BADANIA OPERACYJNE

Zastosowania kolorowania grafów w problemie układania rozkładów zajęć

Maciej Kupczak[1] i Paweł Szołtysek[2]

Politechnika Wrocławska

Streszczenie W niniejszej pracy rozważony zostanie problem układania planu zajęć dla szkół. Jako elementarny problem optymalizacji dyskretnej, zostanie on przedstawiony formalnie, oraz rozwiązany w oparciu o kolorowanie grafów. Zaproponowane zostaną dwa algorytmy rozwiązania (w tym jeden autorski), których skuteczność zostanie przebadana dla różnych danych. Przedstawione również zostanie porównanie ich z istniejącymi już aplikacjami, które pokaże, że zaproponowane przez nas podejście jest konkurencyjnym do owych komercyjnych rozwiązań.

1. Przedstawienie problemu

Problem układania planu zajęć jest problemem dobrze znanym. Zagadnienie to polega na rozdzieleniu zajęć na konkretne godziny, tak by nie występowały konflikty. Przez konflikt rozumiana będzie sytuacja, gdy pewien nauczyciel lub klasa ma przydzielone dwa różne zajęcia w tym samym czasie. Dodatkowo by uczynić uzyskany plan bardziej odpowiednim do sytuacji rzeczywistej wprowadzone zostaną pewne dodatkowe założenia co do jego postaci:

– jeden nauczyciel może prowadzić wiele przedmiotów;
– plan powinien zawierać jak najmniej okienek, tj. niezagospodarowanych godzin lekcyjnych.

2. Formalny opis problemu

Przedstawiony problem można w prosty sposób sprowadzić do problemu kolorowania krawędziowego grafów. Zdefiniowane będą zatem dwie grupy wierzchołków:

– N - zbiór nauczycieli
– n_i - nauczyciel, gdzie $i = 1, \ldots, I$
– K - zbiór klas
– k_j - nauczyciel, gdzie $j = 1, \ldots, J$

Krawędzie natomiast definiowane będą jako $e_{i,j}^{(m)} = \{n_i, k_j, m\}$ - krawędź reprezentująca m-te pojedyncze zajęcia jakie prowadzi nauczyciel n_i z klasą k_j, gdzie $m \in M_{i,j}$ oznacza numer krawędzi łączącej wierzchołki v_i i v_j, a $M_{i,j}$ to ilość wszystkich krawędzi łączących v_i i v_j. Dalej niech $E_{i,j}$ reprezentuje zbiór wszystkich zajęć jakie odbywa nauczyciel n_i z klasą k_j, oraz E oznacza zbiór wszystkich krawędzi.

Jako V oznaczany będzie zbiór wszystkich wierzchołków, stąd $V = N \cup K$. W ten sposób skonstruowany został graf $G(V, E)$ Dalej niech

– $c_{i,j}^{1,(m)}$ - będzie indeksem krawędzi $e_{i,j}^{(m)}$ i oznacza rodzaj przedmiotu jaki reprezentuje krawędź
– $c_{i,j}^{2,(m)}$ będzie kolorem krawędzi $e_{i,j}^{(m)}$ i reprezentuje godzinę w której odbywa się dany przedmiot.

Fakt, iż żadne dwa zajęcia dla jednego nauczyciela nie mogą sie odbywać równocześnie zapisać można nastepująco:

$$\forall_{j,k \in \{1,\ldots,I\}} E_{i,j} \neq \emptyset \wedge E_{i,k} \neq \emptyset \Rightarrow \forall_{m1 \in \{1,\ldots,M_{i,j}\}} \forall_{m2 \in \{1,\ldots,M_{i,k}\}} c_{i,j}^{2,(m1)} \neq c_{i,k}^{2,(m2)} \tag{1}$$

gdzie $i = 1, \ldots, I$.

Podobnie fakt, iż żadna klasa nie powinna mieć równocześnie dwóch lekcji zapisać można w sposób następujący:

$$\forall_{i,k \in \{1,\ldots,I\}} E_{i,j} \neq \emptyset \wedge E_{k,j} \neq \emptyset \Rightarrow \forall_{m1 \in \{1,\ldots,M_{i,j}\}} \forall_{m2 \in \{1,\ldots,M_{k,j}\}} c_{i,j}^{2,(m1)} \neq c_{k,j}^{2,(m2)} \tag{2}$$

gdzie $j = 1, \ldots, J$.

Ostatecznie jako S oznaczane będzie pewne pokolorowanie grafu G. Do oceny rozwiązań posłuży następująca funkcja celu:

$$F(S) = \sum_{v \in V} \left(\max_{j \in V, m=1,\ldots,M_{v,j}} \{c_{v,j}^{m,2}\} - \min_{j \in V, m=1,\ldots,M_{v,j}} \{c_{v,j}^{m,2}\} + 1 - \sum_{j \in V} (M_{v,j}) \right) \tag{3}$$

Ostatecznie problem polega na tym, by przy danych:

- G
- N
- K
- E
- $\forall_{i,j,m}\ c_{i,j}^{1,(m)}$

Wyznaczyć takie pokolorowanie S grafu G, by minimalizować funkcję celu 3

$$S* \rightarrow min_S F(S)$$

przy ograniczeniach 1 i 2

3. Algorytmy rozwiązania

W niniejszym paragrafie przedstawiono dwa algorytmy rozwiązania. Pierwszy można znaleźć w [2], drugi natomiast został skonstruowany przez autorów w oparciu o algorytm zwartego kolorowania drzewa znajdujący się w [2].

3.1. Algorytm NTL

Algorytm ten jest algorytmem heurystycznym. Polega on na kolejnym kolorowaniu krawędzi oraz w razie konieczności przekolorowania już pokolorowanych. Wynikowe pokolorowanie jest oczywiście rozwiązaniem przybliżone. Algorytm pozwala na uzyskanie wyniku w czasie wielomianowym. Jego złożoność można oszacować jako $O(n^4)$.

W celu przedstawienia algorytmu należy najpierw zdefiniować kilka pojęć.

Definicja 1. *Kolorem brakującym dla wierzchołka $v \in V$ grafu $G = (V, E)$, nazywamy kolor, który nie został przypisany żadnej krawędzi incydentnej do v. Oznaczmy przez $C(v)$ zbiór wszystkich brakujących kolorów wierzchołka v.*

Definicja 2. *Dla każdego wierzchołka v ustalamy pewien jego kolor brakujący $c(v)$. Wachlarzem F przy wierzchołku v rozpoczynającym się krawędzią $\{v, w_0\}$ nazywamy taki ciąg krawędzi $\{v, w_0\}$, $\{v, w_1\}$, ..., $\{v, w_s\}$, że $\{v, w_i\}$ ma przydzielony kolor $c(w_i - 1)$, $i > 0$. Liczba s to rozpiętość wachlarza.*

Algorytm NTL opiera się na następującej własności grafów:

Twierdzenie 1. *Przypuśćmy, że wszystkie krawędzie grafu G z wyjątkiem v zostały pokolorowane $\Delta(G) + 1$ kolorami. Oczywiste jest, że zarówno u jak i v mają przynajmniej dwa kolory brakujące. Jeżeli F jest maksymalnym wachlarzem przy v, to albo kolor $c(w_s)$ jest brakujący również w wierzchołku v, czyli $c(w_s) \in C(v)$, albo pewna krawędź w F jest pokolorowana kolorem $c(w_s)$.*

W celu poprawienia pokolorowania algorytm wykorzystuje procedurę *Recolor*, która przedstawia się następująco:

Procedura Recolor

- Kolorować będziemy krawędź $\{u, v\}$
- Wyznaczamy maksymalny wachlarz F(taki, ze $w_0 = u$) przy wierzchołku v.

- Nastepnie rozważyć należy dwa przypadki
  1. Jeżeli $m(w_i) \in M(v)$, gdzie s to rozpiętość wachlarza
     - wówczas kolorujemy krawędź $\{v, w_i\}$ barwą $m(w_i)$ dla każdego $i = 1, ..., s$.
  2. Jeżeli $m(w_i) \notin M(v)$
     - niech P będzie ścieżką w grafie G zaczynającą się w w_s złożoną z krawędzi pokolorowanych barwami $c(v)$ i $c(w_s)$
     - Jeżeli P nie osiąga wierzchołka v, to zamieniamy przeciwne kolory użyte do pomalowania P i przesuwamy cyklicznie kolory poszczególnych krawędzi $\{v, w_i\}$, malując je kolorami $c(w_i)$ dla każdego $0 \leqslant i \leqslant s$, a następnie malujemy $\{u, w_s\}$ kolorem $c(v)$.
     - Jeżeli natomiast P osiąga v, to niech w_j, gdzie $0 \leqslant j \leqslant s - 2$, będzie wierzchołkiem spełniającym równość $c(w_{j-1}) = c(w_s)$. Dalej niech P będzie drogą zaczynającą sie w w_j i pomalowaną kolorami $m(v)$ i $m(w_s)$. W tym przypadku przesuwamy cyklicznie kolory wachlarza, malując każdą krawędź u, w_i kolorem $c(w_i)$ dla $0 \leqslant i \leqslant j - 2$, następnie zmieniamy kolory drogi P i w końcu malujemy krawędź $\{v, E_{w-j}\}$ kolorem $c(v)$.

Właściwy algorytm w pseudokodzie przedstawia się następująco:

```
algorytm KolorujNTL( G )
begin
if Delta(G) <= 2 then
   koloruj G zachłannie trawersując ścieżki i cykle;
else begin
   q := Delta(G) + 1;
   G' := (V, pusty);
   for każda e w E do begin
      G' := G' + e;
      if e = {u,v} nie mogę otrzymać wspólnego koloru brakującego
                   w u i v then
         Recolor(u, v);
         koloruj e;
      end
   end
end
```

3.2. Algorytm zwartego kolorowania grafu

Algorytm przedstawiony poniżej jest autorską interpretacją i rozszerzeniem algorytmu kolorowania zwartego drzew przedstawionego w [?]. Różnica polega na zastosowaniu omówionego tam podejścia do zwartego kolorowania struktur drzewowych z cyklami. Ponadto poniższy algorytm akceptuje i rozwiązuje też problem dla wielokrotnych wystąpień połączeń między wierzchołkami.

1. Utwórz tymczasowy graf $G'(V', E') = G(V, E)$.
2. Jeśli jest cokolwiek w kolejce, dodaj to do G', chyba, że było wrzucone na stos w ostatniej iteracji.
3. Usuń z grafu G' wolne krawędzie.
4. Jeśli E' jest pusty, to sprowadziliśmy zadanie do kolorowania drzewa. Wystarczy pokolorować kolejne ścieżki w dowolnej kolejności przy zachowaniu odpowiednich ograniczeń nałożonych przez zwartość.
5. Jeśli E' jest niepusty, to w grafie są cykle. Znajdź cykl.

6. Dokonaj analizy już pokolorowanych krawędzi i zaproponuj zwarte pokolorowanie do obecnie już istniejącego.
   Jeśli ograniczenie jest na conajwyżej jednym wierzchołku, koloruj z uwzględnieniem istniejącego koloru tego wierzchołka.
   Jeśli ograniczenie jest na większej ilości wierzchołków, podziel cykl na ścieżki pomiędzy nimi i koloruj je zwarcie z istniejącym pokolorowaniem. Jeśli jest to niemożliwe, przekaż ścieżkę do kolejki.
7. Usuń z grafu G' krawędzie które zostały pokolorowane (do innej struktury, wynikowej) lub przekazane do kolejki.
8. Wróć do kroku 2.

Jeśli przez kilka kolejnych iteracji nie będzie dokonywane przyporządkowanie, oznaczać to będzie, że nie znaleziono rozwiązania optymalnego - co jednak nie oznacza, że takiego rozwiązania nie ma. Jest to bowiem algorytm heurystyczny.
Znalezione rozwiązanie można polepszać zmieniając kolejność kolorowania grafów, lub, jeśli jest to akceptowalne, dodać pozostałe krawędzie kolorując je optymalnie, ale nie zwarcie.

4. Badanie jakości przedstawionych algorytmów

4.1. Implementacja i plan badań

Oba przedstawione algorytmy zostały zaimplementowane w języku C++. Badania zostały przeprowadzone dla danych uzyskanych z istniejących szkół. Sprawdzona została skuteczność zaproponowanych algorytmów, czyli wartość funkcji celu, jak i szybkość uzyskania wyniku. Zaproponowane rozwiązania zostały również porównane z komercyjnym programem do układania planów zajęć *Astar05*. Program ten w wersji udostępnianej bezpładnie umożliwia ułożenie i modyfikowanie planu zajęć, nie możliwe jest jednak zapisanie wyników, jednak taka funkcjonalność była wystarczająca do dokonania porównania
Dwa przykładowe plany zajęć przedstawiono poniżej.

4.2. Plan dla średniej wielkości szkoły

Dla tej szkoły program roczny zajęć przedstawia się w sposób następujący (dane rzeczywiste)

nauczyciel klasa	1La	1Lb	1TD	1TLa	1TLb	1TLc	2Da	2Db	2La	2Lb	2Lc	2Ld	2Le	2Lf	1PSa
SG	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0
DB	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0
DG	2	2	0	0	0	0	2	2	0	0	0	0	0	0	0
JP	0	0	0	0	0	0	3	0	3	0	0	0	0	0	0
JG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pa	0	0	3	0	0	0	5	0	0	0	0	5	0	0	0
MP	6	0	0	0	3	0	0	5	0	0	0	0	0	0	0
Sm	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0
BŚ	0	0	0	3	5	0	0	0	6	0	0	0	5	0	0
AD	0	0	0	0	0	4	0	0	0	5	0	0	0	5	0
Go	0	0	0	0	0	0	0	7	0	0	2	2	0	0	2
Mo	3	0	0	0	0	1	4	0	0	0	0	0	0	3	2
AŁ	3	0	0	0	0	1	0	0	2	2	7	0	0	0	0
SB	0	0	3	2	0	0	0	0	0	0	0	2	2	2	0
PW	0	0	0	0	1	4	0	2	0	3	2	3	0	2	0
ZE	2	0	0	0	0	0	7	0	0	0	0	0	0	0	0
JB	0	5	2	2	2	0	0	2	0	2	0	0	0	0	4
RB	2	0	2	2	2	0	0	0	2	0	0	3	3	0	0
SA	0	0	0	0	1	0	0	0	3	3	0	0	0	0	0
ZS	0	0	1	0	0	0	0	0	2	0	2	2	2	2	0
MB	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0
KM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MS	0	0	0	1	1	1	0	0	0	3	0	0	0	0	0
SK	0	4	4	0	0	4	4	0	0	0	0	0	0	0	0
Bu	0	0	0	4	0	0	0	0	0	0	0	0	0	3	0
PS	4	0	0	0	4	0	0	3	0	3	3	3	0	0	0
JŻ	0	0	0	0	0	0	0	0	3	0	0	0	3	0	0
BB	2	2	1	1	1	1	2	2	1	1	1	2	1	1	0
WL	0	0	0	0	0	0	0	0	3	3	4	4	3	3	0
LS	0	0	4	2	2	2	2	2	1	2	2	2	1	1	0
DS.	0	0	0	0	0	0	2	2	1	1	0	0	0	0	0
ME	3	3	2	2	2	2	0	0	0	0	1	1	2	1	0
AM	0	2	2	2	2	2	0	0	2	0	1	1	1	2	0
BL	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
ZJ	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
BD	0	0	3	0	3	0	0	3	0	0	0	0	3	0	3
TS	0	3	3	0	3	0	3	3	3	0	0	3	0	3	3
ML	3	0	0	0	0	3	0	0	0	6	3	3	0	0	0
BK	0	0	0	6	1	3	0	0	0	0	0	0	0	0	3
DK	0	0	1	0	0	0	1	1	1	1	1	1	1	1	0
EG	0	0	7	0	0	0	2	2	0	0	0	0	0	0	0
AA	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
SM	0	0	0	3	0	4	0	0	0	0	0	0	0	0	0
DO	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0
ES	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EM	0	0	0	6	7	0	0	0	0	0	0	0	0	0	0
Wi	0	0	0	4	4	3	0	0	0	0	0	0	0	0	0
MD	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
JM	7	7	0	0	0	0	0	0	3	0	0	0	0	0	0
ZB	0	0	0	0	0	0	0	0	0	3	3	3	3	3	0
GA	0	0	5	0	0	0	4	4	0	0	0	0	0	0	0
MM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Su	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Du	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
PN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Si	0	0	0	2	0	0	2	2	2	2	2	2	2	2	0
Ja	2	2	2	0	2	2	0	0	0	0	0	0	0	0	0
KK	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Algorytm NTL Algorytm zwrócił rozwiązanie, dla którego wartość funkcji celu miała wartość $F(S) = 58$. Poniżej przedstawiony plan uzyskany dla jednej z klas.

	poniedziałek	wtorek	środa	czwartek	piątek
8:00- 8:45	ME	PW	SB	ZS	BU
8:50- 9:35	AD	PW	LS	MO	BU
9:40-10:25	AD	MB	WL	MO	BU
10:40-11:25	AD	AD	SI	MO	ZB
11:30-12:15	AD	ZB	SI	BB	ZB
12:20-13:05	AM	DK	PW	TS	WL
13:10-13:55	AM	SB	PW	TS	WL
14:00-14:45	ZJ	-	-	TS	-

Algorytm zwartego kolorowania grafów Algorytm zwrócił rozwiązanie, dla którego wartość funkcji celu miała wartość $F(S) = 40$. Wartość ta prawdopodobnie byłaby jeszcze mniejsza, gdyby implementacja algorytmu była bardziej dostosowana do konkretnego przypadku. Poniżej przedstawiony plan uzyskany dla jednej z klas.

	poniedziałek	wtorek	środa	czwartek	piątek
8:00- 8:45	PW	BB	WL	ME	Mo
8:50- 9:35	WL	AD	AM	SB	ZB
9:40-10:25	LS	AD	MB	SI	ZB
10:40-11:25	Mo	Bu	Mo	SI	AD
11:30-12:15	AD	PW	AM	ZB	AD
12:20-13:05	TS	Bu	ZJ	PW	TS
13:10-13:55	TS	Bu	ZS	PW	SB
14:00-14:45	WL	DK	-	TS	-

Program Astar05 W przypadku program zwrócił plan dla którego wartość funkcji celu wynosiła $F(s) = 46$. Należy zauważyć, że wszystkie okienka pojawiły się po stronie nauczycieli, natomiast żadna klasa nie miała ani jednego okienka.

4.3. Plan dla małej szkoły

Tym razem przedstawiony zostanie program dla niewielkiej szkoły. Program ten przedstawia się w sposób następujący

nauczyciel klasa	1a	1b	2a	2b	3a	3b
PO	5	0	0	4	0	0
PP	0	5	4	0	5	5
MA	7	1	0	5	0	6
MB	0	5	5	0	6	0
HI	1	1	2	2	0	0
GE	1	1	2	1	0	1
AN	4	2	0	2	0	0
AO	0	2	4	2	4	4
RU	4	0	4	0	4	0
NI	0	4	0	5	0	4
WF	3	3	3	3	4	3
IN	0	1	2	2	1	1
RO	2	2	0	0	0	0
RE	1	1	1	1	1	1
PP	0	0	1	1	0	0

Algorytm NTL W przypadku tej szkoły algorytm NTL zwraca rozwiązanie dla którego wartość funkcji celu wynosi $F(S) = 3$. A oto plan zajęć jaki uzyskała klasa 1a

	poniedziałek	wtorek	środa	czwartek	piątek
8:00- 8:45	-	-	-	-	-
8:50- 9:35	RE	WF	AN	MA	MA
9:40-10:25	RO	RU	AN	MA	PO
10:40-11:25	RO	RU	AN	MA	PO
11:30-12:15	WF	RU	HI	MA	PO
12:20-13:05	WF	RU	GE	MA	PO
13:10-13:55	-	AN	MA	-	PO
14:00-14:45	-	-	-	-	-

Natomiast tak prezentuje się plan dla nauczyciela IN

	poniedziałek	wtorek	środa	czwartek	piątek
8:00- 8:45	-	-	-	-	-
8:50- 9:35	1b	2b	-	-	-
9:40-10:25	2a	-	-	-	-
10:40-11:25	2a	-	3b	-	-
11:30-12:15	2b	-	-	-	-
12:20-13:05	3c	-	-	-	-
13:10-13:55	-	-	-	-	-
14:00-14:45	-	-	-	-	-

Algorytm zwartego kolorowania grafów Dla tego przypadku algorytm zwartego kolorowania grafów znajduje takie ułożenie planu, dla którego $F(S) = 0$.
Klasa 1a ma poniższy plan:

	poniedziałek	wtorek	środa	czwartek	piątek
8:00- 8:45	MA	MA	WF	WF	-
8:50- 9:35	PO	GE	RU	WF	-
9:40-10:25	PO	HI	RU	RU	-
10:40-11:25	MA	MA	AN	RU	-
11:30-12:15	MA	AN	AN	PO	-
12:20-13:05	PO	MA	PO	PO	-
13:10-13:55	PO	AN	-	-	-
14:00-14:45	MA	RE	-	-	-

Widać na nim niedoskonałość algorytmu w obecnej implementacji - pierwszego dnia klasa 1a ma zajęcia tylko z nauczycielami MA i PO, ale w sumie jest to 8 godzin. Jest to spowodowane specyficznym ułożeniem klasy 1a, nauczycieli MA i PO oraz cykli w całym grafie. Inna klasa (2a) przedstawia się następująco:

	poniedziałek	wtorek	środa	czwartek	piątek
8:00- 8:45	PP	GE	RU	IN	-
8:50- 9:35	MB	HI	WF	PP	-
9:40-10:25	MB	MB	AO	WF	-
10:40-11:25	PP	PP	RU	RE	-
11:30-12:15	PP	MB	AO	-	-
12:20-13:05	MB	HI	RU	-	-
13:10-13:55	GE	RU	AO	-	-
14:00-14:45	AO	WF	IN	-	-

Gdzie zróżnicowanie zajęć jest już odpowiednie, a blokowanie ich zachowane.
Dla nauczyciela IN, plan pracy jest następujący:

	poniedziałek	wtorek	środa	czwartek	piątek
8:00- 8:45	-	-	-	2a	-
8:50- 9:35	-	-	-	2b	-
9:40-10:25	-	-	-	2b	-
10:40-11:25	-	-	-	3b	-
11:30-12:15	-	-	-	3a	-
12:20-13:05	-	-	-	-	-
13:10-13:55	-	-	1b	-	-
14:00-14:45	-	-	2a	-	-

Program Astar05 Dla tej szkoły program Astar zwrócił plan dla którego Funkcja celu przyjmowała wartość $F(S) = 5$.

5. Wnioski

Podsumowując, przedstawione algorytmy poradziły sobie zadowalająco z budową planów zajęć testowanych przypadków. Co ważne, mogą one nawet stanowić, po pewnych modyfikacjach, poważną konkurencję dla istniejących obecnie na rynku rozwiązań komercyjnych.
W większości testowanych programów zajęć, analizując wartość funkcji celu, algorytm oparty na zwartym kolorowaniu był nieco lepszy, zarówno od drugiego algorytmu jak i od rozwiązania komercyjnego. Natomiast algorytm NTL spisywał się nieznacznie gorzej od systemu Astar05. W przypadku zaproponowanych rozwiązań, wadą uzyskiwanych wyników jest fakt, iż blokują one zajęcia w taki sposób, iż większość zajęć z jednego przedmiotu odbywa się po kolei. Jednocześnie, odnosząc się do algorytmu bazującego na kolorowaniu zwartym, należy zauważyć, że możnaby łatwo uzyskać lepsze plany, nieznacznie modyfikując zasadę działania algorytmu, a konkretnie kolejność odczytywania i pracy na cyklach (która jest decydującą operacją w tym procesie).
Trudno jest określić jaki wpływ ma rozmiar problemu na otrzymywane wyniki, praktycznie faktem jest iż w zależności od struktóry połączeń między nauczycielami i klasami (ilością odbywanych wspólnie godzin) otrzymane wyniki dla takiego samego rozmiaru problemu mogą się różnić nawet w sposób znaczny. Da się jednak zaobserwować ogólna tendencja do pogorszenia się otrzymanych wyników wraz ze wzrostem rozmiaru problemu.
Należy pamiętać, iż wygenerowany przez nas plan może podlegać pewnym zmianom. Zawsze w realnym świecie jest tak, że niektórzy nauczyciele nie mogąc zgodzić się na zaproponowany plan negują go, lub też na własną rękę przesuwają zajęcia.
Ostatecznie najlepsze wyniki uzyskano dla algorytmu autorskiego algorytmu kolorowania zwartego. W celu zastosowań praktyznych wymaga on jednak pewnych dalszych modyfikacji, by dostosowć go do rzeczywistych warunków jakie spełniać musi plan.

6. Dalsze możliwości rozwoju

Przedstawiony problem ma wiele perspektyw dalszego rozwoju. Przede wszystkim algorytmy zmodyfikować należy tak by nie przydzielały wszystkich zajęć z danego przedmiotu w ten sam dzień. Problem można rozwijać również dodając dodatkowe wymagania, takie jak np.
– dodanie blokowania zajęć
– dodanie możliwości przypisywania zajęć do sal lekcyjnych
– możliwość dzielenia klas na grupy
Dodanie każdego założenia w pewien sposób wpływa na zastosowane rozwiązania, wymagać one będą w takich wypadkach, albo modyfikacji, albo zaproponowania nowych rozwiązań.

Literatura

1. *www.astar05.pl*
2. Marek Kubale: *Optymalizacja dyskretna: modele i metody kolorowania grafów. Zwarte kolorowanie grafów*, Wydawnictwa Naukowo-Techniczne, Warszawa 2002, 167-179

3. Krzysztof Kamiński : *Metody sztucznej inteligencji w rozwiązywaniu wybranych problemów kolorowania grafów*, Rozprawa doktorska, Wrocław 2006
4. Mrozicki C.: *Zastosowanie metody kolorowania grafów do automatycznego układania harmonogramów zajęć*, Autoreferat pracy doktorskiej, Warszawa 2006