6B-06

頂点の次数に着目したラムゼーグラフ結合の評価

越村 三幸 †
九州大学 †

頂点数 N の完全グラフ K_N の各辺を２色で塗り分けることを考える。彩色後の K_N の誘導部分グラフ（頂点数 i）を K_i とする。ラムゼーの定理によれば、ある自然数 s と t が与えられたとき、N が十分大きいならば、全辺が第１色の K_s、あるいは全辺が第２色の K_t の少なくとも一つが彩色後の K_N から抽出できる。そのような N のうち最小のものを**ラムゼー数**といい、$R(s,t)$と表す。ラムゼー数未満の N については、第１色の単色 K_s と第２色の単色 K_t のいずれも含まないような K_N が作れるはずで、これを**ラムゼーグラフ**と呼び、$(s,t;N)$-graphと表す。また、$(s,t;N)$-graph の頂点 v の**次数**を、v を端点とする第１色の辺の数と定義する。

$3 \leq s \leq t$ を満たす任意の自然数 s,t について $R(s,t)$ の値を直接的に求める公式はなく、既知のラムゼー数は $R(3,3\cdots9), R(4,4), R(4,5)$ のわずか 9 個である。未知ラムゼー数に関しては上下界が知られており、例えば $R(5,5)$ の現在最良の上下界はそれぞれ $48, 43$ である [1, 3]。したがって、$(5,5;43)$-graph を具体的に示せば $R(5,5)$ の下界を 44 に更新できる。また、$(5,5;43)$-graph が存在しないことを示せば $R(5,5)$ を 43 に確定できる。本研究では $(s,t;N)$-graph を全て枚挙する手法 [2] に着目し、その効率化を考察する。この手法では、二つの小さなラムゼーグラフを結合して、より大きなラムゼーグラフを作る。

1 ラムゼーグラフの結合

F を $(s,t;N)$-graph とする。F の頂点 v の次数を d とし、G_v を v と第１色の辺で繋がっている頂点からなる F の誘導部分グラフ、H_v を v と第２色の辺で繋がっている頂点からなる F の誘導部分グラフ、とする。このときラムゼーグラフの定義より、G_v は $(s-1,t;d)$-graph、H_v は $(s,t-1;N-d-1)$-graph となる [2]。

逆に、$(s-1,t;d)$-graph である G_v と $(s,t-1;N-d-1)$-graph である H_v が何らかの方法で求められていると、G_v と H_v と頂点 v から次のようにして F を作成できる：

> v と G_v の各頂点を辺で結び第１色で彩色すると共に、v と H_v の各頂点も辺で結び第２色で彩色する。そして、G_v の頂点と H_v の頂点も辺で結び、$(s,t;N)$-graph になるように彩色する。

具体的に $s=3, t=7, N=21$ の場合を考えると、F は $(3,7;21)$-graph、G_v は $(2,7;d)$-graph、H_v は $(3,6;20-d)$-graph となる。$R(2,7)=7$、$R(3,6)=18$ であるので、$d<7$ かつ $20-d<18$ である必要があり、これより $2<d<7$ となる。したがって、可能な d と $20-d$ の組合せは $(d,20-d) = (3,17),(4,16),(5,15),(6,14)$ の 4 通りになる。この全ての組合せについて $(2,7;d)$-graph と $(3,6;20-d)$-graph の作成を試みれば、$(3,7;21)$-graph を全て列挙できる。

しかしながらこの手法を単純に実装した計算機実験によると現実的な時間での列挙は難し

An evaluation of Ramsey graph connections with constraints on the degree of vertices
† Miyuki Koshimura, Kyusyu University

表 1　$(3,7;21)$-graph の列挙

			制約なし		制約あり		
d	$\|(2,7;d)\|$	$\|(3,6;20-d)\|$	$\|(3,7;21)\|$	時間（秒）	制約	$\|(3,7;21)\|$	時間（秒）
3	1	7	5674	19	-	5674	19
4	1	2576	601799	10012	≥ 4	598971	11635
5	1	64732	1118001	1049558	≥ 5	513460	296863
6	1	263520	$\geq$633327	$\geq$20 日	≥ 6	331	31613

い。表 1[*1]の「制約なし」欄が、その結果である。表中、$|(2,7;d)|$ 欄は $(2,7;d)$-graph の個数、$|(3,6;20-d)|$ 欄は $(3,6;20-d)$-graph の個数を表す。また $|(3,7;21)|$ 欄は、対応する行の $(2,7;d)$-graph と $(3,6;20-d)$-graph の組合せの結果得られた $(3,7;21)$-graph の個数を表し、「時間（秒）」欄はその計算に要した時間を表す。

$(2,7;6)$-graph と $(3,6;14)$-graph の組合せは $263520(=1\times 263520)$ 通りあるが、20 日かけて試せたのはその内の 2855 通りである。このペースで進むと全てを試すのに約 5 年かかることになる。このような膨大な時間を要する主な原因は重複計算にある、と考えられる。例えば $(3,7;21)$-graph は、$d=3$ の時に 5674 個、$d=4$ の時に 601799 個あるが、重複しているのは 2828 個ある。また、$d=4$ の時と $d=5$ の時の重複は 601695 個ある。

2　次数制約付き結合

本研究では、重複計算を回避するために頂点の次数に着目する。表 1 を上の行から順に埋めていくことにして、$d=3$ の行を先ず埋めたとしよう。得られている 5674 個のグラフには、次数 3 の頂点を持つ $(3,7;21)$-graph が全て含まれているはずである。したがって、$d=4$ の行を埋める時には、次数 3 の頂点を持つ $(3,7;21)$-graph を除外してもよい。つまり、次数 4 以上の頂点のみを持つ $(3,7;21)$-graph を作ればよい。以下、$d=5$ の行の場合は次数 5 以上、$d=6$ の行の場合は次数 6 以上、とすればよい。

表 1 の「制約あり」欄はこのように制約加えてグラフを列挙した結果である。「制約」欄の “$\geq k$” は「各頂点に次数 k 以上の制約を付加して $(3,7;21)$-graph を作成した」とことを示す。4 日ほどで、重複なく全 1118436 個の $(3,7;21)$-graph を求めることができた。

3　終わりに

二つのラムゼーグラフを結合して大きなラムゼーグラフを作りグラフを列挙していく際に、頂点の次数に制約を課すことにより、重複して作られるグラフの数を削減できた。そして、$(3,7;21)$-graph を現実的な時間で列挙できることを示した。今後は、個々の誘導部分グラフに着目した重複削減手法を探究していきたい。

■謝辞　本研究は JSPS 科研費 22K19813 の助成を受けたものです。

参考文献

[1] V. Angeltveit, B. D. McKay: “$R(5,5)\leq 48$,” J. Graph Theory, Vol.89, pp.5-13, (2018)

[2] B. D. McKay, S. P. Radziszowski: “$R(4,5)=25$,” J. Graph Theory, Vol.19, pp.309-322, (1995)

[3] S. P. Radziszowski: “Small Ramsey numbers,” Electron. J. Combin., DS1, (2021)

[*1] 実験には、Intel Xeon W-1207P CPU, 64GB Memory マシン上の Ubuntu 22.04 を用いた。