PRACE NAUKOWE UNIWERSYTETU EKONOMICZNEGO WE WROCŁAWIU nr 108
Współczesne tendencje rozwojowe badań operacyjnych 2010

Bogumiła Krzeszowska, Tadeusz Trzaskalik
Akademia Ekonomiczna w Katowicach

Ewa Nocoń
Szkoła Podstawowa nr 5 w Olkuszu

ZASTOSOWANIE ALGORYTMÓW GENETYCZNYCH DO UKŁADANIA PLANÓW LEKCJI NA PRZYKŁADZIE SZKOŁY PODSTAWOWEJ NR 5 W OLKUSZU

Streszczenie: Układanie planu szkolnego jest problemem takiego ustalenia sekwencji spotkań nauczycieli ze studentami w określonym przedziale czasu, aby były spełnione różnego typu ograniczenia. Ręczne ułożenie planu, w zależności od stopnia jego złożoności, może zająć od kilku godzin do kilku dni, a plan taki może być niedoskonały pod wieloma względami. Z tego powodu zwrócono uwagę na możliwość automatyzacji procesu układania planów. Głównymi metodami automatycznego planowania są: metody heurystyczne, metody poszukiwania lokalnego oraz redukcja do kolorowania grafów. Praca ma na celu zastosowanie algorytmów genetycznych do ułożenia planu szkolnego. Pomimo dużej złożoności problemu algorytm genetyczny w niedługim czasie znalazł rozwiązanie dopuszczalne. Jako że jest to metoda przybliżona, otrzymany wynik nie musi być rozwiązaniem optymalnym.

Słowa kluczowe: algorytmy genetyczne, modele optymalizacyjne, harmonogramowanie

1. Wstęp

Układanie planu szkolnego jest problemem ustalenia sekwencji spotkań nauczycieli z uczniami w określonym przedziale czasu, aby były spełnione różnego rodzaju ograniczenia [Schaerf 1995]. Ograniczenia możemy podzielić na dwie grupy: ograniczenia twarde oraz ograniczenia miękkie. Ograniczenia twarde to takie, które muszą być spełnione, aby problem miał rozwiązanie dopuszczalne (m.in. grupa uczniów (klasa) nie może mieć w tym samym czasie więcej niż jedno zajęcie; nauczyciel nie może prowadzić w tym samym czasie więcej niż jedna lekcja; w jednej sali nie może się znajdować więcej niż jedna klasa [De Werra 1997]). Ograniczenia miękkie są to dodatkowe ograniczenia, które spełnione być nie muszą.

Ograniczenia te w główniej mierze mogą dotyczyć preferencji osób, dla których plan jest układany.

Z problemem optymalizacji mamy do czynienia, gdy poszukujemy planu spełniającego wszystkie ograniczenia twarde i minimalizującego (lub maksymalizującego) funkcję celu zawierającą ograniczenia miękkie.

Problem układania planów jest problemem NP-zupełnym[1], dlatego też idealne rozwiązanie może być znalezione tylko w przypadkach o niewielkich rozmiarach [Norberciak 2002].

Praca ma na celu próbę zastosowania algorytmów genetycznych do ułożenia planu szkolnego. W artykule przedstawiono postać matematyczną planu lekcji, a także metody automatycznego planowania. Następnie omówiono zastosowanie algorytmów genetycznych w układaniu planów oraz ułożono plan dla szkoły podstawowej. Pomimo dużej złożoności algorytm genetyczny w niedługim czasie znalazł rozwiązanie dopuszczalne. Ze względu na to, że jest to metoda przybliżona, otrzymany wynik nie musi być rozwiązaniem optymalnym.

2. Postać matematyczna problemu układania planu lekcji [De Werra 1985]

Oznaczamy przez:

– $i = 1, ..., m$ – liczba klas $\{c_1, ..., c_m\}$,
– $j = 1, ..., n$ – liczba nauczycieli $\{t_1, ..., t_n\}$,
– $k = 1, ..., p$ – liczba terminów $\{1, ..., p\}$,
– $R_{m \times n}$ – macierz wymagań, której elementy określają, ile lekcji klasa c_i ma z nauczycielem t_j (macierz nieujemnych liczb całkowitych),
– r_{ij} – element macierzy $R_{m \times n}$, oznacza liczbę lekcji, jakie nauczyciel t_j daje klasie c_i,
– x_{ijk} – element macierzy x będącej szukanym rozwiązaniem, gdzie i odpowiada klasie, j odpowiada nauczycielowi, a k odpowiada terminowi.

[1] Problem NP-zupełny to problem, który należy do klasy NP (może być rozwiązany w czasie wielomianowym) oraz jest NP-trudny (jego rozwiązanie jest co najmniej tak trudne, jak rozwiązanie każdego problemu klasy NP). Innymi słowy, każdy problem należący do NP można zredukować w czasie wielomianowym do problemu NP-zupełnego. Taka definicja problemów NP-zupełnych implikuje fakt, że jeśli tylko potrafimy rozwiązać jakikolwiek problem NP-zupełny w czasie wielomianowym, to potrafimy rozwiązać w czasie wielomianowym wszystkie problemy NP. Problemy NP-zupełne można więc traktować jako najtrudniejsze problemy klasy NP (z punktu widzenia wielomianowej rozwiązywalności).

Problem układania planów lekcji jako zadanie optymalizacji (podejście Jungingera) ma następującą funkcję celu:

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{p} d_{ijk} x_{ijk} ,$$

gdzie d_{ijk} to przypisanie do terminów k, w których lekcja nauczyciela t_j z klasą c_i jest mniej pożądana.

Rozwiązując zadanie planowania, szukamy x_{ijk} spełniającego następujące warunki (podstawowy problem poszukiwania):

1. $\sum_{k=1}^{p} x_{ijk} = r_{ij}$.
2. $\sum_{j=1}^{n} x_{ijk} \le t_{jk}$.
3. $\sum_{i=1}^{m} x_{ijk} \le c_{ik}$.
4. $x_{ijk} = \begin{cases} 1, & \text{gdy klasa } c_i \text{ i nauczyciel } t_j \text{ spotykają się w terminie } k \\ 0, & \text{gdy klasa } c_i \text{ i nauczyciel } t_j \text{ nie spotykają się w terminie } k \end{cases}$,

$t_{jk} = 1$ $(c_{ik} = 1)$, gdy nauczyciel t_{jk} (klasa c_{ik}) jest dostępny w czasie k oraz $t_{jk} = 0$ $(c_{ik} = 0)$ nie jest dostępny w czasie k.

Warunek 1 daje pewność, że nauczyciel ma z daną klasą określoną liczbę zajęć zapisaną w macierzy wymagań. Warunek 2 oznacza, że nauczyciel może mieć zajęcia z co najwyżej jedną klasą w danym terminie oraz że termin ten jest terminem, w którym nauczyciel jest dostępny. Warunek 3 zapewnia, że klasa ma co najwyżej jedne zajęcia w danym terminie oraz że jest to termin, w którym klasa jest dostępna.

Można rozważać również przypisanie danej lekcji w określonym czasie. Przypisanie to możemy wyrazić poprzez dodanie zbioru ograniczeń następującej postaci:

$$(i = 1, ..., m; \quad j = 1, ..., n; \quad k = 1, ..., p) ,$$

gdzie $p_{ijk} = 0$, gdy nie ma żadnych przypisań oraz $p_{ijk} = 1$, gdy lekcja nauczyciela t_j lub klasy c_i jest przypisana do czasu k.

Istnieje rozwiązanie powyższego problemu, jeżeli żadna klasa ani żaden nauczyciel nie ma więcej lekcji niż p:

$$\sum_{i=1}^{m} r_{ij} \le p ,$$

$$\sum_{j=1}^{n} r_{ij} \le p,$$

gdzie p oznacza liczbę interwałów czasowych.

3. Metody automatycznego planowania

Ręczne ułożenie planu, w zależności od stopnia jego złożoności, może zająć od kilku godzin do kilku dni. Skonstruowany w ten sposób plan może być niedoskonały pod wieloma względami. Z tego powodu zwrócono uwagę na możliwość automatyzacji procesu układaniu planów. Pierwsze prace z tej dziedziny powstały w latach sześćdziesiątych ubiegłego stulecia. Od tego czasu zostało wdrożonych wiele aplikacji rozwiązujących problem układania planów w sposób efektywny [Norberciak 2002].

Metody heurystyczne – bezpośrednie heurystyki wypełniają plan, wkładając do niego po jednym zdarzeniu tak długo, aż nie pojawi się konflikt. Od tego momentu zamienia się zdarzenia miejscami albo usuwa z planu, tak aby znalazło się miejsce dla innych. Najprostszym podejściem heurystycznym jest metoda porządkowa. Podstawą tej metody jest określenie pewnej strategii porządkowania. Używa się heurezy w celu ustalenia, jak trudne do rozwiązania byłoby rozważane zdarzenie, aby można było je ułożyć w kolejności malejącej trudności. Podobnej strategii można użyć do ułożenia w kolejności dostępnych terminów. Metody porządkowania mają jedną wadę – dają tylko jedno rozwiązanie, co może oznaczać, że istnieje inne, równie dobre, a nawet lepsze. Aby je znaleźć, należałoby zmienić strategię porządkowania lub wprowadzić do heurystyki elementy niedeterministyczne [Norberciak 2002].

Metody poszukiwania lokalnego – podstawową koncepcją lokalnego przeszukiwania jest pojęcie sąsiedztwa rozwiązania x oznaczanego $N(x) \subset D$ (D – przestrzeń rozwiązań dopuszczalnych), tj. zbioru rozwiązań, które można uzyskać, wykonując jeden ruch (modyfikację) m z pewnego zbioru możliwych ruchów $M(x)$. Ruchem nazywamy modyfikację transformującą rozwiązanie w jeden z jego sąsiadów. Sąsiedztwo można zdefiniować następująco:

$$N(x) = \{y \mid \exists m \in M(x) \mid y = m(x)\}.$$

Ruch powinien być stosunkowo prostą modyfikacją rozwiązania x. Rozmiar sąsiedztwa powinien być dużo mniejszy niż wielkość całej przestrzeni rozwiązań i z reguły, choć nie zawsze, jest wielomianowo zależny od wielkości instancji problemu [Jaszkiewicz 2005].

Metoda rozpoczyna działanie od pewnego rozwiązania początkowego wygenerowanego w sposób losowy lub przez zastosowanie innych metod. Następnie algorytm wchodzi w pętlę, po której porusza się po przestrzeni rozwiązań, przechodząc od rozwiązania do jednego z sąsiadów [Norberciak 2002].

Redukcja do kolorowania grafów – podejście do układania planów lekcji oparte na redukcji grafu do kolorowania nazywa się harmonogramowaniem chromatycznym. Poszczególnym zdarzeniom odpowiadają węzły, istnienie krawędzi pomiędzy dwoma węzłami oznacza, że zdarzenia te nie mogą być zaplanowane w tym samym terminie. Kolor węzła reprezentuje termin. Taki model jest dobry dla prostych zadań układania planów, może on być jednak rozszerzony tak, aby uwzględniał także słabe ograniczenia. Rozszerzenie takie obejmuje przedstawienie wymagań w postaci multigrafu, co pozwala na uwzględnienie zdarzeń z góry przypisanych do zbioru możliwych terminów, i niedostępność niektórych terminów dla pewnych zdarzeń [Norberciak 2002].

Algorytmy genetyczne – używają one łańcuchów binarnych (struktur) zwanych chromosomami, które reprezentują dziedzinę optymalizowanej funkcji. Każdy chromosom składa się z genów. Zbiór struktur tworzy populację. Algorytm wykonuje wielokierunkowe przeszukiwania poprzez przekształcenia populacji potencjalnych rozwiązań i prowadzi do zbierania informacji genetycznej i jej wymiany między tymi kierunkami.

4. Algorytmy genetyczne w układaniu planów

Intensywne badania nad zastosowaniem algorytmów genetycznych do rozwiązywania problemów planowania są prowadzone od początku lat dziewięćdziesiątych XX wieku.

Możemy wyróżnić dwie podstawowe reprezentacje osobnika w problemie układania planów lekcji: reprezentacja macierzowa (zaprezentowana po raz pierwszy przez Colorni, Dorigo i Maniezzo [1992]) oraz reprezentacja wektorowa (zaprezentowana po raz pierwszy w pracy Ross i Corne [1995])

W macierzowej reprezentacji osobnika wiersz odpowiada nauczycielowi, kolumna terminowi, a jej elementami są klasy. Użyto trzech operatorów genetycznych: mutacji rzędu k (operator wybiera dwa sąsiednie k-elementowe ciągi z tego samego wiersza macierzy i zamienia je miejscami), mutacji dni (zamienia miejscami dwie grupy kolumn – godzin – macierzy, które odpowiadają różnym dniom) oraz krzyżowania (dla danych dwóch macierzy operator ustawia wiersze pierwszej macierzy w porządku malejącym tzw. *lokalnej funkcji celu*, która jest składową funkcji celu związaną tylko z charakterystyką danego nauczyciela). Uwzględniono tylko silne ograniczenia.

W ramach postaci wektorowej osobnik jest wektorem symboli długości $3v$ (gdzie v jest liczbą zdarzeń), podzielonym na trójki (reprezentacja bezpośrednia). Poszczególne elementy w trójce oznaczają kolejno czas, miejsce i nauczyciela. Autorzy zaproponowali również operator mutacji, który wybiera losowo kilka wartości mutowanej części rozwiązania i przyjmuje za nowe rozwiązanie tę, która daje największą poprawę funkcji celu. Uwzględniono zarówno ograniczenia silne, jak i słabe.

5. Zastosowanie programu FET do ułożenia planu lekcji w Szkole Podstawowej nr 5

Program FET[2] został napisany w języku C++ z wykorzystaniem biblioteki Qt oraz pakietu KDevelop. Dane do tworzenia planu program pobiera z plików w formacie XML.

Przystępując do układania planu, musimy przyporządkować określonych nauczycieli określonym przedmiotom oraz określonym grupom uczniów. Jeżeli połączymy ze sobą te trzy elementy, to uzyskamy pewien element podstawowy, który nazwiemy zajęciami (rys. 1). Jednostka zajęciowa będzie stanowiła podstawowy element, z którego zbudowane są chromosomy, a więc gen.

Zajęcia 1	Zajęcia 2	Zajęcia 3	…	Zajęcia N

Rys. 1. Konstrukcja chromosomu w programie FET

Źródło: [Jaszuk 2003].

W programie FET wyróżnione zostały dwie strategie układania planu. Plan można ułożyć, przydzielając zajęciom godziny i sale jednocześnie (wówczas mamy $N_g^N \times N_s^N$ możliwych kombinacji). Można także obliczenia podzielić na dwa etapy. W pierwszym etapie zajęciom przyporządkowuje się godzinę, a dopiero później salę. Podział obliczeń na dwa etapy może przyspieszać obliczenia. Jest to możliwe dzięki znacznej redukcji wymiaru przestrzeni przeszukiwania. Jeśli obliczenia prowadzone są w dwóch osobnych etapach, wymiary przestrzeni przeszukiwania się sumują (wówczas liczba możliwych kombinacji wynosi $N_g^N + N_s^N$) [Lalescu 2003]. Podejście to ma jednak poważną wadę. W pierwszym etapie zajęcia umieszczane są w zbiorze interwałów czasowych, bez uwzględnienia jakichkolwiek informacji o dostępności sal. Przystępując do rozmieszczania zajęć w salach, algorytm genetyczny ma już ściśle określone godziny zajęć, których nie można zmieniać. Jeżeli liczba dostępnych w danej godzinie sal danego typu jest większa lub równa liczbie zajęć, to rozmieszczenie zajęć w salach nie powinno stanowić większych trudności. Może się jednak zdarzyć sytuacja, w której liczba zajęć będzie większa od liczby dostępnych sal określonego typu. W takim przypadku problem staje się już na wstępie nierozwiązywalny.

Parametry programu FET:

Populacja początkowa – w programie występują dwie metody inicjalizowania populacji: Inicjalizuj (NIEPRZYDZIELONE), która jest eksperymentalną metodą twórców programu, oraz Inicjalizuj (LOSOWO). Inicjalizowanie „nieprzydzielo-

[2] Autorem programu FET jest Liviu Lalescu [2003].

ne” oznacza, że symulacja rozpoczyna się od ustanowienia „pustych” planów, podczas gdy inicjalizacja losowa oznacza, że inicjowane są chromosomy, które mają przyporządkowane wszystkie zajęcia w losowym terminie.

Program ma możliwość ustalenia liczby populacji z zakresu 512 do 8192. Wraz ze wzrostem liczby populacji zwiększa się zapotrzebowanie na moc obliczeniową komputera, zwiększa się także dokładność obliczeń. W populacjach o mniejszej liczebności szybciej osiąga się wyniki, lecz nie są one tak dokładne i dobre jak wyniki uzyskane w przypadku zastosowania większej populacji.

Budowa chromosomu – każdy chromosom składa się z genów reprezentujących każde zajęcia w planie. Chromosom jest macierzą genów. Gen reprezentuje interwały czasowe dla zaplanowanych zdarzeń. Zdarzenie to spotkanie nauczyciela z klasą.

Funkcja przystosowania – mierzy stopień dopasowania chromosomu. Każdy chromosom ma czynnik dopasowania ograniczeń twardych i czynnik dopasowania ograniczeń miękkich, które reprezentują liczbę niespełnionych ograniczeń. Funkcja przystosowania jest liczona następująco: dopasowanie ograniczeń twardych jest mierzone liczbą konfliktów ograniczeń twardych, dopasowanie ograniczeń miękkich jest liczone jako suma wag konfliktów ograniczeń miękkich. W programie występuje także funkcja porównania, której zadaniem jest rozróżnienie, który chromosom jest lepszy. Funkcja porównuje dwa chromosomy i wybiera ten, który ma lepsze dopasowanie ograniczeń twardych, czyli ten, która ma mniej konfliktów ograniczeń twardych. Jeśli obydwa porównywane chromosomy mają tyle samo konfliktów ograniczeń twardych, wybierany jest ten, który ma niższą wartość funkcji dopasowania graniczeń miękkich.

Metody ewolucji – metody definiujące strategię generowania nowej populacji przy użyciu operatorów genetycznych. W programie dostępne są dwie: 3-etapowa metoda klasyczna oraz eksperymentalna metoda zaproponowana przez autorów. W metodzie klasycznej z bieżącej populacji wybiera się losowo 3 chromosomy, każdy z równym prawdopodobieństwem. Jeśli chromosom przechodzi do następnej populacji bez zmian lub jeśli ma nastąpić mutacja, wybierany jest jeden z nich (najlepszy). Jeśli nastąpić ma krzyżowanie, wybierane są dwa najlepsze. W metodzie eksperymentalnej każdy etap ewolucji składa się z generowania populacji z podwojoną liczbą jednostek z bieżącej populacji, wybierana jest połowa składająca się z najlepszych chromosomów. Podwajanie populacji zachodzi poprzez krzyżowanie lub mutację. Chromosomy do krzyżowania lub mutacji są wybierane z takim samym prawdopodobieństwem z bieżącej populacji.

Metody mutacji – dostępne są dwie metody: zamiana dwóch losowo wybranych genów oraz eksperymentalna metoda losowania pojedynczego genu (losowanie czasu rozpoczęcia zajęcia).

6. Obliczenia i wyniki

Należy ułożyć plan lekcji dla Szkoły Podstawowej nr 5 w Olkuszu, mając dane:
- 13 klas,
- 25 nauczycieli,
- 35 terminów,
- 18 sal,
- macierz wymagań $R_{m\times n}$: macierz nieujemnych liczb całkowitych, gdzie r_{ij} oznacza liczbę lekcji, jakie nauczyciel t_i daje klasie c_j (w tym przypadku macierz wymagań jest macierzą o wymiarach 13×25, łączna liczba zdarzeń wynosi 340),
- ograniczenia obowiązkowe (dodatkowo za ograniczenia obowiązkowe uznano terminy, w których nauczyciele są niedostępni),
- ograniczenia nieobowiązkowe (niedostępność nauczycieli w danych terminach oraz minimalna i maksymalna ilość godzin dla uczniów).

Parametry ustawione do obliczeń (w celu ustalenia parametrów programu dla zadanego problemu została przeprowadzona symulacja na przykładowych danych, co pozwoliło określić, w jaki sposób należy ustawić parametry w zadaniu, aby możliwe było uzyskanie lepszych wyników w krótszym czasie):

1. Liczba populacji: 3000.
2. Liczba pokoleń: 100 000.
3. Prawdopodobieństwa ewolucji:

10% – prawdopodobieństwo „rozpowszechnienia”, czyli oczekiwana liczba chromosomów, które pozostają takie same w następnej populacji,

70% – mutacja,

20% – krzyżowanie.

Taki stosunek parametrów zalecany jest przez twórców programu, którzy w toku symulacji pewnego zbioru uznali go za najlepszy [Lalescu 2003].

4. Metoda ewolucji:

Wybrano metodę klasyczną.

5. Metoda mutacji:

W obliczeniach wykorzystano obydwie proponowane przez autorów programu metody mutacji. Prawdopodobieństwa obydwu mutacji wynosiły 35% każda.

6. Metoda inicjalizacji:

Inicjalizuj (NIEPRZYDZIELONE).

Wyniki: Układanie planu zostało podzielone na dwa etapy. W pierwszym etapie do zdarzeń przydzielone zostają godziny, w drugim sale. Rozwiązanie dopuszczalne w etapie pierwszym zostało osiągnięte w 42 100 pokoleniu. Po zakończeniu obliczeń konflikty ograniczeń nieobowiązkowych wynosiły 72.

Tabela 1. Plan lekcji dla nauczyciela*

	Poniedziałek	Wtorek	Środa	Czwartek	Piątek
7:55-8:40					
8:45-9:30				IVa Przyroda sala 29	
9:40-10:25		IVa Przyroda sala 3			
10:35-11:20					
11:25-12:10					IVa Przyroda sala 24
12:20-13:05			Ib Zajęcia wyrównawcze sala 21		
13:25-14:10					IIa Zajęcia wyrównawcze sala 30

* Na szaro na planach zaznaczono niespełnione ograniczenia „miękkie".

Źródło: opracowanie własne.

Tabela 2. Plan lekcji dla klasy*

	Poniedziałek	Wtorek	Środa	Czwartek	Piątek
7:55-8:40					
8:45-9:30	Nauczyciel 1a Kształcenie zintegrowane sala 25			Nauczyciel 1a Kształcenie zintegrowane sala 24	
9:40-10:25	Nauczyciel 1a Kształcenie zintegrowane sala 20	Nauczyciel 1a Kształcenie zintegrowane sala 30	Nauczyciel 1a Kształcenie zintegrowane sala 25	Nauczyciel 1a Kształcenie zintegrowane sala 30	Nauczyciel 1a Kształcenie zintegrowane sala 25
10:35-11:20	Katecheta 1 Religia sala 30	Anglista 2 Język angielski sala 25	Polonista 1 Zajęcia wyrównawcze sala 24	Nauczyciel 1a Kształcenie zintegrowane sala 20	Nauczyciel 1a Kształcenie zintegrowane sala 25
11:25-12:10	Anglista 2 Język angielski sala 19	Nauczyciel 1a Kształcenie zintegrowane sala 20	Nauczyciel 1a Kształcenie zintegrowane sala 24	Nauczyciel 1a Kształcenie zintegrowane sala 20	Katecheta 1 Religia sala 19
12:20-13:05		Polonista 1 Zajęcia wyrównawcze sala 19	Nauczyciel 1a Kształcenie zintegrowane sala 30	Nauczyciel 1a Kształcenie zintegrowane sala 24	Nauczyciel 1a Kształcenie zintegrowane sala 20
13:25-14:10		Nauczyciel 1a Kształcenie zintegrowane sala 24	Nauczyciel 1a Kształcenie zintegrowane sala 20	Nauczyciel 1a Kształcenie zintegrowane sala 25	Nauczyciel WF 3 Gimnastyka korekcyjna sala 9

* Na szaro na planach zaznaczono niespełnione ograniczenia „miękkie".

Źródło: opracowanie własne.

W następnym etapie do zdarzeń z określonymi godzinami należało przydzielić sale. W tym przypadku okazało się, że zadanie nie posiada rozwiązania dopuszczalnego, ponieważ w jednej sali miały się odbywać zajęcia dwóch klas.

W związku z powyższym zostało wykonane jednoetapowe układanie planu, w którym jednocześnie przydzielane są godziny i zdarzenia. Rozwiązanie dopuszczalne zostało otrzymane w pokoleniu 16 000. Otrzymano 456 konfliktów ograniczeń nieobowiązkowych.

Rozwiązaniem problemu jest plan lekcji. Plany lekcji można przedstawić z różnych perspektyw, jako plan dla uczniów, nauczycieli, a także jako plan obłożenia sal. Poniżej przykłady takich planów (tab. 1, 2 oraz 3).

Tabela 3. Obłożenie sali

	Poniedziałek	Wtorek	Środa	Czwartek	Piątek
7:55-8:40		VIa(2) Język angielski Anglista I	Va Język polski Polonista 3		Va(2) Język angielski Anglista I
8:45-9:30		Ib Kształcenie zintegrowanie Nauczyciel Ib	Va Godzina wychowawcza Polonista 3	Ib Kształcenie zintegrowanie Nauczyciel Ib	VIa(1) Język angielski Anglista I
9:40-10:25	VIa Przyroda Nauczyciel przyrody	IVa Przyroda Dyrektor I	IVa Matematyka Matematyk I	VIa Plastyka nauczyciel plastyki	
10:35-11:20		Va Technika nauczyciel techniki	IVa Język angielski Anglista I	VIa Godzina wychowawcza Nauczyciel WF I	Va Język polski Polonista 3
11:25-12:10	Va Przyroda Nauczyciel przyrody	Ib Kształcenie zintegrowanie Nauczyciel Ib	Ib Kształcenie zintegrowanie Nauczyciel Ib	Va Religia Katecheta I	Ib Kształcenie zintegrowanie Nauczyciel Ib
12:20-13:05		Ib Kształcenie zintegrowanie Nauczyciel Ib	VIa Przyroda Nauczyciel przyrody		
13:25-14:10	Va Przyroda Nauczyciel przyrody				Va(1) Język angielski Anglista I

Źródło: opracowanie własne.

7. Podsumowanie

Algorytm genetyczny znalazł rozwiązanie dopuszczalne, niestety znalezienie rozwiązania optymalnego nie było możliwe. W takim przypadku warto rozważyć hybrydyzację algorytmu genetycznego, czyli połączenie go z wyspecjalizowaną techniką wyszukiwania. Hybrydyzacja umożliwia wykorzystanie „globalnego spojrzenia” algorytmu genetycznego z jednej strony i zbieżność techniki wyspecjalizowanej z drugiej. Idea ta jest dość prosta i może być stosowana w celu poprawy efektywności końcowej procesu poszukiwań genetycznych [Goldberg 2003].

Istnieje wiele sposobów hybrydyzacji algorytmu genetycznego. Wyróżniamy *podejście wsadowe*, przy którym algorytm genetyczny pracuje aż do osiągnięcia znacznego stopnia zbieżności populacji, a następnie włącza się procedura optymalizacji lokalnej, startując z 5-10% najlepszych punktów ostatniego pokolenia.

W *podejściu równoległym* zakładamy dostępność wielu pracujących równolegle procesów o mocy obliczeniowej wystarczającej do współbieżnego wyznaczania wartości funkcji dla poszczególnych ciągów kodowych w danym pokoleniu. Procesory współbieżne mogą być w ten sposób użyte do obliczania wskaźników przystosowania ciągów kodowych. Można ich również używać do wykonywania sporadycznych iteracji procedury lokalnego poszukiwania w celu ulepszenia bieżącego ciągu kodowego [Goldberg 2003].

Metodą lokalną, nadającą się do hybrydyzacji z algorytmem genetycznym, jest metoda iterowanych ulepszeń pozycyjnych. W metodzie tej wybiera się najlepsze ciągi z bieżącej populacji, zamienia się wartości kolejnych bitów w wybranym ciągu lub ciągach, zachowując za każdym razem lepszy z dwóch wariantów, a następnie włącza najlepszy znaleziony ciąg do populacji i kontynuuje poszukiwania genetyczne [Goldberg 2003].

Warto również rozważyć wprowadzenie systemów interaktywnych, które pozwalają manipulować swoim wyjściem. Przestrzeń poszukiwań w problemie układania planów jest bardzo duża, interwencja człowieka może nakierować poszukiwania w odpowiednią stronę.

Literatura

Colorni A., Dorigo M., Maniezzo V., *A genetic algorithm to solve the timetabling problem*, Politechnico di Milano 1992 (dostępne na http://www.asap.cs.nott..ac.uk/ASAP/publications).

De Werra D., *An introduction to timetabling*, "European Journal of Operation Research" 1985 vol.19 no 2, s. 151-162.

De Werra D., *The combinatorics of timetablig*, "European Journal of Operation Research" 1997 vol. 96.

Goldberg D., *Algorytmy genetyczne i ich zastosowania*, WNT, Warszawa 2003.

Jaszkiewicz A., *Metaheurystyki w praktyce*, [w:] *Algorytmy genetyczne*, *ewolucyjne i metaheurystyki*. *Wybrane zagadnienia*, red. T. Trzaskalik, Wydawnictwo AE, Katowice 2005.

Jaszuk M., *Zastosowanie algorytmów genetycznych do układania planu zajęć*, Państwowa Wyższa Szkoła Zawodowa w Chełmie, Chełm 2003.

Lalescu L., *Timetabling experiments using genetic algorithms*, Universitatea din Craiova 2003.

Norberciak M., *Przegląd metod automatycznego planowania – przykład wykorzystania algorytmu genetycznego w rozwiązaniu prostego problemu planowania*, Prace Naukowe Wydziałowego Zakładu Informatyki Politechniki Wrocławskiej. Zeszyt Sztuczna Inteligencja nr 1, PW, Wrocław 2002.

Ross P., Corne D., *Comparing genetic algorithms*, *simulated annealing*, *and stochastic hilliclimbing on timetabling problem*, Evolutionary Computing, AISB Workshop, 1995.

Schaerf A., *A survey of automated timetabling*, CWI Amsterdam 1995 (dostępne na: http://cwi.nl/ftp//CWIreports/AP).

THE APPLICATION OF GENETIC ALGORITHMS IN SCHOOL TIMETABLING ON THE EXAMPLE OF THE ELEMENTARY SCHOOL NO 5 IN OLKUSZ

Summary: The timetabling problem relies on scheduling a sequence of lectures between teachers and students in a prefixed period of time, satisfying a set of constraints of various types. The manual solution of the timetabling problem usually requires many person-days of work. In addition, the solution obtained may be unsatisfactory in some respect. For the above reason, a considerable attention has been devoted to automated timetabling. The most popular techniques for solving this problem are: simulated annealing, heuristics and graph coloring.

In this article, genetic algorithms were used to solve the timetabling problem. Genetic algorithms are a solution technique for optimization problems. They are an adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. Genetic algorithms have been applied for school timetabling first in early 90s.

The purpose of this paper is to use genetic algorithms for the school timetabling problem. The school timetabling problem is difficult to solve (underlying problem is NP-complete), but genetic algorithm found a solution in short period of time. Because genetic algorithms are the approximate methods, the acquired solution is not necessarily an optimal solution.