Applying reinforcement learning to Ramsey problems

Tomasz Dzido[0000−0001−6712−8541]

Department of Information Systems, Gdynia Maritime University, 81-225 Gdynia, Poland
t.dzido@wznj.umg.edu.pl

Abstract. The paper presents the use of reinforcement learning in edge coloring of a complete graph, and more specifically in the problem of determining Ramsey numbers. To the best of our knowledge, no one has so far dealt with the use of RL techniques for graph edge coloring.
The paper contains an adaptation of the method of Zhou *et al.* to the problem of finding specific Ramsey colorings. The proposed algorithm was tested by successfully finding critical colorings for selected known Ramsey numbers. The results of proposed algorithm are so promising that we may have a chance to find unknown Ramsey numbers.

Keywords: Reinforcement learning · Ramsey numbers · Learning-based optimization.

1 Introduction

One of the popular ways looking at Ramsey's theory is in the context of graph theory, and more specifically edge coloring of graphs. To put it quite simply, we want to answer the following question: If we have a complete graph K_n on n vertices where every edge is arbitrarily colored either blue or red, what is the smallest value of n that guarantees the existence of either a subgraph G_1 which is blue, or a subgraph G_2 which is red? This smallest search n is called a 2-color Ramsey number $R(G_1, G_2)$. Initially, only the case when subgraphs G_1 and G_2 are complete subgraphs was considered. Greenwood and Gleason [8] established the initial values $R(K_3, K_4) = 9$, $R(K_3, K_5) = 14$ and $R(K_4, K_4) = 18$ in 1955. Unfortunately, in the case of exact values, there has been very little progress for many years, and for many Ramsey numbers. For example, note that the most recent exact result for a 2-color Ramsey number for two complete graphs is $R(K_4, K_5) = 25$ and was obtained by McKay and Radziszowski in 1995 [9]. Therefore, Ramsey numbers for subgraphs other than complete became popular very quickly. Many interesting applications of Ramsey theory arose in the field of mathematics and computer science, these include results in number theory, algebra, geometry, topology, set theory, logic, information theory and theoretical computer science. The theory is especially useful in building and analyzing communication nets of various types. Ramsey theory has been applied by Frederickson and Lynch to a problem in distributed computations [7], and by

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_46

Snir [13] to search sorted tables in different parallel computation models. The reader will find more applications in Rosta's summary titled "Ramsey Theory Applications" [11].

In recent years, the use of Machine learning (ML) techniques in solving combinatorial problems has significantly increased. Bengio et al. [1] noted that models that are formed by combining ML techniques and combinatorial optimisation strengthen the training procedures. ML is useful especially in discovering or creating certain desirable patterns in graphs, which will be shown later in this article. In 2022, Kai Siong Yow and Siqiang Luo gave a very interesting survey [14]. They reviewed classic graph problems that have been addressed by using learning-based algorithms, particularly those employ ML techniques.

Common approach that is gaining popularity is reinforcement learning (RL), where an agent interacts with its environment in discrete time steps, and learns an (nearly) optimal policy to maximise the reward over a course of actions. There are three key elements in a RL agent, i.e., states, actions and rewards. At each instant a RL agent observes the current state, and takes an action from the set of its available actions for the current state. Once an action is performed, the RL agent changes to a new state, based on transition probabilities. Correspondingly, a feedback signal is returned to the RL agent to inform it about the quality of its performed action [16].

Grouping problems aim to partition a set of items into a collection of mutually disjoint subsets according to some specific criterion and constraints. Grouping problems naturally arise in numerous domains, including, of course, the problem of graph coloring. Zhou, Hao and Duval in [16] presented the reinforcement learning based local search (RLS) approach for grouping problems, which combines reinforcement learning techniques with a descent-based local search procedure. To evaluate the viability of the proposed RLS method, the authors used the well-known graph vertex coloring problem (GCP) as a case study. To the best of our knowledge, no one has so far dealt with the use of RL techniques for graph edge coloring. All the more, there are no known attempts to use these techniques in estimating the value of Ramsey numbers. The most commonly used heuristics are local search, simulated annealing or tabu search. In this paper, we present how the reinforcement learning based local search (RLS) approach presented in [16] can be used to find lower bounds of some Ramsey numbers. Our proposed application of RLS approach belongs to the category of learning generative models of solutions. This method was used, among others, in [4] for the solution of the flow-shop scheduling problem.

To sum up the introduction, the main method used in the article is an adaptation of the RLS aproach presented in [16] and the rest of this paper is organized as follows. Section 2 provides useful notation and definitions. Section 3 describes the application of the RLS method in Ramsey's theory. In Section 4, we discussed the results obtained from the computer simulations and present possible improvements. The article ends with a short summary and an indication of the direction of further research.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_46

2 Notation and definitions related to graphs and Ramsey numbers

Let $G = (V(G), E(G))$ be an undirected graph. K_m denotes the complete graph on m vertices, C_m - the cycle of length m, P_m - the path on m vertices and $K_{1,m}$ - the star of order $m+1$. An edge k-coloring of a graph G is any function $f : E(G) \to \{1, 2, 3, ..., k\}$. In this paper we only consider edge 2-colorings. Since colorings involve only two colors (blue and red) futher on will be reffered to as colorings (instead of edge 2-colorings). For graphs G_1, G_2 a coloring f is a $(G_1, G_2; n) - coloring$ if and only if f is a 2-coloring of the complete graph K_n and f contains neither a G_1 colored with color 1 nor a G_2 colored with color 2.

Definition 1. *The Ramsey number $R(G_1, G_2)$ for graphs G_1 and G_2 is the smallest positive integer n such that there is no $(G_1, G_2; n) - coloring$.*

Definition 2. *A coloring $(G_1, G_2; n)$ is said to be critical if $n = R(G_1, G_2) - 1$.*

The following theorem is a well-known result on Ramsey number for two cycles, which was established independently in [6] and [12].

Theorem 1 ([6], [12]). *Let m, n be integers, where $3 \le m \le n$.*

$$R(C_m, C_n) = \begin{cases} 6 & (m,n) = (3,3), (4,4), \\ 2n-1 & m \textit{ is odd and } (m,n) \ne (3,3), \\ n-1+\frac{m}{2} & m \textit{ and } n \textit{ are even and } (m,n) \ne (4,4), \\ max\{n-1+\frac{m}{2}, 2m-1\} & m \textit{ is even and } n \textit{ is odd.} \end{cases}$$

As we have seen, in the case of two cycles we know everything, however only partial results for C_m versus stars $K_{1,n}$ are known. The most known general exact result for even cycles is:

Theorem 2 ([15]).

$$R(C_m, K_{1_n}) = \begin{cases} 2n & \textit{for even } m \textit{ with } n < m \le 2n, \\ 2m-1 & \textit{for even } m \textit{ with } 3n/4+1 \le m \le n. \end{cases}$$

Besides the exact values many lower and upper bounds for various kinds of graphs have appeared in the literature. Radziszowski in his regularly updated dynamic survey "Small Ramsey Numbers" [10] lists all known nontrivial values and bounds for Ramsey numbers. Lower bounds on Ramsey numbers are mostly proved by giving a witness that doesn't have the desired Ramsey property. Such a witness (called a critical coloring) could be part of a general construction, or found 'at random' by a heuristic algorithm. While lower bounds on Ramsey numbers can be established by giving one coloring which does not have the desired property, to prove an upper bound one must give an argument implying that all colorings of a certain order complete graph have the desired property. Mostly, this is done by using general or specific theorems to vastly reduce the number of possible counter-example colorings. The remaining colorings sometimes must then be enumerated by a computer to verify that none of them is a critical coloring.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_46

3 RLS applied to determining critical colorings for some Ramsey numbers

In the problem of determining the exact value of the Ramsey number, it is often the case that we can find it for large n, but we do not know the value for small cases. The algorithm presented below can successfully fill this gap. This algorithm can be adapted to various types of graphs, but for the purposes of the article, we will only present a case of applying it to two cycles and to the case of a cycle and a star.

We already know everything about Ramsey numbers of the type $R(C_k, C_m)$ (see Theorem 1), so we will be able to easily verify the obtained results. Let us assume that we are looking for the critical $(C_k, C_m; n) - coloring$, where $k \leq m$. To apply the proposed RLS approach to this purpose, we need to specify the search space Ω, the neighborhood, the evaluation function $f(S)$, final acceptance criterium and method of choosing initial state.

First, for a given partition of complete graph K_n into 2 graphs: G_1 and G_2, where $V(G_1) = V(G_2) = V(K_n)$, $E(K_n) = E(G_1) \cup E(G_2)$ and $E(G_1) \cap E(G_2) = \emptyset$, we define space Ω to be the family of all possible edge 2-colorings $S = \{G_1, G_2\}$ such that the subgraph G_i is colored with the color i, where $i \in \{1, 2\}$. The neighborhood of a given coloring is constructed by changing the color of an edge belonging to at least one forbidden cycle.

The objective function and the final acceptance criterium are defined intuitively: $f(S)$ is simply equal to the number of edges that are in at least one forbidden cycle in S. Accordingly, a candidate solution S is the desired critical coloring if $f(S) = 0$.

As the initial state a complete graph K_n is taken. Originally, the RLS procedure starts with a random solution taken from the search space Ω. In order to increase the size and speed up finding the desired critical Ramsey coloring, we can use various known graph-type-specific properties. They can reduce the number of remaining edges to be colored or narrow down the number of edges of a given color. The use of this type of methods requires specialist knowledge of Ramsey properties for a given type of graphs, here we will focus only on possible examples for the studied two-cycle problem.

The length of a shortest cycle and the length of a longest cycle in G are denoted by $g(G)$ and $c(G)$, respectively. A graph G is weakly pancyclic if it contains cycles of every length between $g(G)$ and $c(G)$. $G \cup H$ stands for vertex disjoint union of graphs, and the join $G + H$ is obtained by adding all of the edges between vertices of G and H to $G \cup H$. The following interesting properties are known.

Theorem 3 ([5]). *Let G be a graph of order $n \geq 6$. Then $max\{c(G), c(\overline{G})\} \geq \lceil 2n/3 \rceil$, where $\overline{G}$ is the complement of G.*

Theorem 4 ([2]). *Every nonbipartite graph G of order n with $|E(G)| > \frac{(n-1)^2}{4} + 1$ is weakly pancyclic with $g(G) = 3$.*

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_46

Theorem 5 ([3]). *Let G be a graph on n vertices and m edges with $m \geq n$ and $c(G) = k$. Then*

$$m \leq w(n,k) = \frac{1}{2}(n-1)k - \frac{1}{2}r(k-r-1), \text{ where } r = (n-1) \bmod (k-1),$$

and this result is the best possible.

For example, consider the number $R(C_8, C_8)$. Theorem 1 leads to $R(C_8, C_8) = 11$. That means we are looking for critical $(C_8, C_8; 10) - \mathit{coloring}$. On the other hand, we know that $R(B_2, B_2) = 10$, where $B_2 = K_2 + \overline{K_2}$. By combining the value of this number and Theorem 3, we get the property that the remaining number of edges to be colored by adapting the RLS algorithm is at least 9 less (where B_2 and C_7 have the greatest possible intersection). Looking at it from another angle, if the number of edges of one color exceeds 27 (combining Theorems 4 and 5, where $w(10,7) = 27$), in this color we have cycle C_8, which we avoid. This means that the number of edges of K_{10} in each color belongs to the set $\{18, ..., 27\}$.

We define a probability matrix P of size $n \times 2$ (n is the number of edges and 2 is the number of colors). An element p_{ij} denotes the probability that the i-th edge $e \in E(G)$ is colored with the j-th color. Therefore, the i-th row of the probability matrix defines the probability vector of the i-th edge and is denoted by p_i. At the beginning, all the probability values in the probability matrix are set as $\frac{1}{2}$. It means that all edges are colored with one of the two available colors with equal probability. To achieve a local optimum, the current solution (coloring) S_t at instant t is then enhanced by DB-LS, a descent-based local search algorithm which iteratively improves this solution by a neighboring solution of better quality according to the evaluation function. In our case, we simply change the color of each edge belonging to any cycle to the opposite color, and calculate the objective function. This process stops either when a critical coloring is found (i.e., a solution with f(S) = 0), or no better solution exists among the neighboring solutions (in this later case, a local optimum is reached). It means that for current solution S_t the locally best solution $\overline{S_t}$ is generated (of course, if it exists). Next, for each edge e_i, we compare its colors in S_t and $\overline{S_t}$. If the edge does not change its color (say c_i), we reward the selected color c_i (called correct color) and update its probability vector p_i according to:

$$p_{ij}(t+1) = \begin{cases} \alpha + (1-\alpha)p_{ij}(t) & \text{if } j = u \\ 1 - (\alpha + (1-\alpha)p_{ij}(t)) & \text{otherwise.} \end{cases}$$

where α ($0 < \alpha < 1$) is a reward factor. When edge e changes its color to the opposite color (say c_v, $v \neq u$), we penalize the discarded color c_v (called incorrect color) and update its probability vector p_i according to:

$$p_{ij}(t+1) = \begin{cases} (1-\beta)p_{ij}(t) & \text{if } j = v \\ 1 - ((1-\beta)p_{ij}(t)) & \text{otherwise.} \end{cases}$$

where β ($0 < \beta < 1$) is a penalization factor. In the next step, a smoothing technique is applied on the probability vector of each edge $e_i \in E(G)$. For this

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_46

we first calculate the value $p_{iw} = max\{p_{i1}, p_{i2}\}$. Then we check if $p_{iw} > p_0$ is true, where p_0 is a smoothing probability. If so, then we update probability vector p_i according to:

$$p_{ij}(t) = \begin{cases} \rho \cdot p_{ij}(t) & \text{if } j = w \\ 1 - \rho \cdot p_{ij}(t) & \text{otherwise.} \end{cases}$$

Once the probability of a color in a probability vector achieves a given threshold (i.e., p_0), it is reduced by multiplying a smoothing coefficient (i.e., $\rho < 1$) to forget some earlier decisions.

At each iteration of RLS, each edge e_i needs to select a color c_j from two available colors according to its probability vector p_i. As in [16], we adopted the hybrid selection strategy which combines randomness and greediness and is controlled by the noise probability ω. With a noise probability ω, random selection is applied; with probability $1 - \omega$, greedy selection is applied. The purpose of selecting a color with maximum probability (greedy selection) is to make an attempt to correctly select the color for an edge that is most often falsified at a local optimum. Selecting such a color for this edge may help the search to escape from the current trap. On the other hand, using the noise probability has the advantage of flexibility by switching back and forth between greediness and randomness. Also, this allows the algorithm to occasionally move away from being too greedy [16].

Now consider the case of Ramsey numbers of the type $R(C_m, K_{1,n})$. The procedure is basically the same as above, with the difference that the objective function is calculated in a different way. The objective function is defined as follows: $f(S)$ is equal to the sum of the number of edges that are in at least one forbidden cycle C_m colored with color 1 and the number of vertices of red (color 2) degree at least n, and the number of red edges in the red neighborhood of these vertices. This surprising last number came from observing the behavior of the machine learning algorithm considered above for two cycles. Probably the current geometric similarity speeds up the algorithm.

4 Computational experiments

The basis of our software framework consisted of the package NetworkX, which includes a graph generator, tool to find all cycles of a given length and several other utilities for graph manipulation. All tests were carried out on a PC under 64-bit operating system Windows 11 Pro Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz, RAM 16GB compiled with aid of Python 3.9.

To obtain the desired colorings, each instance was solved 10 times independently with different random seeds. Each execution was terminated when a Ramsey coloring is found or the number of iterations without improvement reaches its maximum allowable value (500). As a result of the computational experiments, the values of all learning parameters were determined. Table 1 shows the descriptions and setting of the parameters used for our experiments. The considered colorings and the times of receiving the appropriate colorings are

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_46

presented in Table 2. For small cases, it was from a dozen to several dozen iterations, for larger ones it did not exceed 700. The case of coloring $(C_8, C_8; 10)$ was considered in 3 versions: random coloring or random $(C_8, C_8; 8)$-coloring or random $(C_8, C_8; 9)$-coloring was given at the start, respectively. In addition to the colorings presented in Table 2, a number of calculations lasting several dozen hours were also performed in order to find the $(C_8, K_{1,10}; 15) - coloring$. The calculations were started with various types of $(C_8; K_{1,10}; 14) - colorings$. Each of the calculations stopped at some point and for at least 600 iterations the locally best solution was no longer corrected. Due to this fact, it can be assumed that $R(C_8, K_{1,10}) = 15$ and Conjecture 1 from [15] holds for $m = 8$ and $k = 10$.

Parameter	Description	Value
ω	noise probability	0.2
α	reward factor for correct color	0.1
β	penalization factor for incorrect color	0.5
ρ	smoothing coefficient	0.55
p_0	smoothing probability	0.955

Table 1. Parameters of Algorithm RLS.

Coloring	Comp. time(s)	Coloring	Comp. time(s)
$(C_6, C_6; 7)$	$<$ 1s.	$(C_{10}, C_{10}; 10)$	$<$ 1 s.
$(C_6, C_8; 8)$	$<$ 1s.	$(C_{10}, C_{10}; 11)$	159-473 s.
$(C_6, C_8; 9)$	9-13 s.	$(C_6, K_{1,6}; 10)$	$<$ 30 s.
$(C_8, C_8; 10)$ ver 1.	106-372 s.	$(C_6, K_{1,7}; 10)$	$<$ 30 s.
$(C_8, C_8; 10)$ ver 2.	101-118 s.	$(C_8, K_{1,5}; 9)$	29-552 s.
$(C_8, C_8; 10)$ ver 3.	12-85 s.	$(C_8, K_{1,6}; 11)$	17-23 min.
$(C_8, C_{10}; 10)$	45-180 s.	$(C_8, K_{1,9}; 14)$	69-236 min.
$(C_8, C_{10}; 11)$	6-78 min.	$(C_8, K_{1,10}; 14)$	62-189 min.

Table 2. The times of determining the given colorings.

The frequent spread of computation time comes from the random, and therefore unpredictable, pre-coloring of the graph. In order to improve the speed of the algorithm, various pre-coloring can be applied and a number of useful Ramsey properties can be used. Examples of such actions are presented above. The structure of the graphs and the objective function used are also important. It is possible that other objective functions than those presented above in the article can be used. The same is true of machine learning parameters. It is likely that for other classes of graphs they should be adapted to them. The performed calculations indicate that within a dozen or so hours at most, we are able to determine the coloring for a graph with 15 vertices (i.e. having 105 edges), as long as the

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_46

task is to color all the edges. Python was used for the implementation, but there are faster languages, such as ANSI C.

5 Conclusion

The adaptation of the method of Zhou, Hao and Duval [16] and the obtained results show that reinforcement learning can be considered as another and promising heuristic that can be used in determining Ramsey numbers. Appropriate selection of the graph structure, objective function, learning parameters, precoloring or even fixing the colors of certain edges can truly bring measurable results in determining unknown values of Ramsey numbers. Future work can be started by applying this method to the open problems contained in [10].

References

1. Bengio, Y., Lodi, A., Prouvost, A.: Machine learning for combinatorial optimization: a methodological tour d'horizon. European Journal of Operational Research **290**(2), 405–421 (2021)
2. Brandt, S.: A sufficient condition for all short cycles. Discrete Applied Mathematics **79**, 63–66 (1997)
3. Caccetta, L., Vijayan, K.: Maximal cycles in graphs, Discrete Mathematics **98**, 1–7 (1991)
4. Ceberio, J., Mendiburu, A., Lozano, J. A.: The plackett-luce ranking model on permutation-based optimization problems. In: Proceedings of the IEEE congress on evolutionary computation (CEC), pp. 494–501, IEEE, Cancun, Mexico (2013)
5. Faudree, R.J, Lesniak, L., Schiermeyer, I: On the circumference of a graph and its complement. Discrete Mathematics **309**, 5891–5893 (2009)
6. Faudree, R.J., Schelp, R.H.: All Ramsey Numbers for Cycles in Graphs. Discrete Mathematics **8**, 313–329 (1974)
7. Frederickson, G., Lynch, N.: Electing a leader in a synchronous ring. Journal of Association for Computing Machinery **34**, 98–115 (1987)
8. Greenwood, R.E., Gleason, A.M.: Combinatorial Relations and Chromatic Graphs. Canadian Journal of Mathematics **7**, 1–7 (1955)
9. McKay, B.D., Radziszowski, S.P.: $R(4,5) = 25$. Journal of Graph Theory **19**, 309–322 (1995)
10. Radziszowski, S.: Small Ramsey Numbers. The Electronic Journal of Combinatorics, Dynamic Survey 1, revision 16 (2021)
11. Rosta, V.: Ramsey Theory Applications. Electronic Journal of Combinatorics, Dynamic Survey 13 (2004)
12. Rosta, V.: On a Ramsey Type Problem of J.A. Bondy and P. Erdoös, I & II. Journal of Combinatorial Theory, Series B **15**, 94–120 (1973)
13. Snir, M.: On parallel searching. SIAM Journal Computing **14**, 688–708 (1985)
14. Yow, K.S., Luo, S.:Learning-Based Approaches for Graph Problems: A Survey, https://arxiv.org/abs/2204.01057. Last accessed 4 Jan 2022
15. Zhang, Y., Broersma, H., Chen, Y.:Narrowing down the gap on cycle-star Ramsey numbers. Journal of Combinatorics **7**(2-3), 481–493 (2016)
16. Zhou, Y., Hao, J-K., Duval, B.: Reinforcement learning based local search for grouping problems: A case study on graph coloring. Expert Systems With Applications **64**, 412–422 (2016)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_46