

Colorarea grafurilor

Căramidă Iustina-Andreea - 322CA

Facultatea de Automatică și Calculatoare
Universitatea Politehnica București
`iustina.caramida@stud.acs.upb.ro`

Abstract. Problema colorării nodurilor unui graf (*graph coloring* - *GCP* sau *colorarea grafurilor*) este una dintre cele mai importante probleme, fapt datorat numărului mare de situații din viața reală care se pot rezuma la o problemă de colorare a unui graf. În această temă, voi prezenta câteva comparații între diferiți algoritmi dedicați acestei probleme, precum Greedy, Dsatur și RLF din punct de vedere al timpului de execuție, al memoriei folosite și al performanței în raport cu numărul de noduri și muchii.

Keywords: Brute-Force · Greedy · DSatur · RLF · Colorarea grafurilor.

1 Introducere

1.1 Descrierea problemei

Colorarea grafurilor este o problemă centrală în teoria grafurilor [1]. Ea constă în alegerea unui set de culori pentru nodurile unui graf, astfel încât niciun nod adiacent să nu primească aceeași culoare.

Problema colorării graficului a început cu încercarea lui Francis Guthries de a colora toate țărilor pe harta Angliei. Acesta a presupus inițial că patru culori sunt suficiente pentru a procesa orice hartă, astfel încât să nu fie asociate două țări vecine cu aceeași culoare. Această sarcină este doar una dintre cele peste 200 de probleme [2], legate de aria analizei grafice cromatice, iar această situație poate fi tradusă prin colorarea fiecărui vârf dintr-un graf în care marginile sale ar reprezenta vecinătatea dintre două regiuni [3].

Colorarea grafurilor este o problemă de tip NP-hard bine studiată cu aplicații importante în optimizarea combinatorie și într-un domeniu de cercetare activ, cu multe aplicații practice [4] în inginerie, cum ar fi *alocarea registrelor*, *atribuirea frecvenței*, *potrivirea șablonului* și *programări*. În consecință, colorarea grafurilor a fost subiectul unor cercetări intense [5] [2].

Colorarea grafurilor este asociată cu două tipuri de colorare: colorarea vârfurilor și colorarea marginilor. Scopul ambelor tipuri de colorare este de a colora întregul graf fără conflicte. Prin urmare, vârfurile adiacente sau muchiile adiacente trebuie să fie în culori diferite. Numărul celor mai mici posibile culori care pot fi utilizate pentru colorarea grafului se numește **număr cromatic**. Pe măsură ce numărul de vârfuri sau muchii dintr-un graf crește, complexitatea problemei

crește și ea. Din această cauză, fiecare algoritmul nu poate găsi numărul cromatic exact și pot fi, de asemenea, diferențe în timpul lor de execuție [6]. Pentru a obține o soluție mai bună pentru colorarea grafurilor, mulți algoritmi euristici și meta euristici au fost inventați [7].

1.2 Specificarea algoritmilor aleși

Cerință: Fie un graf neorientat G cu N noduri și M muchii. Problema cere să asociem o culoare fiecărui nod, astfel încât oricare două noduri adiacente (conectate printr-o muchie directă) să aibă culori diferite. Care este numărul minim de culori necesare pentru a colora toate nodurile conform restricției menționate anterior? [8]

Brute Force: Când încercăm să oferim o soluție la această problemă, primul instinct este de a folosi o abordare "Brute Force". Acest lucru ar duce la o soluție care ar fi din punct de vedere al timpului de execuție exponențială, făcând această soluție inutilă pentru cazuri mari. A spune că nu putem găsi un algoritm eficient deoarece acesta nu există ar fi la fel ca și când am spune că problema nu are nicio soluție eficientă [9]. În prezent, există algoritmi care se ocupă să rezolve problema colorării grafurilor, deși obțin un număr cromatic apropiat de al grafului în schimbul unui timp rezonabil sau rezultate rapide care sunt suficient de utile [10].

Greedy Algorithm: Logica algoritmului ia vârfurile grafului unul câte unul, urmând o ordine (care poate fi aleatorie) și atribuie prima culoare disponibilă fiecărui vârf [11]. Deoarece este un algoritm euristic, soluția oferită de acesta poate să nu fie optimă. Cu toate acestea, o alegere corectă a ordinii vârfurilor pentru colorarea lor poate oferi o soluție optimă pentru orice graf. În practică, algoritmul Greedy produce soluții rapid practicabile, deși aceste soluții pot fi "sărace" pe baza numărului de culori pe care algoritmul le cere, în comparație cu numărul cromatic al grafului.

DSatur Algorithm: Algoritmul DSatur (abreviere din engleză pentru "Degree Saturation"), propus de Brelaz (1979), se comportă foarte asemănător cu algoritmul Greedy, cu excepția că, în acest caz, ordonarea vârfurilor este generată de algoritmul însuși. La fel ca în algoritmul Greedy, ordonarea a fost decisă înainte ca orice vârf să fie colorat, ordinea vârfurilor fiind decisă euristic pe baza caracteristicilor colorării parțiale a grafului la momentul în care se selectează fiecare dintre vârfuri [12]. În cel mai rău caz, complexitatea sa are aceeași situație ca și în Algoritmul Greedy, deși în practică poate fi luat în considerare și faptul că monitorizarea saturației vârfurilor necolorate produce o complexitate puțin mai mare. Este de reținut că Algoritmul DSatur este **exact** pentru grafurile bipartite [13].

RLF Algorithm: Algoritmul RLF (abreviere din engleză pentru “Recursive Largest First”), propus de Leighton (1979), lucrează prin colorarea unui graf cu o singură culoare pentru fiecare iterație a algoritmului, în loc de un vârf per repetare. În fiecare iterație, algoritmul caută seturi de vârfuri independente din graf, care vor fi asociate cu aceeași culoare. Acel set independent a fost eliminat din graf, iar subgraful rămas va continua în același mod, până când subgraful menționat este gol, caz în care toate vârfurile vor fi atribuite unei culori, producând astfel o soluție ce satisface toate cerințele [11].

1.3 Evaluarea soluțiilor

Sursele vor fi testate pe grafuri de dimensiuni diferite, de la 10 la 2 000 de noduri, cu un număr de muchii de la 10 la 1 000 000. Pentru fiecare graf se va genera o configurație aleatoare de noduri și muchii, iar apoi se va testa performanța algoritmilor pe aceste grafuri. Pentru fiecare algoritm, se va calcula timpul de execuție și numărul de culori folosite pentru a colora graful. De asemenea, se va calcula și numărul de noduri colorate în fiecare iterație a algoritmului, pentru a vedea cât de eficient este algoritmul în ceea ce privește numărul de iterații necesare colorării grafului.

References

1. J. Bondy and U. Murty, Graph Theory - Graduate Texts in Mathematic. Springer, 2008.
2. Z. Ādām Mann and A. Szajkò, “Average-case complexity of backtrack search for coloring sparse random graphs,” Journal of Computer and System Sciences, vol. 79, no. 8, pp. 1287–1301, 2013.
3. “Backtrack: An $o(1)$ expected time algorithm for the graph coloring problem,” Information Processing Letters, vol. 18, no. 3, pp. 119–121, 1984.
4. N. Barnier and P. Brisset, “Graph coloring for air traffic flow management,” Annals of Operations Research, vol. 130, 03 2002.
5. “The application of a graph coloring method to an examination scheduling problem,” Institute for Operations Research and the Management Sciences (INFORMS), vol. 11, no. 5.
6. A. Murat and B. Nurdan, “A performance comparison of graph coloring algorithms,” International Conference on Advanced Technology Sciences (ICAT’16), vol. 4, pp. 1–19, 12 2016.
7. Z. Mann, “Complexity of coloring random graphs: An experimental study of the hardest region,” Journal of Experimental Algorithmics, vol. 23, pp. 1–19, 03 2018.
8. Enunțul problemei
9. M. Garey and D. Johnson, Computer and Intractability: A Guide to the Theory of NP-Completeness, 01 1979.
10. D. Porumbel, J.-K. Hao, and P. Kuntz, “An evolutionary approach with diversity guarantee and well-informed grouping recombination for graph coloring,” Computers Operations Research, vol. 37, pp. 1822–1832, 10 2010.
11. L. Ouerfelli and H. Bouziri, “Greedy algorithms for dynamic graph coloring,” 2011 International Conference on Communications, Computing and Control Applications, CCCA 2011, 03 2011.

12. À. E. Eiben, J. K. Van Der Hauw, and J. I. van Hemert, "Graph coloring with adaptive evolutionary algorithms," *Journal of Heuristics*, vol. 4, no. 1, pp. 25–46, 1998.
13. D. Brèlaz, "New methods to color the vertices of a graph," *Commun. ACM*, vol. 22, pp. 251–256, 04 1979.
14. Graph coloring - Wikipedia
15. A Comparison of Parallel Graph Coloring Algorithms
16. A Performance Comparison of Graph Coloring Algorithms
17. Moodle - Analiza algoritmulor