

Bożena REBAJN

UKŁADANIE PLANU ZAJĘĆ ZA POMOCĄ SYSTEMU PLAN

Streszczenie. Problem układania planu zajęć należy do trudnych, klasycznych problemów kombinatorycznych i jest intensywnie badany ze względu na praktyczne zastosowania. W artykule przedstawiono opis algorytmu rozwiązywania tego problemu, jaki zastosowano w systemie wspomagającym układanie planu zajęć PLAN, a także omówiono strukturę i zasady działania systemu.

TIMETABLING IN THE PLAN SYSTEM

Summary. The timetable problem, one of the difficult classical combinatorial problems, has been intensively investigated due to its practical relevance. The paper presents the algorithm implemented in the PLAN timetabling system. The system's structure and its principles of operation are also described.

TRAÇAGE D'HORAIRESCOLAIRESDANS LE SYSTEME PLAN

Résumé. Le traçage d'horaires scolaires est un problème combinatoire difficile classique qui a fixé beaucoup d'attention pour de raisons pratiques. L'article présente l'algorithme de traçage appliqué dans le système logiciel PLAN. La structure et les principes du système sont aussi présentés.

1. Wprowadzenie

Problem układania planu zajęć dla szkół jest złożonym problemem kombinatorycznym. Z problemem tym spotyka się wielu nauczycieli w każdej szkole, od typu podstawo-

wego po wyższy. Ręczne układanie planu zajęć jest bardzo czasochłonne. Powstają więc pomysły wykorzystania do tej pracy komputera.

Rozważany problem można rozwiązać na wiele sposobów. Jednak nie wszystkie rozwiązania są korzystne dla przyszłych użytkowników planu: uczniów, studentów czy pracowników szkoły. W praktyce istnieje potrzeba znalezienia rozwiązania optymalnego, uwzględniającego wszelkie ograniczenia związane z konkretną szkołą, dla której plan powstaje, zapewniającego wszystkim dogodnie warunki pracy i nauki.

Układanie planu zajęć polega na wiązaniu ze sobą w gotowe jednostki planu informacji dotyczących:

- grup słuchaczy,
- nauczycieli, grup prowadzących,
- puli zajęć, które powinny odbyć się w danym semestrze,
- dostępnych pomieszczeń dydaktycznych,
- dopuszczalnych godzin pracy w poszczególnych dniach tygodnia.

Wszystkim zajęciom należy przypisać ściśle określony termin i salę dydaktyczną. Zajęcia muszą być prowadzone przez określonego wcześniej prowadzącego, z danym zespołem słuchaczy i z danego przedmiotu nauczania. Oprócz tych podstawowych założeń prawidłowo ułożony plan zajęć powinien uwzględnić jeszcze wiele dodatkowych ograniczeń i często przeciwstawnych wymagań. Należą do nich między innymi:

- ciągłość zajęć grup słuchaczy i prowadzących, czyli brak luk w planie zajęć,
- równomierne obciążenie dostępnych pomieszczeń dydaktycznych,
- spełnienie indywidualnych preferencji i wymagań dotyczących terminów poszczególnych zajęć zgłaszanych przez prowadzących.

Dodatkowymi czynnikami, które komplikują proces tworzenia planu zajęć, są: niewystarczająca liczba sal dydaktycznych, ściśle określone przeznaczenie niektórych sal, przypisanie terminów do ściśle określonych zajęć, zajęcia prowadzone z kilkoma grupami równocześnie, zajęcia prowadzone przez zespoły prowadzących.

Złożoność problemu sprawia, że powstające systemy wspomagają układanie planu zajęć w szkołach określonego typu, np. w szkołach podstawowych i średnich lub w szkołach wyższych, uwzględniając specyficzne uwarunkowania związane z danym typem szkoły. Przykładem programu wspomagającego planowanie zajęć w szkołach średnich i podstawowych może być produkt firmy Vulcan, a w szkołach wyższych system, jaki powstaje w Politechnice w Mediolanie [5] oraz system Plan [4].

W dalszej części artykułu zaprezentowane zostanie rozwiązanie problemu układania planu zajęć, jakie zastosowano w systemie Plan, który powstał w Instytucie Informatyki Politechniki Śląskiej. Zanim jednak poznamy sam system, przyjrzyjmy się bliżej problemowi.

2. Sformułowanie problemu

Problem układania planu zajęć dla szkół jest problemem NP–zupełnym, znanym w literaturze jako problem TT (ang. TimeTabling). Klasa problemów NP obejmuje te wszystkie problemy, które mogą być rozwiązane za pomocą niedeterministycznych algorytmów o wielomianowym czasie przetwarzania [2]. Algorytmy niedeterministyczne charakteryzują się tym, że w danej chwili muszą dokonać wyboru jednej spośród wielu możliwości i nie stosują do tego celu żadnej deterministycznej procedury. W pracy [3] podana została definicja transformacji wielomianowej, z której wynika, że dla wykazania NP–zupełności badanego problemu decyzyjnego wystarczy przetransformować do niego wielomianowo dowolny znany problem NP–zupełny.

Korzystając z tej właściwości problemów NP–zupełnych, Even, Itai i Shamir [1] udowodnili, że problem układania planu zajęć jest problemem NP–zupełnym. Problem TT został przetransformowany do problemu spełnialności wyrażeń boolowskich. W cytowanej pracy [1] formułując rozważany problem pominięto wiele czynników, które w praktyce odgrywają ważną rolę. Mimo przyjętych ograniczeń udowodniono, że problem nadal jest problemem klasy NP.

Problem układania planu zajęć można sformułować w następujący sposób. Dane są:

- skończony zbiór godzin w tygodniu, H ,
- zbiór godzin, które muszą być przeprowadzone przez każdego z n nauczycieli $\{T_1, T_2, \dots, T_n\}$, gdzie $T_i \subseteq H$; każdy i -ty nauczyciel może prowadzić zajęcia w pewnym podzbiorze T_i zbioru godzin H ,
- zbiór m klas $\{C_1, C_2, \dots, C_m\}$, gdzie $C_j \subseteq H$, z których każda jest dostępna tylko w czasie określonych godzin tygodnia, oznaczonych C_j ,
- macierz R o rozmiarach $n \times m$ nieujemnych liczb całkowitych, w której R_{ij} jest liczbą godzin, które dany nauczyciel musi przeprowadzić z każdą klasą.

Należy ustalić, czy istnieje funkcja spotkania nauczyciela z daną klasą określona następująco: $f(i, j, h) : \{1 \dots n\} \times \{1 \dots m\} \times H \rightarrow \{0, 1\}$: Wartość funkcji $f(i, j, h) = 1$, jeśli nauczyciel i prowadzi zajęcia z klasą j podczas godziny h . Funkcja ta powinna spełniać następujące warunki:

- spotkanie może mieć miejsce wtedy i tylko wtedy, gdy zarówno nauczyciel, jak i klasa są dostępne: $f(i, j, h) = 1 \Rightarrow h \in T_i \cap C_j$,
- liczba spotkań podczas tygodnia pomiędzy nauczycielem i oraz klasą j w wymaganym rozmiarze określana jest wartością R_{ij} :

$$\sum_{h \in H} f(i, j, h) = R_{ij}$$

dla każdego $i \in [1, n]$ oraz dla każdego $j \in [1, m]$,

- na każdą klasę przypada co najwyżej jeden nauczyciel w danym czasie:

$$\sum_{i=1}^n f(i, j, h) \leq 1$$

dla wszystkich $j \in [1, m]$ oraz $h \in H$,

- nie ma nauczyciela, który uczy w dwóch klasach równocześnie:

$$\sum_{j=1}^m f(i, j, h) \leq 1$$

dla wszystkich $i \in [1, n]$ oraz $h \in H$.

Spełnienie wszystkich przedstawionych warunków jest konieczne, aby powstał prawidłowo ułożony plan zajęć, ale nie jest wystarczające. Udowodniono, że złożoność obliczeniowa algorytmu rozwiązującego problem TT jest wielomianowa, jeżeli przyjęte zostanie ograniczenie, że wszyscy nauczyciele są 2-nauczycielami. k -nauczyciel, jest to taki nauczyciel, który może poświęcić danej klasie k godzin, czyli

$$\sum_{j=1}^m R_{ij} = k$$

W praktyce, rozwiązując rozważany problem, nie można przyjąć żadnych ograniczeń co do liczby godzin prowadzonych przez nauczycieli. Należy natomiast uwzględnić wiele dodatkowych uwarunkowań i ograniczeń, o których wspomniano na wstępie.

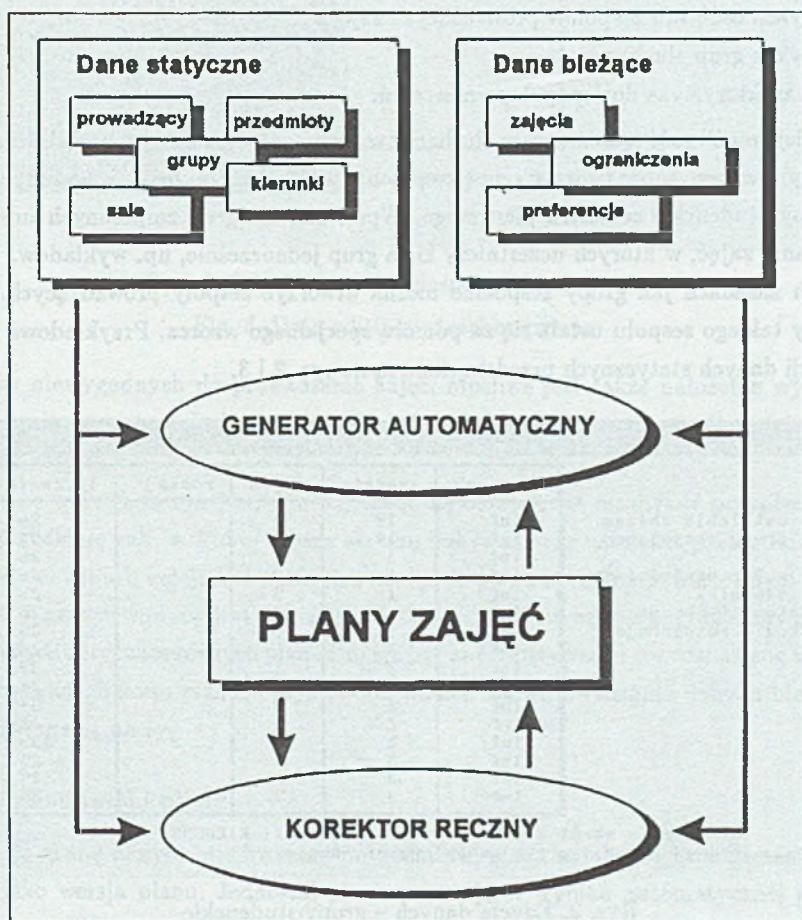
3. Charakterystyka systemu PLAN

System wspomagający układanie planów zajęć dla szkół wyższych PLAN powstał w Instytucie Informatyki Politechniki Śląskiej w latach 1992–1994. System operuje na złożonych strukturach danych, które zorganizowane są w bazy opisane w systemie Paradox 3.5. Centralną częścią systemu jest moduł automatycznej generacji planu zajęć, który napisany jest w języku C. System przystosowany jest do pracy na komputerach klasy IBM-PC pod systemem operacyjnym DOS.

Po wprowadzeniu danych wejściowych można zlecić systemowi automatyczną generację określonej wersji planu lub dokonywać modyfikacji już istniejących wersji planu zajęć. Wyniki można wyświetlać na ekranie monitora w postaci zestawu plansz z tygodniowymi planami zajęć dla poszczególnych grup słuchaczy, osób prowadzących zajęcia i pomieszczeń dydaktycznych. Można także zlecić systemowi drukowanie tych plansz.

3.1. Struktura danych systemu PLAN

Ogólny schemat ilustrujący przepływ danych w systemie PLAN przedstawiony jest na rys. 1.



Rys. 1. Schemat przepływu danych w systemie PLAN

Fig. 1. Data flow diagram in the PLAN system

Dane wejściowe systemu można podzielić na trzy grupy: dane statyczne, dane bieżące oraz jednostki planu.

3.1.1. Dane statyczne

Dane statyczne opisują środowisko związane z realizacją zajęć dydaktycznych. Dane te są jednokrotnie gromadzone w systemie i rzadko zmieniane. Należą do nich:

- wykaz kierunków studiów,
- parametry zajęć dydaktycznych, przewidzianych planem studiów,
- wykaz osób lub zespołów prowadzących zajęcia,
- wykaz grup słuchaczy,
- charakterystyka dostępnych pomieszczeń.

Istnieje możliwość łączenia grup słuchaczy w grupy zespolone, np. grupa słuchaczy semestru pierwszego może tworzyć grupę zespoloną, w skład której będą wchodziły wszystkie grupy studenckie semestru pierwszego. Wprowadzenie grup zespolonych umożliwia planowanie zajęć, w których uczestniczy kilka grup jednocześnie, np. wykładów. Na podobnych zasadach jak grupy zespolone można utworzyć zespoły prowadzących. Skład osobowy takiego zespołu ustala się za pomocą specjalnego wzorca. Przykładowe wzorce do edycji danych statycznych przedstawione są na rys. 2 i 3.

Edycja danych - GRUPY STUDENCKIE					
[F1]-pomoc [F2]-zakończenie [F6]-wzrost [Esc]-przerwanie					
	kierunek	semestr	grupa	rodzaj	liczność
[F3] - ustalanie składu grupy zespolonej	inf	10	4		24
	inf	10		z	94
	inf	1	1		26
[F5] - wybór wartości ze słownika: [Enter] - akceptacja, [Esc] - rezygnacja.	inf	1	2		26
	inf	1	3		25
	inf	1	4		26
	inf	1	5		25
	inf	1		z	120
	inf	2	1		33
	inf	2	2		34
	inf	2	3		34
	inf	2	4		32
	inf	2		z	133
inf	3	1		25	
inf	3	2		25	
inf	3	3		26	

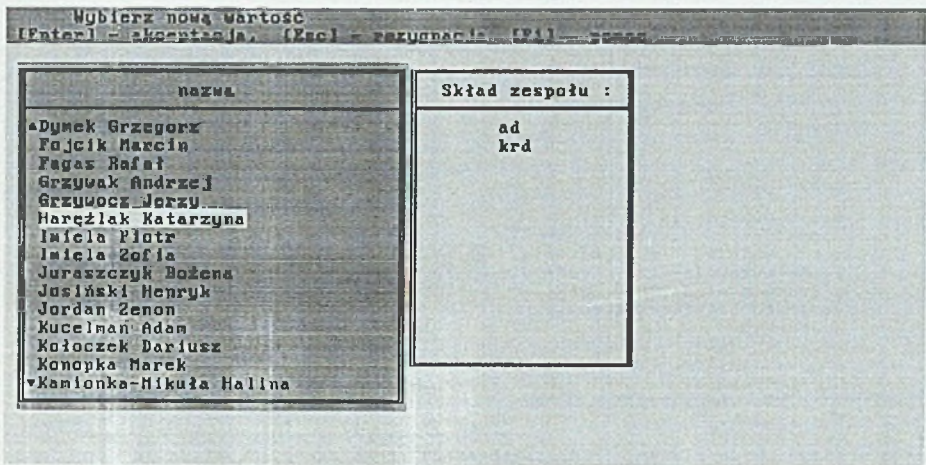
kierunek = skrót kierunku studiów z tabeli KIERUNKI
rodzaj = z (grupa zespolona np. semestr)

Rys. 2. Edycja danych – grupy studenckie

Fig. 2. Data editing – student groups

3.1.2. Dane bieżące

Dane bieżące dotyczą konkretnego planu dla słuchaczy danego semestru lub roku. Należą do nich: pula przewidzianych zajęć oraz lista postulatów i ograniczeń zgłaszana przez przyszłych użytkowników planu. Postulaty dotyczą terminów preferowanych oraz



Rys. 3. Edycja danych – zespół prowadzących

Fig. 3. Data editing – teacher groups

terminów niewygodnych do prowadzenia zajęć. Możliwe jest także nałożenie wymagań co do terminu rozpoczęcia i zakończenia zajęć określonego typu oraz długości przerwy pomiędzy zajęciami.

Podczas wprowadzania danych o zajęciach użytkownik ma możliwość przypisania zajęciom określonej sali, w której muszą się one odbywać oraz ustalenia priorytetu zajęć. Priorytet decyduje o kolejności wstawiania zajęć do planu i przyjmuje wartości od 0 do 7. Priorytet 0 zarezerwowany jest dla zajęć, dla których termin będzie przydzielony ręcznie. Dane bieżące dotyczące różnych planów mogą być zapamiętywane i rozpoznawane na podstawie wybranego kodu planu. Przykładowe wzorce dla wprowadzania danych bieżących przedstawione są na rys. 4 i 5.

3.1.3. Jednostki planu

Trzecią grupę danych stanowią gotowe jednostki planu ustalone i zapamiętane przez system jako wersja planu. Jednostki planu powstają w wyniku automatycznej generacji lub w wyniku ręcznych modyfikacji i stanowią krotkę: < zajęcia, grupa słuchaczy, prowadzący, sala, termin > .

3.2. Ręczna korekcja planu

Moduł ręcznej korekcji planu umożliwia wprowadzenie do istniejących wersji planu dowolnych zmian oraz wizualizację aktualnej postaci planu zajęć.

Edycja danych - OGRANICZENIA dla planu z93				
[F1]-pomoc [F2]-zakonczenie [F6]-wudruk [Esc]-przerwanie				
proW.	oper.	arg.	zaj.1	zaj.2
ad	<	10	c	
ad	p	2	w	u
chj	>	12	l	
chj	>	12	c	

proW = kod prowadzącego:
 oper = operator ograniczenia:
 < nie później niż o godzinie,
 > nie wcześniej niż o godzinie,
 p przerwa między zajęciami,
 arg = wartość ograniczenia lub
 przerwy
 zaj = rodzaj zajęć:
 u - wykład,
 c - ćwiczenia,
 l - laboratorium,
 p - projekt.

[F3] - zmiana kodu planu [F5] - wybór kodu prowadzącego ze słownika

Rys. 4. Edycja danych – ograniczenia

Fig. 4. Data editing – constraints

Użytkownik może usuwać terminy zajęć oraz przydzielać zajęciom nowe terminy zarówno we wstępnej fazie układania planu zajęć, jak i po fazie automatycznej generacji. W module korzysta się z danych statycznych, danych o zajęciach zaplanowanych i nie zaplanowanych, informacji o terminach wolnych dla nauczycieli, grup słuchaczy i dla sal, natomiast nie uwzględnia się preferencji i ograniczeń.

Zajęcia mogą być planowane w jednym z trzech trybów: cotygodniowym, parzystym i nieparzystym. Wstawiając nowe zajęcia do planu program sprawdza zajętość prowadzącego i grupy oraz czy są wolne odpowiednie dla wstawianych zajęć sale (pod względem rodzaju zajęć i liczności grupy). W przypadku, gdy są wolne sale, użytkownik ma możliwość wyboru jednej z nich.

Na ekranie, plan zajęć przedstawiony jest w postaci grafiku podzielonego na poszczególne dni tygodnia i strefy czasu o długości jednostki lekcyjnej począwszy od godziny 8.00 do 20.00. Rysunek 6 przedstawia grafik planu zajęć, z którego korzysta się podczas ręcznej korekcji planu. Komórki grafiku są puste, jeśli brak jest zajęć zaplanowanych dla danej grupy albo wyświetlone są informacje o zaplanowanych zajęciach takie, jak: kod zajęć, kod prowadzącego zajęcia oraz rodzaj zajęć. Wyświetlenie w komórce grafiku znaków "#" oznacza, że grupa jest w danym terminie zajęta, odbywając zajęcia w innym trybie niż aktualnie wybrany. W przypadku, gdy na planie grupy zespolonej w komórce grafiku wyświetlane są znaki "@", oznacza to, że w danym terminie zajęcia są już zaplanowane dla jednej z grup składowych.

Edycja danych - REPERTUAR ZAJĘĆ dla planu z93

[F1]-menue [F2]-kończąc [F6]-usuwać [Esc]-przerwanie

	grupa	kod	godz.	tryb	prom.	rodz.	pr.	sala
[F3] -	inf1z	UF	2	t	pwf	c	0	osir
zmiana kodu planu	inf1z	J01	2	t	pjo	c	0	sd_jo
aktualnego;	inf1z	J02	2	t	pjo	c	0	sd_jo
[F5] -	inf1z	MH1E	2	t	pnh	w	0	au_222
wybór wartości ze	inf1z	F1Z	2	t	pf	w	0	gr_300
słownika;	inf11	F1Z	2	d	pf	c	0	
[Enter] - akceptacja,	inf12	F1Z	2	d	pf	c	0	
[Esc] - rezygnacja.	inf13	F1Z	2	d	pf	c	0	
	inf14	F1Z	2	d	pf	c	0	
	inf15	F1Z	2	d	pf	c	0	
	inf1z	ALG	2	t	pal	w	0	
	inf11	ALG	2	d	pal	c	0	
	inf12	ALG	2	d	pal	c	0	
	inf13	ALG	2	d	pal	c	0	
	inf14	ALG	2	d	pal	c	0	
	inf15	ALG	2	d	pal	c	0	

tryb = t(isp) / d(co dwa tygodnie) sala = kod sali

grupa = kierunek + semestr + numer_grupy (dla lab. i zajęć

rodzaj = w(wykład) / c(ćwiczenia) / l(laboratorium) o priorytecie 0)

Rys. 5. Edycja danych - repertuar zajęć

Fig. 5. Data editing - a set of lectures

3.3. Automatyczna generacja planu zajęć

Centralną częścią systemu PLAN jest moduł automatycznej generacji realizujący wstawianie do planu zajęć z puli zajęć nie zaplanowanych. Danymi wejściowymi modułu są dane statyczne, dane bieżące oraz gotowe jednostki planu. W wyniku pracy modułu otrzymujemy wygenerowaną wersję planu zajęć z rozmieszczonymi wszystkimi zajęciami lub ewentualnie wskazanie zajęć bez przydzielonych terminów w przypadku nałożenia sprzecznych wymagań. Moduł umożliwia automatyczną generację planu od podstaw lub uzupełnienie już istniejącej wersji planu powstałej w wyniku pracy korektora ręcznego.

4. Opis zastosowanego algorytmu oraz elementy optymalizacji

4.1. Przyjęte założenia realizacyjne

Jak wspomniano wcześniej, problem układania planu zajęć jest jednym z tych problemów kombinatorycznych, dla których nie jest znany efektywny algorytm rozwiązania. Algorytm nazywamy algorytmem o czasie wielomianowym, jeżeli rząd wielkości czasu potrzebnego na jego wykonanie jest ograniczony z góry przez funkcję wielomianową, której argumentem jest rozmiar danych. Rozwiązanie problemu układania rozkładu zajęć

MODUŁ KOREKCYI PLANU ZAJĘĆ

Grafik planu zajęć dla grupy Infr

Tryb: Co tydzień

	Poniedz.	Wtorek	Środa	Czwartek	Piątek	Sobota
- 8:00		J01 c pJo	J02 c pJo	F12 w pf		
- 9:00		J01 c pJo	J02 c pJo	F12 w pf		
- 10:00		AN w pan				
- 11:00		AN w pan				
- 12:00		AN w pan	ALG w pal			
- 13:00			ALG w pal			

F1 Pomoc F2 Tryb F3 Grupa F4 Opis F5 Usun Enter Wstaw Esc Koniec

Rys. 6. Grafik planu zajęć

Fig. 6. The timetable sheet

poprzez przegląd całej przestrzeni rozwiązań prowadzi do wykładniczego czasu obliczeń. Jednym ze sposobów rozwiązywania tak trudnych zadań jest osłabienie warunku uzyskania optymalnego rozwiązania i zadowolenie się rozwiązaniami suboptymalnymi, zbliżonymi do rozwiązania optymalnego. Przyjęcie takich założeń redukuje czas obliczeń z wykładniczego na wielomianowy.

W opisywanym algorytmie założenie uzyskania optymalnego rozwiązania zostało osłabione na rzecz optymalizacji lokalnej z możliwością zmiany parametrów funkcji oceny jakości rozwiązania. W rezultacie, do rozwiązania rozważanego problemu zastosowano częściowo sparametryzowany algorytm heurystyczny. Autorem pierwszej wersji algorytmu jest D. Głowacki [4]. Ograniczenie przestrzeni wszystkich stanów rozwiązań następuje poprzez selekcję kierunków poszukiwań w tej przestrzeni. W przypadku układania planu zajęć sprowadza się to do określania kolejności, w jakiej zajęcia będą wstawiane do planu oraz kolejności wyszukiwania potencjalnych terminów.

4.2. Kryteria oceny jakości terminu

Podstawą funkcji oceny jakości terminu są ograniczenia ogólne dotyczące terminów zajęć grup słuchaczy i prowadzących oraz indywidualne preferencje i wymagania zgłaszane przez przyszłych użytkowników planu. Dodatkowym elementem oceny są przyjęte założenia dotyczące jakości planu, takie jak:

- spójność i maksymalna ciągłość zajęć grup słuchaczy i prowadzących,
- równomierne rozłożenie zajęć w tygodniu,
- równomierne wykorzystanie pomieszczeń dydaktycznych.

W opisywanym algorytmie przyjęto założenie, że każda z grup informacji wpływająca na jakość terminu generuje tabelę liczbową. Tabele mają strukturę odpowiadającą rzeczywistemu terminarzowi z podziałem na dni tygodnia i strefy czasowe. Tabele generowane są dla każdego wstawianych zajęć i obejmują następujące grupy informacji:

- tabela zajętości prowadzącego – bieżąca postać planu zajęć dla prowadzącego zajęcia,
- tabela zajętości grupy – bieżąca postać planu zajęć dla grupy,
- tabele preferencji i ograniczeń prowadzącego – wykaz terminów preferowanych i niewygodnych dla prowadzącego, określonych na podstawie ankiety,
- tabela preferencji grupy – wartości liczbowe przypisane terminom zależnie od pory dnia; preferowane godziny przedpołudniowe,
- tabela ograniczeń szczegółowych – opis warunków sąsiedztwa, następstwa zajęć,
- tabela wolnych sal – wykaz dostępnych sal dydaktycznych uwzględniający rodzaj i rozmiar sali.

Powyższe tabele łączone są w jedną tabelę wynikową, nazwaną tabelą wskaźników jakości stref czasowych *TWJSC*. Na podstawie wartości liczbowych zawartych w tej tabeli możemy każdej strefie czasowej przyporządkować określoną wartość, nazwaną wskaźnikiem jakości strefy. Terminy zajęte mają przypisaną wartość 0. Wyższe preferencje otrzymują terminy zachowujące ciągłość zajęć, zwłaszcza terminy uzupełniające dotychczasowe luki w planie oraz terminy zapewniające równomierne rozłożenie zajęć dla grup w poszczególnych dniach tygodnia. Dla zajęć planowanych co drugi tydzień, preferowane są terminy uzupełniające już zajęte terminy komplementarne (parzyste i nieparzyste). Tabela wskaźników jakości stref jest podstawą do obliczenia wskaźnika jakości terminu *wjt*. Wskaźnik ten równy jest sumie wskaźników jakości stref czasowych składających się na dany termin. Zajęciom zostaje przydzielony termin o największym wskaźniku jakości.

4.3. Postać algorytmu

Wygenerowanie wersji planu zajęć sprowadza się do wstawiania do planu zajęć z listy zajęć nie zaplanowanych. Moduł realizujący to zadanie działa według następującego algorytmu.

Algorytm 1. Automatyczna generacja planu zajęć

```
procedure GENERACJA;
```

```
begin
```

```
  while pula zajęć nie jest pusta do
```

```
  begin
```

```
    zajęcia := WYBIERZ_ZAJĘCIA;
```

```
    if WYBIERZ_TERMIN(zajęcia) = sukces
```

```
    then
```

```
      wstaw zajęcia do planu;
```

```
    else
```

```
      przenieś zajęcia do puli rezerwowej;
```

```
  end
```

```
end;
```

Zajęcia z puli zajęć nie zaplanowanych wybierane są za pomocą następującej funkcji.

Algorytm 2. Wybór zajęć do rozplanowania

```
function WYBIERZ_ZAJĘCIA;
```

```
begin
```

```
  utwórz listę zajęć nie zaplanowanych uporządkowaną według następujących kryteriów:
```

- priorytet zajęć { ustalany podczas wprowadzania danych bieżących }
- rodzaj zajęć { wykłady, ćwiczenia, laboratoria, projekty }
- licznosc grupy;

```
  WYBIERZ_ZAJĘCIA := pierwsze zajęcia z listy;
```

```
end;
```

Dla wybranych zajęć dokonuje się próby znalezienia najodpowiedniejszego terminu.

Dla każdego z zajęć wykonywane są kolejno:

- próba wstawienia bezpośredniego w którymś z wolnych terminów,
- próba wstawienia z wyłączeniem zajęć już zaplanowanych,
- próba wstawienia bez uwzględnienia preferencji prowadzącego.

Każdy z kolejnych etapów wykonywany jest w przypadku niepowodzenia etapu poprzedniego. Zadanie przydzielania terminów dla poszczególnych zajęć realizuje następujący algorytm.

Algorytm 3. Przydzielanie terminów dla zajęć

```
function WYBIERZ_TERMIN(zajęcia);
begin
  WYBIERZ_TERMIN := sukces;
  if WSTAW_BEZPOŚREDNIO(zajęcia) = porażka
  then
    if WSTAW_Z_WYWŁASZCZENIEM(zajęcia, 0) = porażka
    then
      if WSTAW_BEZ_PREF(zajęcia) = porażka
      then
        WYBIERZ_TERMIN := porażka
      end;
    end;
  end;
end;
```

Wybór terminu dla zajęć w trybie wstawiania bezpośredniego realizuje funkcja WSTAW_BEZPOŚREDNIO, której parametrem są planowane zajęcia .

Algorytm 4. Wstawianie bezpośrednio zajęć do planu

```
function WSTAW_BEZPOŚREDNIO(zajęcia);
begin
  utwórz tabelę wskaźników jakości stref TWJSC;
  wyznacz wartości wskaźników jakości terminów  $wjt$  jako sumy
    wskaźników jakości stref  $wjsc$ ;
  wszystkie terminy, których  $wjt > 0$  wpisz do listy uporządkowanej malejąco;
  if lista terminów jest pusta
  then
    WSTAW_BEZPOŚREDNIO := porażka
  else
    begin
      wybierz z utworzonej listy termin o największym  $wjt$ ;
      przypisz termin planowanym zajęciom;
      zaznacz zajęcia jako zaplanowane;
      WSTAW_BEZPOŚREDNIO := sukces
    end
  end;
end;
```

Próbę wstawiania zajęć z wywłaszczaniem zajęć już zaplanowanych realizuje funkcja WSTAW_Z_WYWŁASZCZENIEM, której parametrami są wstawiane zajęcia oraz licznik wywłaszczzeń.

Algorytm 5. Wstawianie z wywłaszczaniem zajęć już zaplanowanych

```
function WSTAW_Z_WYWŁASZCZENIEM(zajęcia, licznik_wywłaszczeń);
begin
  utwórz nową TWJSC, w której nie eliminuj terminów zajętych
    u prowadzącego i grupy;
  wyznacz liczbę konfliktów zajęć wstawianych z zajęciami już
    zaplanowanymi (ze względu na zajętość grupy lub prowadzącego);
  utwórz nową listę terminów, odrzucając terminy, które powodują
    konflikt wielokrotny (konflikt z więcej niż jednym prowadzącym lub grupą)
    i uporządkuj ją malejąco;
  while lista terminów nie jest pusta i nie było sukcesu do
  begin
    pobierz kolejny termin z utworzonej listy;
    ustal, z którymi zajęciami zaplanowanymi jest konflikt;
    wywłaszcz zajęcia zaplanowane i przypisz uzyskany termin zajęciom wstawianym;
    zapamiętaj historię wywłaszczeń;
    licznik_wywłaszczeń := licznik_wywłaszczeń + 1;
    if WSTAW_BEZPOŚREDNIO(zajęcia_wywłaszczone) = sukces
    then
      zasygnalizuj sukces
    else
      begin
        if licznik_wywłaszczeń > głębokość_wywłaszczeń
        then
          odtwórz, na podstawie historii wywłaszczeń, stan planu zajęć,
            jaki był przed wywłaszczeniem
        else
          begin
            if WSTAW_Z_WYWŁASZCZENIEM(zajęcia_wywłaszczone, licznik) = sukces
            then
              zasygnalizuj sukces
            else
              odtwórz, na podstawie historii wywłaszczeń, stan planu zajęć,
                jaki był przed wywłaszczeniem;
          end
        end
      end
    end
  end
```



```

end
end
if nie było sukcesu
then
  WSTAW_Z_WYWŁASZCZENIEM := porażka
else
  begin
    przypisz termin planowanym zajęciom;
    zaznacz zajęcia jako zaplanowane;
    WSTAW_Z_WYWŁASZCZENIEM := sukces;
  end
end;

```

Głębokość wyłączeń ogranicza poziom rekurencyjnych wywołań funkcji `WSTAW_Z_WYWŁASZCZENIEM`. Funkcja `WSTAW_BEZ_PREF(zajęcia)` próbuje przydzielić termin zajęciom nie uwzględniając preferencji prowadzącego. Funkcja ta różni się od funkcji `WSTAW_BEZPOŚREDNIO(zajęcia)` tylko sposobem tworzenia tabeli wskaźników jakości stref `TWJSC`. Wśród tabel, z których powstaje `TWJSC`, nie ma tabeli preferencji prowadzącego.

W przypadku, kiedy niemożliwe jest znalezienie terminu dla zajęć lub gdy nie są spełnione wszystkie wymagania dotyczące planu zajęć, użytkownik może wpływać na wynik końcowy poprzez zmianę danych wejściowych i ponowną generację planu lub poprzez ręczne ustalenie terminów zajęć.

4.4. Elementy optymalizacji

Optymalizacja postaci planu uzależniona jest od funkcji oceny jakości terminu wyznaczonej na podstawie tabeli wskaźników jakości stref `TWJSC`. Dodatkowo, końcowa postać planu może być sterowana kolejnością wyboru zajęć z puli zajęć nie zaplanowanych, np.:

- według priorytetów prowadzących,
- według priorytetów zajęć (wykłady, ćwiczenia, laboratoria),
- równomiernie zgodnie z powyższymi kryteriami dla poszczególnych grup.

Jak wspomniano wcześniej, algorytm zawiera punkty decyzyjne o charakterze heurystycznym. W tych punktach przeprowadzana jest selekcja różnych postaci rozwiązań, która odbywa się zgodnie z przyjętymi zasadami tworzenia planu zajęć:

- zasada konwersji tabeli zajętości do tabeli `TWJSC`; preferowanie ciągłości zajęć,

- zasada łączenia tabel do tabeli wynikowej *TWJSC*; wyważenie, w jakim stopniu poszczególne czynniki wpływają na harmonijny układ planu,
- zasada poszukiwań wolnych terminów w kolejnych cyklach wyłączeń.

Przyjęte zasady można korygować tak, aby lepiej były dostosowane do konkretnego problemu lub problemów związanych z uwarunkowaniami danej szkoły, uczelni czy wydziału, co może pomóc w uzyskaniu optymalnego rozwiązania.

5. Uwagi końcowe i wnioski

Przyjęty schemat rozwiązania jest jednym z wielu możliwych, jednak wydaje się dość naturalny w odniesieniu do czynności wykonywanych przy ręcznym układaniu planu. Dokonanie pewnych uproszczeń kryteriów optymalizacji i zadowolenie się rozwiązaniami suboptymalnymi doprowadziło do uzyskania efektywnego obliczeniowo algorytmu. Projektując system Plan korzystano z wiedzy i doświadczenia zespołu układającego plan zajęć dla studentów studiów dziennych Wydziału Automatyki, Elektroniki i Informatyki, na którym system był praktycznie wypróbowany w semestrze letnim roku akademickiego 1993/94. Plan zajęć stworzony przez moduł automatycznej generacji został zaakceptowany przez użytkowników.

LITERATURA

- [1] Even S., Itai A., Shamir A.: *On the complexity of timetable and multicommodity flow problems*. SIAM J. on Computing, vol. 5, no.4, 1976, 691-703.
- [2] Reingold E. M., Nievergelt J., Deo N.: *Algorytmy kombinatoryczne*. PWN, Warszawa 1985.
- [3] Błażewicz J.: *Problemy optymalizacji kombinatorycznej - złożoność obliczeniowa, algorytmy aproksymacyjne*. PWN, Warszawa-Lódź 1986.
- [4] Szmaj P. i inni.: *System doradczy dla realizacji planowania. Sprawozdanie z realizacji projektu*. Instytut Informatyki Politechniki Śląskiej, Gliwice 1992.
- [5] Colomi A., Dorigo M., Maniezzo V.: *Genetic Algorithms: A New Approach to the Timetable Problem*. NATO ASI Series, vol. F 82, Combinatorial Optimization 1992.

Recenzent: Dr hab. inż. Stanisław Wolek

Wpłynęło do Redakcji 16 grudnia 1994 r.

Abstract

The paper presents the algorithm implemented in the PLAN timetabling system. It also includes system's structure and its principles of operation.

System Plan offers a computer assistance to the construction of a timetable for academic-level school. It is a kind of an advisory system, which cope with complex data structures (see chapter 3). The data flow diagram in system Plan was shown on figure 1. The data is stored in data bases implemented in Paradox system and involve:

- list of teachers,
- list of student groups,
- list of available classrooms,
- same external conditions,
- existing elementary plan units.

The patterns of input data editing were shown on figures 2,3,4,5. The central part of the system is the automatic plan generation module. Its role is to assign terms to lectures to be accomplished with respect to personal goals and didactic goals (see chapter 3.3). The assignment of terms is carried out according to a heuristic algorithm. The description of the implemented algorithm's operation was discussed in chapter 4. System Plan has been tested in defining the timetable for the Computer Science Institute. The timetable which was created by this system acquired full acceptance plan users.

Summary. The article describes construction of microprograms of various database management software configurations, which are network computers. The analysis of experimental results has been carried out according to the acceptance of the configurations in question. The experiments are described (the hardware platform, operating system and software utilities) and their results are objectively evaluated. The results may be treated as universal and developed independent of specific configurations and configurations of microprogram software configurations they may be helpful in creating a specific solution.