ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2002
Seria: AUTOMATYKA z. 136 Nr kol. 1556

Robert JANCZEWSKI, Michał MAŁAFIEJSKI
Politechnika Gdańska

T-SL, *T*-LF I *T*-DSATUR - NOWE HEURYSTYKI DLA PROBLEMU PRZYDZIAŁU CZĘSTOTLIWOŚCI

Streszczenie. Problem przydziału częstotliwości to zagadnienie, które formułuje się zazwyczaj następująco: na pewnym obszarze znajduje się grupa nadajników radiowych, którym trzeba przydzielić częstotliwości w taki sposób, żeby nie zakłócały się podczas nadawania i aby szerokość wykorzystanego przez nie pasma częstotliwości była minimalna. Zagadnienie to modeluje się zazwyczaj na gruncie teorii grafów za pomocą trzech pojęć: grafów interferencji, kontrastowych pokolorowań i *T*-rozpiętości. Niniejszy artykuł zawiera opis tego modelu, podstawowe informacje o jego złożoności obliczeniowej oraz opis trzech nowych heurystyk — algorytmów, które są bardzo efektywne, ale generują przybliżone rozwiązania problemu przydziału częstotliwości. Algorytmy te zostały zbadane zarówno metodami teoretycznymi - wskazano, dla jakich klas grafów interferencji zachowują się dobrze, a dla jakich źle, jak i doświadczalnymi - przytoczono wyniki testów, jakim zostały poddane na małych i średnich losowych grafach interferencji..

T-SL, *T*-LF AND *T*-DSATUR - NEW HEURISTIC ALGORITHMS FOR THE FREQUENCY ASSIGNMENT PROBLEM

Summary. Frequency assignment problem (FAP) can be formulated as follows: there is a group of transmitters situated in a certain region of a plane; a channel is to be assigned to each of them in such a way that there is no interference during transmitting and the span of used frequency band is minimal. The paper is devoted to the graph-theoretical model of FAP based on three notions: interference graphs, *T*-colorings and the *T*-span. We describe the model, provide basic information about its computational complexity and present three new heuristic approximate algorithms. Results of the theoretical analysis and computer tests of these algorithms are also included.

1. Wprowadzenie

Tematem niniejszego referatu jest wprowadzony przez Hale'a [1] teoriografowy model dla problemu przydziału częstotliwości (problem przydziału częstotliwości formułuje się

zazwyczaj następująco: na pewnym obszarze znajduje się grupa nadajników radiowych, którym trzeba przydzielić częstotliwości w taki sposób, żeby nie zakłócały się podczas nadawania i aby szerokość wykorzystanego przez nie pasma częstotliwości była jak najmniejsza). Model ten opiera się na trzech pojęciach: grafach interferencji, kontrastowych pokolorowaniach i T-rozpiętości.

Graf interferencji to graf prosty, dla którego zbiorem wierzchołków jest zbiór rozważanych nadajników, a zbiorem krawędzi zbiór par nieuporządkowanych postaci $\{u, v\}$, gdzie u, v to para nadajników, które mogą się zakłócać podczas nadawania. *Kontrastowym pokolorowaniem* lub *T-pokolorowaniem* grafu G nazywamy każdą funkcję, której dziedziną jest zbiór wierzchołków grafu G, przeciwdziedziną zbiór liczb całkowitych, a która spełnia następujący warunek: dla każdej pary sąsiadujących w grafie G wierzchołków u, v odległość przydzielonych im *kolorów* (liczb całkowitych) nie należy do zbioru T (T jest tutaj ustalonym, skończonym podzbiorem zbioru liczb całkowitych nieujemnych). *T-rozpiętością* grafu G nazywamy najmniejszą możliwą z *rozpiętości* (różnicy pomiędzy największym, a najmniejszym wykorzystanym kolorem) jego T-pokolorowań.

Po utożsamieniu rzeczywistych częstotliwości z liczbami całkowitymi i odpowiednim dobraniu zbioru T nietrudno będzie zauważyć, że w modelu Hale'a odpowiednikiem rozwiązania problemu przydziału częstotliwości będą te kontrastowe pokolorowania grafu interferencji, których rozpiętość będzie równa T-rozpiętości tego grafu. Model Hale'a ma jednak pewną wadę - wyznaczanie takich T-pokolorowań jest, nawet po narzuceniu wielu dodatkowych ograniczeń, NP-trudne [2, 3]. Z praktycznego punktu widzenia oznacza to, że nie potrafimy ich wyznaczać w czasie wielomianowym. Powstaje naturalne w tej sytuacji pytanie - co zrobić, gdy z jakichś przyczyn musimy w efektywny sposób generować takie T-pokolorowania. Po chwili namysłu odpowiedź narzuca się sama - trzeba zrezygnować z tego, co nieosiągalne i zadowolić się algorytmami, które w efektywny sposób generują T-pokolorowania, ale nie gwarantują tego, że ich rozpiętość będzie równa T-rozpiętości.

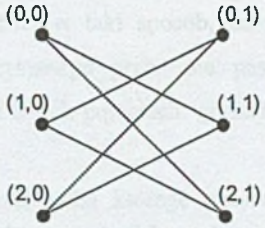
Trzy z takich algorytmów - T-SL, T-LF i T-DSATUR - zostaną w dalszej części niniejszego referacie opisane. Zbadana zostanie ich złożoność obliczeniowa oraz tzw. *dobroć*, czyli funkcja, która przyporządkowuje liczbie naturalnej n maksymalny z ilorazów w postaci rozpiętości pokolorowania wygenerowanego przez badany algorytm przez T-rozpiętość kolorowanego grafu dla wszystkich niepustych n-wierzchołkowych grafów. Wskazane zostaną przykłady grafów dobrze i źle kolorowanych przez rozważane algorytmy. Na samym

końcu przytoczymy wyniki testów komputerowych, jakim zostały poddane, oraz wynikające z nich wnioski.

2. *T*-LF

Algorytm *T*-LF, będący adaptacją znanego od ponad trzydziestu lat algorytmu LF, działa następująco. Na samym początku - w zerowym kroku - wybiera jeden z wierzchołków o maksymalnym stopniu, powiedzmy v_1, i koloruje go zerowym kolorem, tzn. przyjmuje, że $c(v_1) = 0$. Następnie - w i-tym kroku - mając już pokolorowane wierzchołki $v_1, v_2, ..., v_i$, wybiera spośród pozostałych ten wierzchołek, powiedzmy v_{i+1}, którego stopień jest największy i koloruje go zachłannie, tzn. przyjmuje, że $c(v_{i+1}) = \min\{ k \geq 0$: dla każdego $j = 1, 2, ..., i$, jeżeli v_{i+1} sąsiaduje z v_j, to $|c(v_j) - k| \notin T \}$. Złożoność algorytmu *T*-LF jest równa $O(n^2|T|)$, co można łatwo wykazać.

Dobroć algorytmu *T*-LF jest niestety liniowa, a więc najgorsza z możliwych (jak wykazano w [3], dobroć każdego algorytmu zachłannego jest co najwyżej liniowa). Aby to wykazać, wystarczy skonstruować ciąg grafów o stałej *T*-rozpiętości i rosnącej liczbie wierzchołków, które są źle kolorowane przez algorytm *T*-LF. Źle, tzn. w taki sposób, że rozpiętości uzyskanych *T*-pokolorowań rosną co najmniej tak szybko, jak liczba wierzchołków pomnożona przez pewną stałą. Te warunki spełnia m.in. następująco zdefiniowany ciąg grafów Johnsona: J_n jest grafem, którego zbiorem wierzchołków jest $\{0, 1, 2, ..., n\} \times \{0, 1\}$, a zbiorem krawędzi $\{\{(i, 0), (j, 1)\}: i \neq j\}$. Każdy z grafów J_n jest dwudzielny i niepusty, więc jego *T*-rozpiętość jest równa najmniejszej liczbie naturalnej nie należącej do zbioru T [3]. Co więcej, graf J_n jest n-regularny, więc algorytm *T*-LF może kolorować jego wierzchołki w zupełnie dowolnej kolejności. Nietrudno jest zauważyć, że jeżeli zostaną one pokolorowane w następującej kolejności: (0, 0), (0, 1), (1, 0), (1, 1), ..., (n, 0), (n, 1), to uzyskane *T*-pokolorowanie będzie miało rozpiętość równą rozpiętości, jaką ma *T*-pokolorowanie grafu K_{n+1} wygenerowane przez algorytm zachłanny, a zatem będzie miało rozpiętość co najmniej n razy większą od *T*-rozpiętości, co było do wykazania. Przykładowy graf J_2 znajduje się na rys.1.



Rys. 1. Graf J_2

Fig. 1. The graph J_2

To, czy pokolorowanie utworzone przez omawiany algorytm będzie optymalne, zależy w dużym stopniu zarówno od kolorowanego grafu, jak i od postaci zbioru odległości zakazanych. I tak jedynymi grafami, o których wiadomo, że niezależnie od postaci zbioru odległości zakazanych zostaną przez algorytm T-LF optymalnie pokolorowane, są grafy, których składowe spójności są gwiazdami. Co więcej, można łatwo wykazać, że nie ma zbioru odległości zakazanych, dla którego każdy graf zostanie optymalnie pokolorowany przez T-LF. Wiadomo także, że najmniejszym grafem, który jest źle kolorowany przez omawiany algorytm, jest K_3, gdyż jeżeli $T = \{0, 1, 4, 5\}$, to $sp_T(K_3) = 6$, a T- LF wygeneruje T-pokolorowanie o rozpiętości równej 8.

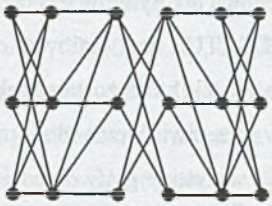
Dodajmy jeszcze, że w przeprowadzonych testach algorytm T-LF wypadł nieźle, bo był niewiele gorszy od najlepszego z testowanych algorytmów.

3. T-SL

Algorytm T-SL, będący adaptacją znanego od prawie trzydziestu lat algorytmu SL, działa następująco. Na samym początku - w pierwszym etapie - ustawia wierzchołki kolorowanego grafu w takiej kolejności $v_1, v_2, ..., v_n$, że dla każdego i stopień wierzchołka v_i jest najmniejszy w podgrafie grafu G indukowanym przez zbiór $\{v_1, v_2, ..., v_i\}$. Następnie – w drugim etapie - koloruje je według tej samej reguły, co T-LF, czyli zachłannie. Złożoność algorytmu T-SL jest równa $O(n^2|T|)$, co można łatwo wykazać.

Dobroć algorytmu T-SL jest liniowa, co można wykazać stosując tę samą metodę, która została wykorzystana do oszacowania dobroci algorytmu T-LF. Wystarczy tym razem wiedzieć, że istnieje taki ciąg grafów S_n (zobacz np. [5]), że: $n(S_n) = 6n+6$; graf S_n jest

dwudzielny i niepusty; *T*-SL kolorując graf S_n wygeneruje *T*-pokolorowanie o rozpiętości większej lub równej rozpiętości *T*-pokolorowania wygenerowanego przez *T*-LF na grafie J_n. Przykładowy graf S_2 znajduje się na rys. 2..



Rys. 2. Graf S_2

Fig. 2. The graph S_2

Podobnie jak to było w przypadku algorytmu *T*-LF, optymalność pokolorowania utworzonego przez algorytm *T*-SL zależy w dużym stopniu zarówno od kolorowanego grafu, jak i od postaci zbioru odległości zakazanych. Jedynymi grafami, o których wiadomo, że niezależnie od postaci zbioru odległości zakazanych zostaną przez algorytm *T*-SL optymalnie pokolorowane, są lasy. Co więcej, można łatwo wykazać, że nie ma zbioru odległości zakazanych, dla którego każdy graf zostanie optymalnie pokolorowany przez *T*-SL. Wiadomo także, że najmniejszym grafem, który jest źle kolorowany przez omawiany algorytm, jest K_3, gdyż jeżeli $T = \{0, 1, 4, 5\}$, to $sp_T(K_3) = 6$, a *T*-SL wygeneruje *T*-pokolorowanie o rozpiętości równej 8.

Dodajmy jeszcze, że w przeprowadzonych testach algorytm *T*-SL wypadł słabo, bo był gorszy od pozostałych algorytmów.

4. *T*-DSATUR

Algorytm *T*-DSATUR, będący adaptacją jednego z najlepszych znanych wielomianowych algorytmów kolorowania grafów, działa następująco. Na samym początku - w zerowym kroku - wybiera jeden z wierzchołków o maksymalnym stopniu, powiedzmy v_1, i koloruje go kolorem 0, tzn. przyjmuje, że $c(v_1) = 0$. Następnie - w *i*-tym kroku - mając już pokolorowane wierzchołki $v_1, v_2, ..., v_i$, wybiera spośród pozostałych ten wierzchołek, powiedzmy v_{i+1}, którego *stopień T-nasycenia*, tj. liczba kolorów, których nie można użyć do

jego pokolorowania, gdyż spowodowałoby to powstanie nielegalnego T-pokolorowania, jest największy. Jeżeli takich wierzchołków jest wiele - więcej niż jeden - to spośród nich wybiera wierzchołek o maksymalnym zwykłym stopniu. Następnie koloruje go tak, jak poprzednie algorytmy, czyli zachłannie. Złożoność algorytmu T-DSATUR jest równa $O(n^2(\max T+1))$, a więc w odróżnieniu od poprzednich algorytmów jest pseudowielomianowa.

Dobroć algorytmu T-DSATUR jest liniowa, czego dowodzić nie będziemy, gdyż metoda dowodu jest podobna do tej, która została wykorzystana w przypadku algorytmu T-SL. Odnotujmy tylko, że w przeciwieństwie do omówionych wcześniej algorytmów T-DSATUR optymalnie koloruje wszystkie grafy dwudzielne i to niezależnie od postaci zbioru odległości zakazanych! Wiadomo także, że najmniejszym grafem źle przez niego kolorowanym jest K_3, gdyż jeżeli $T = \{0, 1, 4, 5\}$, to $sp_T(K_3) = 6$, a T-DSATUR wygeneruje T-pokolorowanie o rozpiętości równej 8.

T-DSATUR jest najlepszym, choć najwolniejszym spośród zbadanych przez nas algorytmów - pokolorowania, jakie tworzy, korzystają z mniejszej liczby kolorów i mają mniejszą rozpiętość od pokolorowań generowanych przez pozostałe algorytmy.

5. Testy

Testy, jakim poddano omówione w niniejszym referacie algorytmy, miały na celu ocenę ich jakości oraz ustalenie, który z nich jest najlepszy. Przeprowadzono dwa testy, w ramach których badano jedno kryterium jakości algorytmów kontrastowego kolorowania grafów: średnią rozpiętość pokolorowania. Uzyskane w ten sposób wyniki zostały zebrane w znajdujących się dalej tabelkach.

W ramach pierwszego testu porównano badane algorytmy z algorytmem optymalnym. Taki test można wykonać tylko na relatywnie małych zestawach danych - algorytm optymalny ma przecież wykładniczą złożoność obliczeniową, - więc na dane, które zostały w tym teście wykorzystane, składały się wszystkie parami nieizomorficzne grafy o $n \leq 8$ wierzchołkach i wszystkie zbiory odległości zakazanych T, spełniające nierówność $\max T \leq 6$. Uzyskane przez algorytm optymalny wyniki zostały zebrane w tab. 1-4; liczba znajdująca się w n-tym polu drugiego wiersza to rozpiętość optymalnego T-pokolorowania grafu G, uśredniona po wszystkich grafach G o n wierzchołkach i wszystkich zbiorach odległości zakazanych T o $\max T \leq 6$ oraz zaokrąglona do drugiego miejsca po przecinku.

Porównując wyniki badanych algorytmów z tab.1, nietrudno jest zauważyć, że jedynie *T*-SL wygenerował pokolorowania o rozpiętości w zauważalny sposób odbiegającej od optymalnej.

W ramach drugiego testu porównano badane algorytmy ze sobą. Taki test można wykonać na danych o większych rozmiarach, więc na dane wykorzystane w tym teście składały się losowo generowane grafy o 10, 20, 30, 40, 50, 60, 70, 80, 90 i 100 wierzchołkach oraz takie same zbiory odległości zakazanych, jak w poprzednim teście. Przygotowano 90 serii grafów; seria oznaczona jako (n, p) składała się z tysiąca grafów losowych o ustalonej liczbie n wierzchołków i ustalonym prawdopodobieństwie p wylosowania krawędzi pomiędzy wierzchołkami. Co zaskakujące, każda seria danych dała ten sam rezultat - algorytm *T*-DSATUR generuje pokolorowania o najmniejszej rozpiętości, następny jest *T*-LF, a na samym końcu *T*-SL. Dokładniejsza analiza tych danych pozwala na wyciągnięcie następujących wniosków:

1. Różnica pomiędzy najlepszym algorytmem a następnym jest w ogólności niezbyt duża; *T*-DSATUR jest znacznie lepszy niż *T*-LF tylko na grafach rzadkich.
2. Chociaż różnice pomiędzy badanymi algorytmami nie są duże, to wraz ze wzrostem liczby wierzchołków kolorowanych grafów powinny się powiększać i to zarówno co do bezwzględnej wartości, jak i w stosunku do rozpiętości generowanych przez najlepszy z badanych algorytmów.

Biorąc pod uwagę wyniki obu testów można stwierdzić, że *T*-DSATUR jest najlepszym spośród badanych algorytmów; *T*-LF jest od niego niewiele gorszy, szczególnie na gęstych grafach. Najsłabszym algorytmem jest *T*-SL.

Tabela 1

Algorytm optymalny — średnia rozpiętość

N	1	2	3	4	5	6	7	8	9
	0.00	0.99	2.37	3.35	4.63	5.55	6.49	7.23	

Tabela 2

Algorytm *T*-LF — średnia liczba kolorów

N	1	2	3	4	5	6	7	8	9
	0.00	0.99	2.38	3.37	4.68	5.60	6.56	7.31	8.03

Tabela 3

Algorytm *T*-SL — średnia liczba kolorów

N	1	2	3	4	5	6	7	8	9
	0.00	0.99	2.38	3.37	4.68	5.60	6.56	7.31	8.03

Tabela 4

Algorytm *T*-DSATUR — średnia rozpiętość

N	1	2	3	4	5	6	7	8	9
	0.00	0.99	2.38	3.37	4.68	5.60	6.56	7.31	8.03

Tabela 5

Algorytm *T*-LF — średnia rozpiętość

n/p	10	20	30	40	50	60	70	80	90
10	2.46	4.36	5.94	7.41	8.93	10.8	12.7	15.1	19.7
20	5.22	7.85	10.7	13.3	16.2	19.5	22.9	27.7	35.1
30	6.71	11.1	14.5	18.9	22.4	27.2	32.3	39.0	48.7
40	8.47	13.5	18.7	23.3	28.7	34.3	41.5	49.8	61.5
50	9.97	16.0	21.6	28.0	34.0	41.4	49.6	60.0	74.7
60	11.8	18.9	25.3	32.1	39.7	48.0	57.5	70.1	87.6
70	12.8	20.7	28.5	36.4	44.8	54.3	65.5	79.8	100
80	13.9	22.9	31.6	40.4	50.2	60.4	73.0	89.0	113
90	15.0	25.1	34.4	44.4	54.8	66.9	80.5	98.3	125
100	16.3	27.3	37.6	48.2	59.7	72.5	87.8	107	138

Tabela 6

Algorytm *T*-SL — średnia rozpiętość

n/p	10	20	30	40	50	60	70	80	90
10	3.00	5.11	6.75	8.38	9.98	11.9	13.7	16.2	20.1
20	6.18	9.18	12.4	15.1	18.4	21.6	25.6	30.5	37.6
30	8.24	12.9	17.1	21.0	25.5	30.3	35.7	42.9	53.3
40	10.3	15.8	21.1	26.3	31.9	38.1	45.2	54.5	67.9
50	12.3	19.1	24.8	31.3	38.2	45.6	54.5	65.5	82.7
60	13.7	21.3	28.5	35.9	43.6	52.7	62.8	75.9	96.0
70	14.8	23.9	32.0	40.5	49.5	59.4	71.0	86.2	109
80	16.9	26.3	35.4	44.6	54.7	66.0	79.2	96.1	122
90	18.5	28.7	38.9	49.0	59.9	72.4	87.0	106	135
100	19.7	30.9	41.8	52.9	65.0	78.7	95.0	115	147

Tabela 7

Algorytm *T*-DSATUR — średnia rozpiętość

n/p	10	20	30	40	50	60	70	80	90
10	2.37	4.17	5.72	7.16	8.66	10.5	12.5	15.0	19.7
20	4.78	7.15	9.92	12.7	15.2	18.7	22.0	26.9	34.8
30	5.96	9.93	13.6	17.6	21.5	26.1	31.2	37.8	47.8
40	7.47	12.6	17.4	21.9	27.3	32.9	39.9	48.2	60.0
50	8.77	14.5	20.6	26.4	32.6	39.8	47.8	58.1	72.6
60	10.1	17.1	23.6	30.6	38.0	46.0	55.4	67.6	85.2
70	11.8	19.6	26.8	34.4	42.8	52.1	63.0	76.7	97.6
80	12.8	21.1	29.9	38.8	47.8	58.1	70.3	85.7	109
90	13.6	23.2	32.6	42.2	52.6	64.1	77.5	96.1	121
100	14.5	25.4	35.3	45.9	57.3	69.7	84.5	103	133

LITERATURA

1. Hale W.K.: Frequency assignment: Theory and Applications. Proceedings of the IEEE, vol. 68, no. 12, 1980, pp. 1497-1514.
2. Janczewski R.: O złożoności problemu przydziału częstotliwości i kontrastowego kolorowania grafów, Zeszyty Naukowe Polit. Śl., ser. Automatyka z.131, Gliwice 2000, s.. 95-101.
3. Janczewski R.: Kontrastowe kolorowanie grafów i jego zastosowania, Praca doktorska, Politechnika Gdańska, Wydz. ETI, 2001.
4. Janczewski R., Kubale M.: The *T*-DSATUR algorithm — an interesting generalization of the DSATUR algorithm, Advanced Computer Systems V, Szczecin 1998, pp. 288-292.
5. Kubale M.: Introduction to Computational Complexity and Algorithmic Graph Coloring, GTN, Gdańsk 1998.

Recenzent: Prof. dr hab. Jerzy Klamka

Abstract

Frequency assignment problem (FAP) can be formulated as follows: there is a group of transmitters situated in a certain region of a plane; a channel is to be assigned to each of them in such a way that there is no interference during transmitting and the span of used frequency band is minimal. The paper is devoted to the graph-theoretical model of FAP based on three notions: interference graphs, *T*-colorings and the *T*-span. We describe the model, provide basic information about its computational complexity and present three new heuristic approximate algorithms based on well-known LF, SL and DSATUR heuristic algorithms. We study their theoretical properties and make computer tests. In particular, some classes of graphs for which these algorithms behave well and badly are indicated. Moreover, we present and discuss results of the behavior of these algorithms on small random graphs.