	R-max(67)	R-min(28)	R-min(41)
	RDM(2499)	RDM(2542)	RDM(2436)	RDM(2536)	RDM(2504)
97	R-min(33)	R-min(38)	R-max(69)	R-min(40)	R-max(56)
	RDM(2561)	RDM(2561)	RDM(2444)	RDM(2539)	RDM(2511)
96	R-min(45)	R-max(60)	R-min(44)	R-min(35)	R-min(36)
	RDM(2555)	RDM(2605)	RDM(2563)	RDM(2539)	RDM(2648)
95	R-max(75)	R-min(39)	R-max(63)	R-min(40)	R-min(37)
	RDM(2711)	RDM(2552)	RDM(2587)	RDM(2683)	RDM(2567)
94	R-max(62)	R-min(36)	R-min(39)	R-min(39)	R-max(62)
	RDM(2668)	RDM(2663)	RDM(2593)	RDM(2667)	RDM(2706)
93	R-max(71)	R-min(42)	R-min(43)	R-min(42)	R-min(39)
	RDM(2671)	RDM(2642)	RDM(2625)	RDM(2644)	RDM(2694)
92	R-min(34)	R-min(43)	R-min(45)	R-min(47)	R-max(73)
	RDM(2708)	RDM(2636)	RDM(2699)	RDM(2727)	RDM(2731)
91	R-min(52)	R-min(38)	R-min(45)	R-max(61)	R-max(69)
	RDM(2729)	RDM(2686)	RDM(2601)	RDM(2746)	RDM(2643)
90	R-min(56)	R-max(80)	R-min(35)	R-max(68)	R-max(60)
	RDM(2736)	RDM(2824)	RDM(2716)	RDM(2735)	RDM(2661)
89	R-min(35)	R-min(48)	R-min(50)	R-min(52)	R-min(51)
	RDM(2695)	RDM(2823)	RDM(2736)	RDM(2799)	RDM(2838)
88	R-max(74)	R-max(62)	R-min(39)	R-max(83)	R-min(45)
	RDM(2732)	RDM(2738)	RDM(2882)	RDM(2857)	RDM(2789)
87	R-min(46)	R-max(79)	R-min(47)	R-max(89)	R-min(31)
	RDM(2836)	RDM(2855)	RDM(2921)	RDM(2794)	RDM(2914)
86	R-min(53)	R-max(79)	R-min(37)	R-min(45)	R-min(40)
	RDM(2877)	RDM(2843)	RDM(2798)	RDM(2937)	RDM(2943)
85	R-min(40)	R-min(51)	R-max(78)	R-min(46)	R-min(47)
	RDM(2852)	RDM(2892)	RDM(2980)	RDM(2868)	RDM(2794)
84	R-min(46)	R-min(34)	R-max(81)	R-max(89)	R-min(40)
	RDM(2971)	RDM(2960)	RDM(2933)	RDM(2999)	RDM(2969)

表8:flat1000_76_0においてR-distが出力した解とそのペナルティ値

色数	1	2	3	4	5
259	R-min(6)	R-min(7)	R-min(6)	R-min(5)	R-min(3)
	RDM(876)	RDM(943)	RDM(973)	RDM(939)	RDM(902)
258	R-max(7)	R-min(4)	R-max(9)	R-min(5)	R-min(5)
	RDM(859)	RDM(927)	RDM(941)	RDM(919)	RDM(955)
257	R-min(3)	R-min(3)	R-min(5)	R-min(6)	R-max(9)
	RDM(952)	RDM(927)	RDM(1029)	RDM(889)	RDM(946)
256	R-min(6)	R-min(3)	R-min(5)	R-min(4)	R-min(4)
	RDM(923)	RDM(953)	RDM(940)	RDM(964)	RDM(924)
255	R-max(11)	R-min(3)	R-min(3)	R-min(4)	R-min(4)
	RDM(889)	RDM(939)	RDM(970)	RDM(958)	RDM(953)
254	R-min(3)	R-min(5)	R-min(3)	R-min(3)	R-max(10)
	RDM(934)	RDM(1011)	RDM(946)	RDM(993)	RDM(938)
253	R-min(2)	R-min(3)	R-min(5)	R-min(3)	R-min(5)
	RDM(945)	RDM(987)	RDM(936)	RDM(944)	RDM(953)
252	R-max(9)	R-min(3)	R-min(5)	R-min(2)	R-min(3)
	RDM(914)	RDM(917)	RDM(975)	RDM(913)	RDM(971)
251	R-min(3)	---	R-min(4)	R-min(3)	R-max(12)
	RDM(967)	---	RDM(916)	RDM(960)	RDM(929)

表 9:R1000.5 において R-dist が出力した解とそのペナルティ値

表 7～9 より,全ての出力において,一方はリサイクル法を用いて生成し,もう一方はランダムに生成されている.このことから,2 つの解の距離について,両方の解を生成する場合に比べて,一方のみを生成した場合の方がより遠くなることがわかる.また,表 6 より,R-min(あるいは R-max)のみで計算した場合よりも R-dist の方がより良い結果となっていることから,初期解の生成において 2 つの解の距離が重要であると考えられる.

2 つの解の距離については,その計算方法から,ランダム法においても R-dist と同等の数値が得られると考えられる.しかし,ランダム法に対する R-dist の優位性が表 6 で示されていることから,ペナルティ値の小さい初期解を使用すること,すなわち,リサイクル法に基づいて初期解を生成することが有効であると考えられる.

5. まとめ

本研究では,頂点彩色問題を解く多点探索型の近似解法における初期解の生成に,リサイクル法を適用した.HEAD を用いた実験の結果,従来のアルゴリズムに対するリサイクル法の有効性を示すことができた.また,多点探索型の近似解法の初期解において,その距離が重要な要素であることが示された.

今後の課題としては,2 つの初期解の距離を保ちつつペナルティ値を下げるような新たな初期解生成法の考案や,HEA のような 3 つ以上の初期解を持つ多点探索型の解法にリサイクル法を適用することなどがあげられる.解の距離やペナルティ値などの要素が初期解の「質」に与える影響について,さらに追及する必要があると考えられる.

謝辞

本論文は多くの方々からのご協力を受け執筆することが出来ました.ご指導いただいた先生,一次審査で貴重なご意見をくださった先生方に心から感謝致します.

参考文献

[1]ロビン・ウィルソン著,茂木健一郎訳(2004).四色問題,新潮社
[2]Garey, M. and Johnson, D. (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman & Company.
[3]内田恭貴(2019).頂点彩色問題に対する近似解法のための初期解生成アルゴリズム.小樽商科大学卒業論文.
[4]Hertz, A. and de Werra, D. (1987). Using tabu search techniques for graph coloring, Computing 39(4): 345-351.
[5]Blöchliger, I. and Zufferey, N. (2008). A graph coloring heuristic using partial solutions and a reactive tabu scheme, Computers & Operations Research 35: 960-975.
[6]L. Moalic and A. Gondran. Variations on memetic algorithms for graph coloring problems. Journal of Heuristics, 24:1-24, 2018.
[7]http://archive.dimacs.rutgers.edu/Challenges/ (2019年12月17日確認)
[8]Galinier, P. and Hao, J. (1999). Hybrid evolutionary algorithms for graph coloring, Journal of Combinatorial Optimization 3: 379-397.
[9]Titiloye, O. and Crispin, A. (2011). Quantum annealing of the graph coloring problem, Discrete Optimization, 8: 376-384.