

Kolorowanie Grafów za Pomocą Algorytmów Mrówkowych

Ant Colony Optimalization for Graph Coloring Problem

Gabriel Wechta, 250111

12.06.2020

1 Abstrakt

Referat zawiera przedstawienie problemu kolorowania grafów, krótki zarys historyczny, przykłady instancji problemu w codziennych zastosowaniach, matematyczny opis metod używanych w dwóch algorytmach ACO, porównanie parametrów algorytmów oraz wyników dla przykładowego problemu. Referat analizuje ANTCOL [2] i MMGC [5]. Praca powstała korzystając ze źródeł zawartych w bibliografii. Istnieje wiele rozsądnych metaheurystyk, które można zastosować do problemu kolorowania grafu takich jak Tabu Search, Simulated Annealing, Genetic Algorithms, wszystkie dają obiecujące rezultaty, niemniej ten referat skupia się na Ant Colony Optimalization.

2 Wprowadzenie

Pierwsze rozważania dotyczące problemu kolorowania grafów zostały formalnie przedstawione w 1852 roku. Podczas tworzenia mapy Anglii, Francis Guthrie postulował, że cztery kolory wystarczą, żeby pokolorować mapę w taki sposób, aby żadne dwa regiony o wspólnej granicy nie miały tego samego koloru. Owe przesłanki były czysto intuicyjne i oparte o doświadczenie. Powyższe twierdzenie nosi teraz nazwę twierdzenia o czterech barwach i doczekało się poprawnego dowodu dopiero w 1976. Niestety owe ograniczenie istnieje tylko dla grafów planarnych (między innymi map). W ogólności optymalne (najmniejsza liczba kolorów) pokolorowanie dowolnego grafu jest problemem NP-trudnym [1].

2.1 Formalizacja problemu

Kolorowaniem grafu $G = (V, E)$ będziemy nazywać nadawanie wierzchołkom G liczb naturalnych tak, aby nie istniała krawędź $e \in E$ pomiędzy dwoma wierzchołkami z przypisaną taką samą liczbą.

Liczbą chromatyczną grafu G , będziemy nazywać najmniejsze n takie, że

pokolorowanie G n kolorami jest legalne. Oznaczać ją będziemy $\chi(G)$. Przykładowo:

1. Dla K_m – graf pełny o m wierzchołkach, $\chi(K_m) = m$, gdyż każdy wierzchołek musi mieć inny kolor niż pozostałe.
2. Dla T – drzewa o co najmniej dwóch wierzchołkach, $\chi(T) = 2$.

Optymalne kolorowanie to takie pokolorowanie G , które zużywa dokładnie $\chi(G)$ kolorów.

2.2 Zastosowania

Zastosowania rozwiązań problemu kolorowania grafów znajduje się między innymi w zagadnieniach planowania pracy z ograniczonymi zasobami, kolorowania map, harmonogramowania. Należy zwrócić uwagę, że sudoku jest instancją tego problemu.

Spójrzmy na komórki jako na wierzchołki grafu, gdzie wszystkie wierzchołki muszą być różne dla wierzchołków w tej samej kolumnie, rzędzie oraz podkwadracie. Taki graf G ma wtedy 81 wierzchołków, 810 krawędzi, a jego liczba chromatyczna $\chi(G) = 9$. Rozwiązanie sudoku sprowadza się wtedy do jego optymalnego pokolorowania.

Ponadto ostatnio pojawiły się zastosowania kolorowania grafów przy użyciu ACO do algorytmów proponowania treści w sieciach społecznościowych [3].

2.3 Algorytmy mrówkowe

Algorytmy mrówkowe są stosowane głównie do szukania najlepszych ścieżek w grafach. Zachowanie mrówek w algorytmie zainspirowane jest zachowaniem prawdziwych mrówek podczas szukania pożywienia. Analogia opiera się na komunikacji między mrówkami oraz zysku całej kolonii poprzez działania wszystkich mrówek. W prawdziwym świecie mrówki rozpoczynają poszukiwanie jedzenia od błędzenia losowego, gdy jedna z nich znajdzie pożywienie wraca do kolonii pozostawiając za sobą ślad feromonów. Gdy kolejna mrówka napotka na ten ślad przestaje poruszać się losowo i zaczyna faworyzować ścieżkę z feromonami. Największe natężenie feromonów jest na ścieżce, którą porusza się najwięcej mrówek. Po pewnym czasie feromony zaczynają wyparowywać co powoduje, że najczęściej przemierzane ścieżki mają najwięcej feromonu, a ścieżkami rzadziej odwiedzanymi mrówki tracą zainteresowanie. Wielką zaletą algorytmów mrówkowych jest szukanie paru najlepszych rozwiązań jednocześnie.

Jakość wyników w implementacji algorytmu mrówkowego opiera się głównie na dobrym doborze parametrów takich jak: rozmiar populacji, tempo parowania feromonu, czy inne parametry występujące w danej implementacji. Przykładowo: zwiększenie populacji daje większą eksplorację, ale zabiera czas, być może niezbędny do przeprowadzenia wystarczającej ilości iteracji, aby mrówki zaczęły wytyczać najlepsze ścieżki. Standardowy algorytm mrówkowy w każdej iteracji wygląda następująco:

1. Mrówki startują, wybierając ścieżki według zadanego prawdopodobieństwa.
2. Mrówka na wybranej trasie pozostawia feromon o natężeniu proporcjonalnym (choć niekoniecznie) do funkcji celu.
3. Feromony ulatniają się.
4. Wybierane są najlepsze rozwiązania.

Zagadnienie kolorowania grafu potrzebuje licznych restartów i badania wielu konkurencyjnych ścieżek na raz, dlatego wybieramy do nich algorytmy mrówkowe.

3 Algorytmy

Problem kolorowania grafu jest NP-trudny [1]. Istnieją algorytmy dające dokładne rozwiązania [4; 6; 7], natomiast ich wysoka złożoność obliczeniowa powoduje, że można je stosować wyłącznie do małych instancji problemu. Co za tym idzie, potrzebne są metody heurystyczne i aproksymacyjne. W referacie zostaną przedstawione dwa algorytmy ANTCOL i MMGC, przy czym drugi jest modyfikacją pierwszego, która na celu miała poprawę mankamentów ANTCOL.

3.1 ANTCOL

Algorytm ANTCOL został przedstawiony przez Coste i Hertza [5], daje wyniki bliskie optimum. Procedura jest następująca:

Populacja mrówek iteracyjnie koloruje graf, poprzez dodawanie do klas kolorów kolejnych napotkanych wierzchołków. Jakość kolorowania $f(s)$ jest oceniana według poniższej funkcji.

$$f(s) = \frac{1}{q(s)} \quad (1)$$

gdzie $q(s)$ to użyta liczba kolorów w rozwiązaniu s . Ślad feromonowy τ_{ij} występuje pomiędzy dwoma $u_i, u_j \in V$, takimi, że w danym s mają przypisany taki sam kolor. ANTCOL używa algorytmu ANTRLF (Ant Recursive Largest First), naturalnego rozszerzenia RLF na algorytmy mrówkowe.

RLF jest algorytmem konstruktywnym - rozpoczyna z pustym rozwiązaniem, następnie konsekwentnie rozszerza obecne, częściowe rozwiązanie o nowy komponent, którego dodanie pozostawia częściowe rozwiązanie poprawnym. Rozszerzenie RLF na algorytmy mrówkowe polega na równoległych i niezależnych wywołaniach RLF [9]. W ANTRLF w fazie k mrówka konstruuje: C_k – klasę koloru zawierającą wierzchołki grafu o danym kolorze, W – zbiór wierzchołków bez przydzielonego koloru, które mogą być dodane do C_k , oraz B – zbiór wierzchołków bez przydzielonego koloru, które **nie** mogą być dodane do C_k . Ponadto w ANTRLF definiuje się następujące metody heurystyczne.

$$\eta_{ik} = \deg_B(v_i) \quad (2)$$

$$\eta_{ik} = \deg_{B \cup W}(v_i) \quad (3)$$

$$\eta_{ik} = |W| - \deg_W(v_i) \quad (4)$$

Powyższe metody oraz ich kombinacje definiują strategię ANTRLF. Doświadczenie pokazuje, że najlepsze wyniki są osiągane przez (2), bądź losowy wybór wierzchołka z W . Ślad feromonowy dla v_i, v_j nie mających wspólnej krawędzi, należących do tej samej klasy koloru, obliczany jest następująco (na początku wszystkie ślady feromonów mają wartość 1).

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \sum_{s \in S_{ij}} \frac{1}{q(s)} \quad (5)$$

Zaś w przypadku obliczania śladu dla niepokolorwanego v_i , w celu dodania go do C_k , następująco.

$$\tau_{ik} = \frac{\sum_{j \in C_k} \tau_{ij}}{|C_k|} \quad (6)$$

Gdzie $\rho \in (0, 1)$ stanowi tempo parowania (zapominania) feromonu.

S_{ij} to podzbiór kolorowania, gdzie v_i i v_j należą do tej samej klasy koloru.

τ_{ik} zawiera wszystkie informacje o śladach feromonowych v_i oraz dotychczas dodanych wierzchołkach do C_k . Dzięki wyżej wymienionym metodom obliczania współczynników, można zadać prawdopodobieństwo, z którym mrówka wybierze v_i do C_k .

$$P_{ik} = \begin{cases} \frac{\tau_{ik}^\alpha \eta_{ik}^\beta}{\sum_{j \in W} \tau_{jk}^\alpha \eta_{jk}^\beta} & v_i \in W \\ 0 & v_i \notin W \end{cases} \quad (7)$$

Standardowe ustawienie parametrów dla ANTCOL $\rho = 0.5, \alpha = 2, \beta = 4$. α, β służą do ustawienia wpływu odpowiednio τ_{ik} oraz η_{ik} na obliczanie P_{ik} .

Funkcja jakości f nie jest idealna. Istnieje wiele różnych kolorowań z tą samą liczbą kolorów, o różnych rozkładach wierzchołków w klasach kolorów. Jeżeli rozważymy kolorowanie, w którym wierzchołki są rozłożone równomiernie pomiędzy klasy kolorów, to takie rozwiązanie zdaje się być bardziej obiecujące od takiego, w którym wierzchołki są rozłożone z dużymi różnicami w gęstości.

Warto również zwrócić uwagę, że metoda propagacji informacji o feromonie w ANTCOL (5) ma pewną wadę, zważywszy na fakt, że w każdej iteracji wszystkie mrówki propagują swój feromon, wpływając tym samym na wybory innych mrówek, przez co ANTCOL ma tendencję do znajdowania rozwiązań średnich. Ponadto strategię wynikającą z metod heurystycznych (2), (3), (4) słabo radzą sobie z eksploracją rozwiązania. Poprawę tych wad wykazuje MMGC.

3.2 MMGC

Max-Min Graph Coloring poprawia działanie ANTCOL głównie poprzez wyłączenie propagacji feromonu mrówek, które nie znalazły najlepszych rozwiązań oraz modyfikację ANTRLF - parametry dodawania pierwszych wierzchołków do

pustych C_k ulegają zmianie.
Korzystając z metody:

$$\eta_{ik} = \deg_W(v_i) \quad (8)$$

algorytm przydziela v_i do C_k z P_{ik} .

$$P_{ik} = \begin{cases} \frac{\eta_{ik}^\gamma}{\sum_{j \in W} \eta_{jk}^\gamma} & v_i \in W \\ 0 & v_i \notin W \end{cases} \quad (9)$$

W celu poprawienia funkcji jakości rozwiązania w MMGC zmienia się $f(s)$ tak, aby miała dwie cechy. Jedna cecha dąży do kolorowania z mniejszą liczbą konfliktów, zaś druga do minimalizacji liczby klas kolorów.

$$f(s) = \sum_{i=1}^q 2|C_i||\overline{C_i}| - \sum_{i=1}^q |C_i|^2 \quad (10)$$

Gdzie $\overline{C_i}$ to zbiór konfliktów kolorów w klasie C_i . Podczas każdej iteracji mrówki z najwyższą wartością $f(s)$ są najwyżej oceniane i tylko tych mrówek wyniki są brane pod uwagę przy liczeniu śladu feromonu.

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \sum_{i=1}^q |C_i|^2 \quad (11)$$

Aby ta propagacja miała sens, niezbędne jest narzucenie ograniczeń na ślady feromonu, mianowicie $\tau_{ij} \in [\frac{f(s_{gb})(1-\sqrt{P_{best}})}{\rho(\frac{|V|}{2}-1)\sqrt{P_{best}}}, \frac{f(s_{gb})}{\rho}]$. Gdzie s_{gb} oznacza najlepszy globalny wynik pod koniec iteracji, a P_{best} jest stałą oznaczającą prawdopodobieństwo stworzenia optymalnego kolorowania [2] oraz zainicjowanie wszystkich wartości τ_{ij} na $\frac{|V|^2}{\rho}$. W modyfikacji ANTRLF dla MMGC pojawia się nowy parametr γ , który pozwala modyfikować wyższość wierzchołka nad innymi, podczas dodawania pierwszego wierzchołka do klasy koloru. Standardowe ustawienie parametrów dla MMGC $\rho = 0.04, \alpha = 2, \beta = 4, \gamma = 3, P_{best} = 0.05$, dla którego są poniższe wyniki.

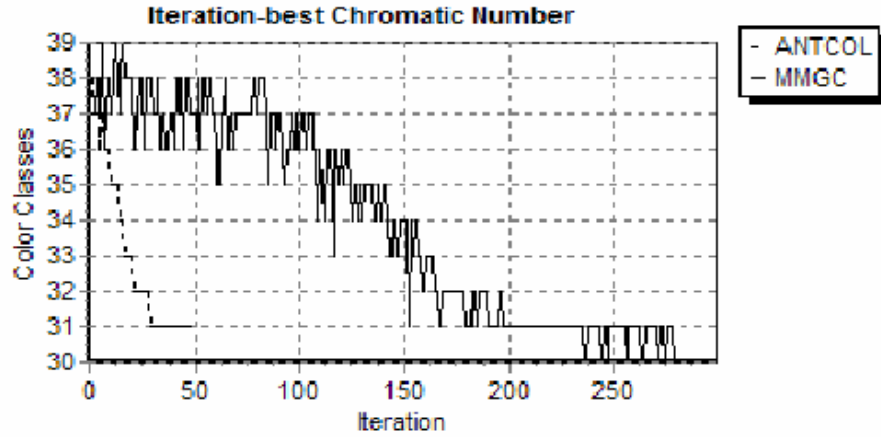
4 Wyniki i porównanie

MMGC, zważywszy na sposób konstruowania legalnych rozwiązań, potrzebuje więcej czasu na eksplorację, kiedy ANTCOL szybko zbiega do rozwiązań średnich (Rys 1), MMGC zajmuje to więcej czasu, ale nie pozostaje w lokalnym minimum.

Wyniki Salari i Eshghi [2] oraz Costy i Hertza [5] dla algorytmów MMGC i ANTCOL przedstawia Rys. 2.

W pierwszej kolumnie znajdują się metody generowania grafów, kolumna Best-Known to najlepsze znane wyniki dla danych grafów.

We wszystkich przypadkach MMGC dał lepszy wynik niż ANTCOL, natomiast



Rysunek 1: Wykres porównujący zbieżności ANTCOL i MMGC

Graph	Best-Known	MMGC				ANTCOL			
		k	No. Iterations / No. Ants	No. Success / No. Trials	CPU time (Sec.)	k	No. Iterations / No. Ants	No. Success / No. Trials	CPU time (Sec.)
DSJC $G_{125,0.1}$	5	5	150 / 70	4/5	155	6	50 / 100	5/5	173
DSJC $G_{125,0.5}$	17	18	150 / 70	5/5	186	18	50 / 100	4/5	125
DSJC $G_{125,0.9}$	44	44	150 / 70	4/5	133	44	50 / 100	4/5	97
DSJC $G_{250,0.1}$	8	9	300 / 20	5/5	650	9	50 / 100	5/5	713
DSJC $G_{250,0.5}$	28	30	300 / 20	5/5	736	31	50 / 100	3/5	548
DSJC $G_{250,0.9}$	72	74	300 / 20	4/5	713	75	50 / 100	3/5	421
DSJC $G_{500,0.1}$	12	15	500 / 20	5/5	3942	15	50 / 100	5/5	3125
DSJC $G_{500,0.5}$	48	53	500 / 20	4/5	4131	55	50 / 100	3/5	2638
DSJC $G_{500,0.9}$	126	135	500 / 20	4/5	3063	136	50 / 100	4/5	2043
Le 450 15c	15	15	200 / 20	5/5	1809	15	50 / 100	5/5	2254
Le 450 25c	25	27	500 / 20	5/5	4002	29	50 / 300	5/5	4918
Queen 15 15	16	17	275 / 20	3/5	845	18	50 / 100	4/5	624
School1 nsh	14	14	200 / 20	5/5	841	14	50 / 100	5/5	1316

Rysunek 2: Zestawienie wyników dla ANTCOL i MMGC. Parametry komputera: Pentium 4, 1.8GHz CPU, 256MB RAM.

z reguły potrzebuje więcej czasu. Należy również zwrócić uwagę, że zwłaszcza dla dużych grafów, MMGC jest daleki od najlepszego znanego rozwiązania. Zmniejszenie parametru ρ w ANTCOL zmniejszyłoby tendencję do utknieć się w lokalnym minimum, ale mogłoby również spowodować przeszukiwanie tylko ścieżek na pozór dobrych.

5 Podsumowanie

Modyfikacja ANTCOL pod postacią MMGC, poprawia przeszukiwanie lokalne poprzez zastosowanie innych metod i strategii, oraz poprawienie funkcji oceny rozwiązania, kosztem wydłużenia czasu potrzebnego na przeprowadzenie całej procedury.

Oba algorytmy dają zadowalające rozwiązania. Mając krótki czas na obliczenia, lepiej wybrać ANTCOL ze względu na krótki czas znalezienia niezłego rozwią-

zania. W przypadku gdy potrzebne jest rozwiązanie bliskie optimum MMGC wypada lepiej.

Literatura

- [1] R. GAREY, D.S. JOHNSON, Computers and intractability: a guide to the theory of NP-completeness, W.H. Freeman and Company, New York, 1979.
- [2] An ACO Algorithm for Graph Coloring Problem E. Salari, K. Eshghi Department of Industrial Engineering Sharif University of Technology, Tehran, Iran
- [3] Mohamed Atef Mosa , Alaa Hamouda , Mahmoud Marei , Graph Coloring and ACO based Summarization for Social Networks, Expert Systems With Applications (2017), doi: 10.1016/j.eswa.2017.01.010
- [4] Mehrotra A. and Trick M., 1996, "A column Generation Approach for Graph Coloring Problem", INFORMS Journal on Computing. 8(4), 344-354.
- [5] Costa D. and Hertz A., 1997, "Ants Can Color Graphs", Journal of the Operational Research Society 48, 295-305.
- [6] Swell E.C., 1996, "An Improved Algorithm for Exact Graph Coloring", in D.S. Johnson and M. Trick editors, DIMACS Series in Computer Mathematics and Theoretical Computer Science AMS, 26, 359-373.
- [7] Caramia C. and Dell'Olmo P., 2002, "Vertex Coloring by Multistage Branch and Bound", in D.S. Johnson, A. Mehrotra, M. Trick editors, Proceedings of the Computational Symposium on Graph Coloring and its Generalization, Ithaca, New York, USA.
- [8] Alexander Katz, Gabe Smith, Jeein Kim Four Color Theorem. Brilliant.org. Retrieved 13:05, June 12, 2020, from <https://brilliant.org/wiki/four-color-theorem/>
- [9] Zerovnik, Janez and Vesel, Aleksander. (2000). How Well Can Ants Color Graphs?. Journal of Computing and Information Technology (citsrce.hr); Vol.8 No.2. 8. 10.2498/cit.2000.02.04.