ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1997
Seria: INFORMATYKA z. 33 Nr kol. 1377

Bożena REBAJN
Politechnika Śląska, Instytut Informatyki

ZASTOSOWANIE ALGORYTMÓW GENETYCZNYCH DO PROBLEMU UKŁADANIA ROZKŁADU ZAJĘĆ W SZKOŁACH WYŻSZYCH[1]

Streszczenie. W artykule przedstawiono wyniki badań nad wykorzystaniem możliwości, jakie oferują algorytmy genetyczne do rozwiązywania problemu układania rozkładu zajęć. Problem ten, znany w literaturze jako *timetable* problem (TTP) [1], został wybrany z grupy złożonych problemów kombinatorycznych, N-P zupełnych [2,3]. W artykule przedstawiona została reprezentacja problemu, a mianowicie dobrano postać chromosomu, uściślono sposób działania operatorów genetycznych, zdefiniowano funkcję dopasowania. Wyniki badań pokazują, jaki wpływ na jakość uzyskiwanych rozkładów zajęć ma sposób przeprowadzenia reprodukcji i mutacji, procentowy udział operatorów mutacji i krzyżowania oraz rozmiar populacji i liczba generacji.

A GENETIC ALGORITHM TO SOLVE THE TIMETABLE PROBLEM IN HIGH SCHOOL

Summary. This paper presents the results of the investigation of posibilities offered by genetic algorithms to solve the timetable problem [1]. The timetable problem has been chosen as one of the difficult combinatorial, NP-hard problems [2,3]. The paper describes a model of the problem including the definition of chromosome, the genetic operators and the fitness function. The results of the experiments present how kinds of the reproduction and mutation operators, mutation and crossover percentage, population size and count of generations the timetable influence the quality of.

---

[1] Praca częściowo finansowana w ramach grantu KBN Nr 8 T11C 041 12.

1. Wprowadzenie

Algorytmy genetyczne (AG) są to algorytmy poszukiwania oparte na procesach naturalnej selekcji [6]. Rozpoczynając od początkowego zbioru możliwych rozwiązań generowanych losowo (populacji), modyfikuje się wybrane rozwiązania, aż do uzyskania rozwiązania opytymalnego. W AG każde możliwe rozwiązanie jest reprezentowane przez pojedynczego osobnika nazywanego chromosomem. Kolejne pozycje w chromosomie nazywane są genami, a wartość, którą w danej chwili przyjmuje gen, nazywa się allelem. Wartości te należą do określonego zbioru symboli.

Po wygenerowaniu początkowej populacji osobniki są losowo krzyżowane lub podlegają mutacji, co pozwala na rekombinację materiału genetycznego. Natępnie w procesie naturalnej selekcji wybierane są osobniki najlepiej dopasowane, które uczestniczą w tworzeniu nowej populacji. Odbywa się to za pomocą operatorów genetycznych, do których należą:

- reprodukcja; umieszcza w nowej populacji kopie tych osobników, których wartości funkcji dopasowania są wyższe od średniej wartości funkcji dopasowania całej populacji; funkcja dopasowania określa, w jakim stopniu dane rozwiązanie spełnia warunki narzucone przez konkretny problem;
- krzyżowanie; stosowane z pewnym określonym prawdopodobieństwem, polega na wybieraniu dwóch osobników z populacji w sposób losowy, a więc niezależny od ich indywidualnych cech, rekombinowaniu ich genów i tworzeniu w ten sposób dwóch potomków;
- mutacja; stosowana z określonym prawdopodobieństwem powoduje zmianę losowo wybranych genów pojedynczego osobnika.

Zastosowanie algorytmu genetycznego do konkretnego zadania wiąże się z koniecznością określenia sposobu kodowania zbioru parametrów tego zadania w postać skończonego ciągu znaków z określonego alfabetu, zdefiniowania funkcji dopasowania oraz uściślenia sposobu działania operatorów genetycznych.

2. Problem układania rozkładu zajęć

Układanie rozkładu zajęć w szkole wyższej polega na wiązaniu ze sobą informacji dotyczących:

- grup studentów,
- nauczycieli prowadzących zajęcia,

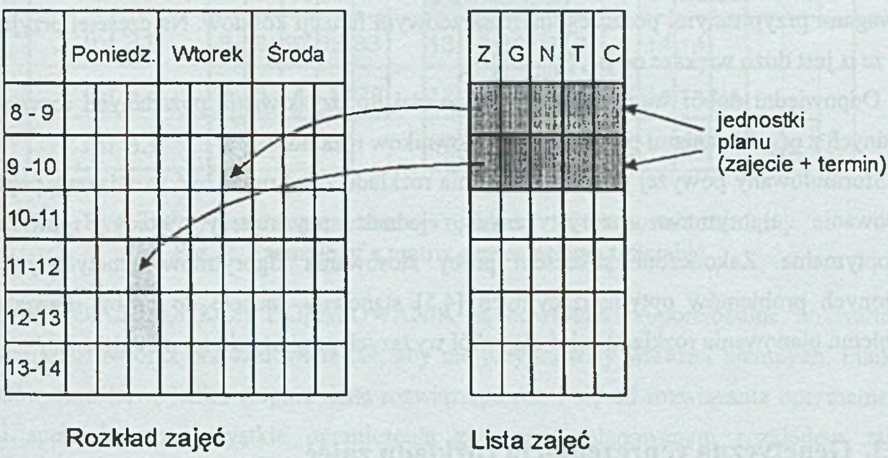
- listy zajęć, które należy przeprowadzić w danym semestrze, oraz
- dopuszczalnych godzin pracy w poszczególnych dniach tygodnia.

Zajęcia, które należy zaplanować, opisane są przez przedmiot nauczania (Z), numer grupy studenckiej (G), numer nauczyciela (N) oraz typ (T) i czas trwania zajęć (C). Zajęcia mogą być prowadzone przez nauczycieli z jedną grupą (ćwiczenia i laboratoria) lub z kilkoma grupami równocześnie (wykłady). Niektóre zajęcia muszą odbywać się w ściśle określonym terminie, co ustalane jest w fazie wstępnej. Dodatkowym problemem jest umiejętne gospodarowanie dostępnymi pomieszczeniami, ale ten temat traktowany jest jako oddzielne zagadnienie i nie jest przedmiotem rozważań w niniejszej pracy.

W algorytmie planowania przydziela się zajęciom ściśle określony termin ich rozpoczęcia (dzień oraz godzina), w taki sposób, że dana grupa i dany nauczyciel nie pojawiają się więcej niż raz w tym samym terminie. Działanie algorytmu planowania zilustrowano na rys. 1.

Prawidłowo ułożony rozkład powinien uwzględnić jeszcze dodatkowe zależności pomiędzy gotowymi jednostkami rozkładu, do których należą między innymi:

- ciągłość zajęć grup studentów i nauczycieli,
- równomierne rozmieszczenie zajęć w tygodniu,
- spełnienie indywidualnych ograniczeń i preferencji nauczycieli dotyczacych terminów poszczególnych zajęć.



Rys. 1. Ilustracja działania algorytmu układania rozkładu zajęć
Fig. 1. Illustration of timetable algorithm

W celu formalnego sformułowania problemu układania rozkładu zajęć wprowadzimy następujące pojęcia. Niech dana będzie piątka <C, A, H, R, f>, gdzie:

- C jest skończonym zbiorem $\{C_1, C_2, .. C_i, .. C_m\}$ grup studentów;
- A jest zbiorem zajęć, które należy zaplanować w danym semestrze;
- H jest skończonym zbiorem $\{H_1, H_2, .. H_j, .. H_n\}$ terminów;
- R jest macierzą o rozmiarze m × n o elementach r_{ij} należących do zbioru zajęć A;
- f jest funkcją kosztów, której wartość jest odwrotnie proporcjonalna do jakości danego rozkładu zajęć.

Problem ułożenia rozkładu zajęć polega na znalezieniu minimalnej wartości funkcji kosztów **min** $f(\Omega, \Pi, \Delta)$), gdzie:

- Ω jest liczbą konfliktów w rozkładzie zajęć (co zostanie zdefiniowane później),
- Π jest liczbą nie spełnionych założeń organizacyjnych (np. ciągłość zajęć grup studenów), zaś
- Δ jest liczbą nie spełnionych indywidualnych ograniczeń i preferencji.

Przyjmujemy, że każde generowane rozwiązanie nie zawiera konfliktów, jeżeli spełnia następujące warunki:

- wszystkie zajęcia z listy zajęć do zaplanowania mają przydzielony termin,
- każda grupa ma przypisanego co najwyżej jednego nauczyciela w danym terminie,
- żaden nauczyciel nie uczy w dwóch grupach równocześnie.

Funkcja kosztów dla rozkładu zajęć **R** jest równa $f = \alpha\,\Omega + \beta_1 \Pi + \beta_2 \Delta$, gdzie α, β_1, β_2 są wagami przypisanymi poszczególnym składowym funkcji kosztów. Najczęściej przyjmuje się, że α jest dużo większe od β_1, β_2.

Odpowiedni dobór wag składników prowadzi do uzyskiwania pożądanych rozwiązań, zgodnych z oczekiwaniami przyszłych użytkowników rozkładu.

Sformułowany powyżej problem układania rozkładu zajęć może być rozwiązany poprzez stosowanie algorytmów heurystycznych, jednak uzyskujemy wtedy rozwiązania suboptymalne. Zakończone sukcesem próby stosowania algorytmów genetycznych do złożonych problemów optymalizacyjnych [4,5] stanowiły zachętę do próby rozwiązania problemu planowania rozkładu zajęć dla szkół wyższych przy użyciu tej metody.

3. Genetyczna reprezentacja rozkładu zajęć

Stosując algorytmy genetyczne do omawianego problemu przyjęto następującą reprezentację:

CHROMOSOMEM jest rozkład zajęć, prezentowany w postaci macierzy **R** o rozmiarach $m \times n$ o elementach r_{ij} należących do zbioru A. Każdy wiersz tej macierzy reprezentuje daną grupę studentów, a każda kolumna kolejny termin (dzień oraz godzina). Przykładowa reprezentacja rozkładu zajęć w postaci macierzy przedstawiona jest na rys. 2.

ALFABETEM jest zbiór zajęć, które należy zaplanować w danym semestrze. Każde z zajęć jest opisane przez nazwę przedmiotu, nauczyciela, typ i czas trwania zajęć oraz status zajęcia. Status określa, czy dane zajęcia mają z góry określony termin, którego nie można zmienić, czy też może ono odbywać się w dowolnie wybranym terminie.

GENEM są pojedyncze zajęcia określone przez numer z listy zajęć do zaplanowania.

	Poniedziałek						Wtorek						Środa					
Grupy studentów	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
Inf 11	15	39							7	7	18	36	10	10				
Inf 12	16	19					9	9	7	7			10	10	37	40		
Inf 13	38	41					9	9	7	7	20		10	10	17			
Inf 21		8	8	4	24	24	11	11	12	12	42	42	21					
Inf 22	5	8	8				11	11	12	12	25		22	43	43			
Inf 23	23	8	8				11	11	12	12			6	26	26	44	44	
Inf 31	3	3	27	33	33		13	13	30	0	0		14	14				
Inf 32	3	3	1	1	28		13	13	34	34			14	14	31			
Inf 33	3	3		35			13	13	29	32			14	14		2	2	

Rys. 2. Przykładowa reprezentacja rozkładu zajęć w postaci macierzy
Fig. 2. Example of a matrix representing a timetable

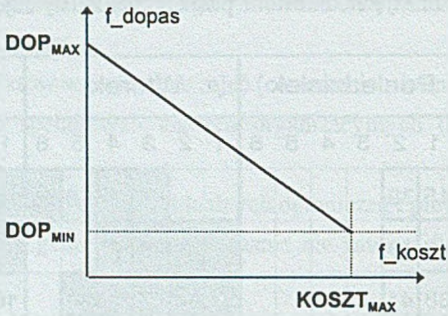
WARTOŚCI FUNKCJI DOPASOWANIA są odwrotnie proporcjonalne do wartości funkcji kosztów oraz przeskalowane tak, aby nie przyjmowały wartości ujemnych. Funkcja kosztów określa, w jakim stopniu dane rozwiązanie różni się od rozwiązania optymalnego, czyli spełniającego wszystkie ograniczenia związane z planowanym rozkładem zajęć. W implementacji założono, że wartości funkcji kosztów zależą od liczby konfliktów w rozkładzie zajęć oraz od liczby nie spełnionych założeń organizacyjnych (ciągłość zajęć grup studentów).

Funkcja dopasowania f_dopas(n) wyznaczana jest dla każdego osobnika z następującego wzoru:

$$\mathbf{f\_dopas(n) = KOSZT_{MAX} - f\_koszt(n) + \alpha}$$

gdzie:

- **$KOSZT_{MAX}$** jest maksymalną wartością funkcji kosztów osobników w populacji,
- **f_koszt(n)** jest aktualną wartością funkcji kosztów dla osobnika **n**,
- jest wartością kosztu pojedynczego konfliktu.



Rys. 3. Odwzorowanie funkcji kosztów w funkcję dopasowania
Fig. 3. Mapping the cost function into the fitness function

Natomiast funkcja kosztów wyznaczana jest następująco:

$$\mathbf{f\_koszt(n) = \alpha \cdot \Omega + \beta \cdot \Pi}$$

gdzie:

- Ω jest liczbą konfliktów,
- Π jest liczbą nieciągłości zajęć w rozkładzie, zaś
- β jest wartością kosztu za pojedynczą nieciągłość.

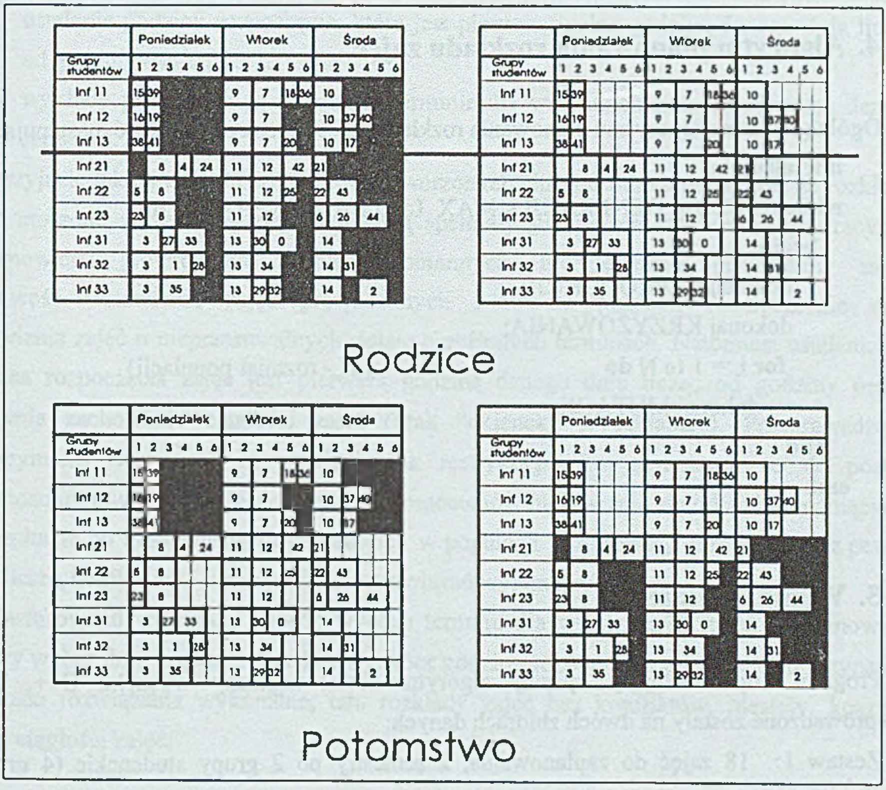
OPERATORY GENETYCZNE

Zastosowanie operatorów krzyżowania i mutacji do omawianego problemu w sposób dowolny może powodować takie zmiany w chromosomach, że osobniki potomne nie przedstawiają już wejściowego zbioru danych, tj. listy zajęć do zaplanowania. Dzieje się tak dlatego, iż operatory te powodują wymianę *jednostek planu* z określonego terminu jednego rodzica na *jednostki planu* z odpowiadającego terminu drugiego rodzica. W wyniku takiej wymiany pewne jednostki planu są gubione, a inne powielane u potomków. Wadę tę można korygować na dwa sposoby:

- zdefiniować genetyczne operatory w taki sposób, aby nie zmieniały wejściowego zbioru danych,
- zastosować operatory i dokonać genetycznej naprawy usuwając powstałe niewykonalności.

Zdecydowano się na pierwszy z wymienionych sposobów, tzn. tak zdefiniowano genetyczne operatory, aby zajęcia rozmieszczone w fazie wstępnej nie były przesuwane w wyniku zastosowania operatorów genetycznych, ponadto wykłady, które planowane są dla wszystkich grup danego semestru, były przesuwane dla wszystkich grup jednocześnie. Zastosowano trzy operatory genetyczne, zdefiniowane następująco:

REPRODUKCJA - umieszczanie w populacji osobników z prawdopodobieństwem wprost proporcjonalnym do wartości funkcji dopasowania z zastosowaniem metody ruletki [5].



Rys. 4. Przykład realizacji operacji krzyżowania
Fig. 4. Example of the crossover operation

KRZYŻOWANIE - losowanie z pewnym określonym prawdopodobieństwem par osobników danej populacji (rozkłady zajęć). Następnie losowe wybieranie punktu krzyżowania dla każdej pary i wymienianie *k* kolejnych rzędów jednego rodzica z pozostałymi *m-k* rzędami drugiego rodzica. Wybrany punkt krzyżowania nie może dzielić grup należących do jednego semestru. Przykład realizacji operatora krzyżowania przedstawiony jest na rys. 4.

MUTACJA - wybieranie grupy (rząd macierzy), następnie losowe wybieranie zajęcia (gen lub k przyległych genów, gdzie k oznacza czas trwania zajęć) i wymienianie ich z innymi zajęciami, których czas trwania nie jest krótszy niż czas trwania zajęć wylosowanych lub w otoczeniu których znajdują się wolne terminy. Zajęcia mające status "ustalone" nie podlegają mutacji.

4. Algorytm planowania rozkładu zajęć

Ogólną strukturę algorytmu planowania rozkładu zajęć można przedstawić następująco:

```
inicjalizacja;
while (LICZBA_GENERACJI < MAX_LICZBA_GENERACJI) do
  begin
    dokonaj REPRODUKCJI;
    dokonaj KRZYŻOWANIA;
    for i := 1 to N do                    { N - rozmiar populacji}
      dokonaj MUTACJI;
  end
end.
```

5. Wyniki obliczeń

Program implementujący opisany algorytm został napisany w języku C++. Testy przeprowadzone zostały na dwóch zbiorach danych:

Zestaw 1: 18 zajęć do zaplanowania, 2 semestry po 2 grupy studenckie (4 grupy), 2-dniowy rozkład zajęć z 5 godzinami dziennie, 5 prowadzących.

Zestaw 2: 45 zajęć do zaplanowania, 3 semestry po 3 grupy studenckie (9 grup), 3-dniowy rozkład zajęć z 6 godzinami dziennie, 7 prowadzacych.

Zaplanowanie wszystkich zajęć dla poszczególnych grup studenckich wypełniło dostępne terminy w 70%.

5.1. Tworzenie pierwszej populacji chromosomów

W fazie inicjalizacji tworzona jest populacja N chromosomów, z których każdy spełnia następujące ograniczenia:

- każda grupa ma zaplanowaną odpowiednią ilość zajęć w semestrze,
- pewne zajęcia mają przypisane ustalone terminy, które nie mogą być modyfikowane,
- zajęcia umieszczane są w rozkładzie zajęć w określonej kolejności zgodnie z przypisanym im priorytetem (najpierw wstawiane są wykłady, następnie ćwiczenia, a na końcu zajęcia laboratoryjne),
- dobór terminu dla zajęć dokonywany jest przez losowy wybór dnia tygodnia oraz ustalenie godziny rozpoczęcia, która jest pierwszą wolną godziną danego dnia licząc od godziny ósmej,
- wykłady planowane są w jednym terminie dla wszystkich grup składowych danego semestru.

Przyjęcie takich założeń co do postaci tworzonych chromosomów powoduje, że rozkłady zajęć utorzone w pierwszej populacji będą spełniały już pewne założenia organizacyjne, a mianowicie: preferowanie terminów porannych, uwzględnienie priorytetów zajęć, możliwość wstawienia zajęć prowadzonych z kilkoma grupami równocześnie oraz wstawienia zajęć o nieprzesuwalnych, ściśle określonych terminach. Natomiast ustalenie, że godzina rozpoczęcia zajęć jest pierwszą godziną danego dnia licząc od godziny ósmej zapewnia zachowanie ciągłości zajęć (brak "okienek" w rozkładzie). Przeprowadzone eksperymenty wykazały, że przyjęcie tak restrykcyjnych ograniczeń co do postaci chromosomu powodowało generowanie chromosomów nieznacznie się od siebie różniących. W rezultacie po kilkudziesięciu generacjach w populacji pozostawały chromosomy z pewną stałą liczbą konfliktów i nie uzyskiwano rozwiązań wykonalnych.

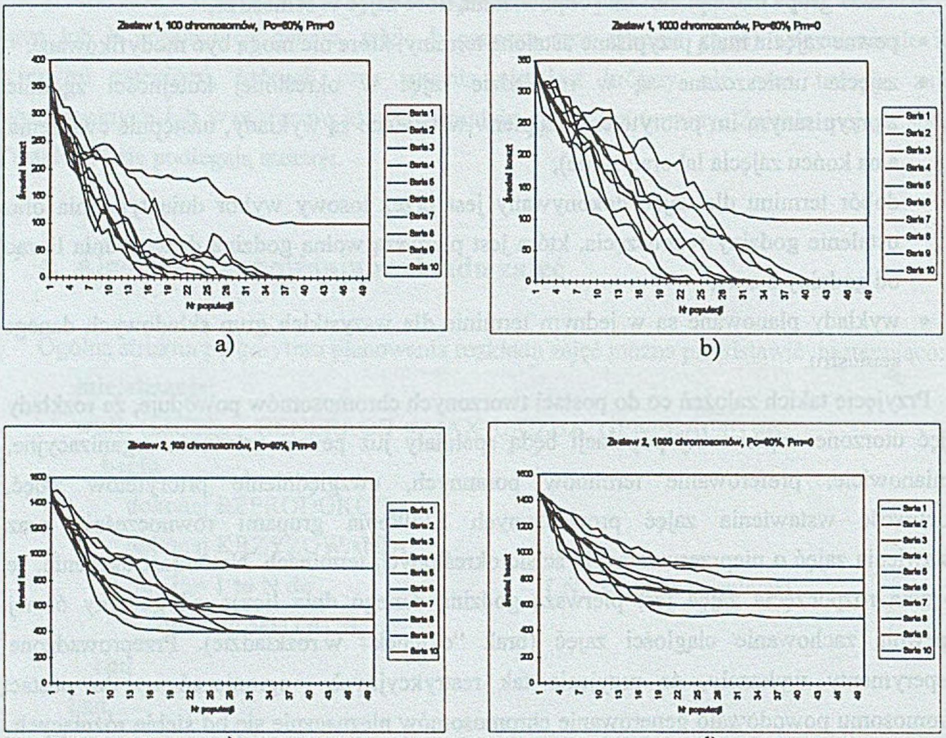
Zastosowano więc inny sposób wyboru terminu dla wstawianych zajęć, a mianowicie losowy wybór dnia tygodnia oraz losowy wbór godziny rozpoczęcia zajęć. W tym przypadku uzyskano rozwiązania wykonalne, tzn. rozkłady zajęć bez konfliktów, niestety, kosztem utraty ciągłości zajęć.

5.2. Dobór parametrów algorytmu genetycznego

Przyjęto wstępnie następujące wartości parametrów algorytmu genetycznego:

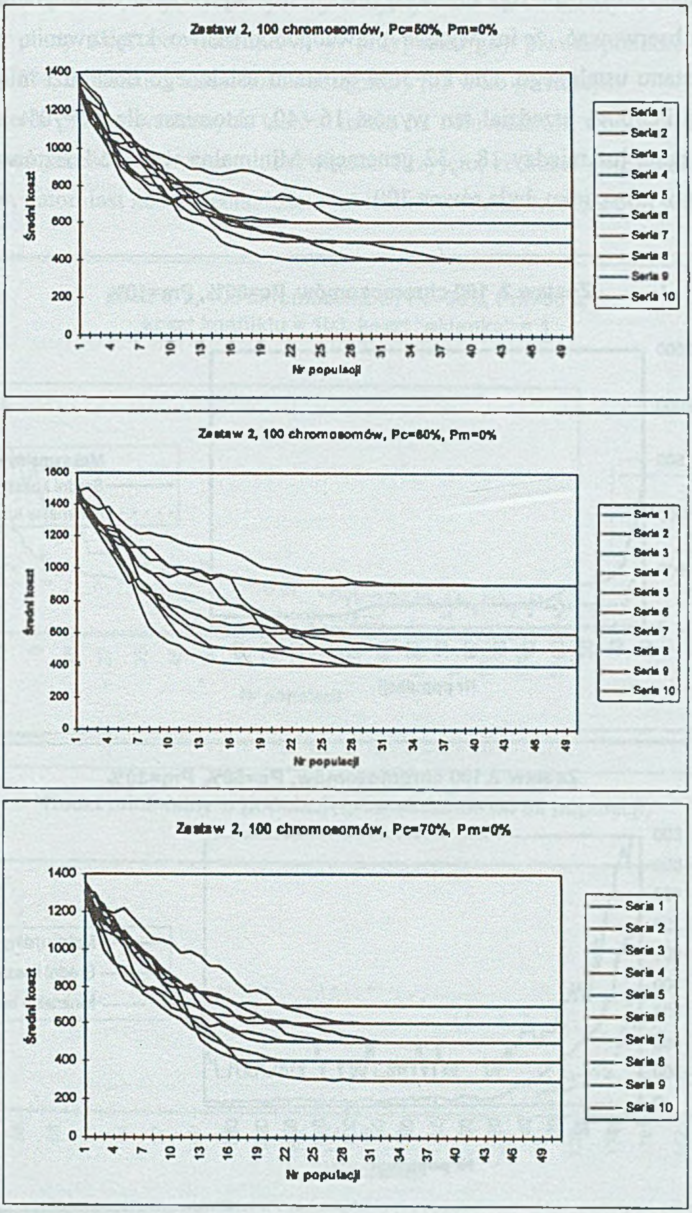
- liczba generacji - do 150,
- rozmiar populacji - 100 i 1000 chromosomów,
- prawdopodobieństwa zastosowania krzyżowania 50%, 60% i 70%,
- prawdopodobieństwa zastosowania mutacji 0%, 10%, i 30%.

Dla każdej kombinacji parametrów przeprowadzono po 10 testów. Wyniki przeprowadzonych eksperymentów przedstawione zostały na wykresach.



Rys. 5. Średnia wartość kosztów w populacjach 100- i 1000- elementowych
Fig. 5. Average costs in 100- and 1000- element populations

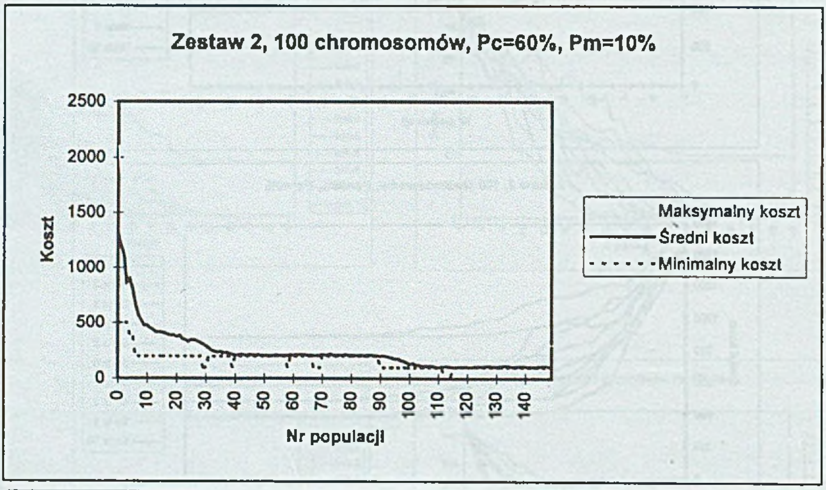
Dla liczby 100 lub 1000 chromosomów w populacji charakter zmian średniej wartości kosztów praktycznie się nie różni. Wynik ten potwierdził się dla obydwu zestawów danych. Wykresy na rys. 5a i 5b przedstawiają średnie wartości kosztów pierwszych 50 generacji pierwszego zestawu danych dla odpowiednio 100- i 1000- elementowych populacji. Wykresy na rys. 5c i 5d pokazują tę samą zależność dla drugiego zestawu danych. Można zatem przyjąć, że 100 chromosomów jest reprezentatywnym rozmiarem populacji.

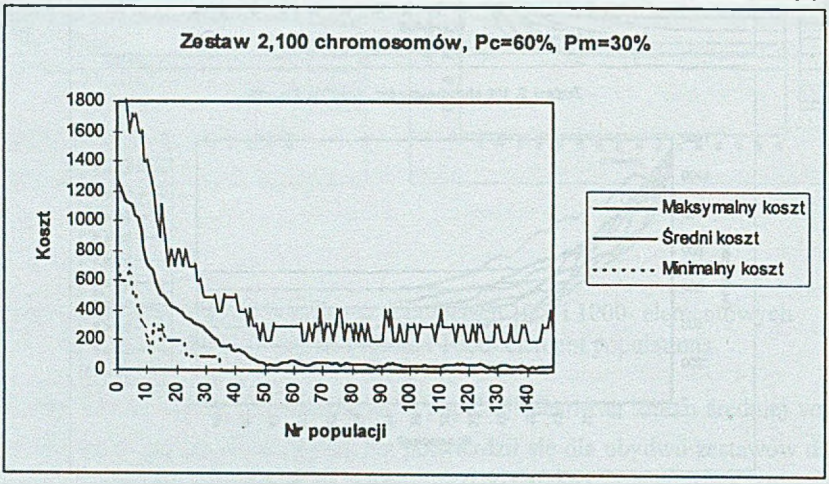


Rys. 6. Zależność zmian średniego kosztu od prawdopodobieństwa krzyżowania Pc

Fig. 6. Change of average cost with crossover probability Pc

Zastosowanie samego krzyżowania (bez stosowania mutacji) spowodowało, że populacja szybko osiągała stan niezmienniczy, już między 16 a 44 generacją wszystkie chromosomy były takie same i dalsza selekcja oraz krzyżowanie nie wprowadzały żadnych zmian. Warto przy tym zaobserwować, że im wyższe prawdopodobieństwo krzyżowania, tym szybciej dochodzi do stanu ustalonego. Dla Pc=50% do stanu ustalonego dochodzi między 18 a 44 generacją, dla Pc=60% przedział ten wynosi 16 - 40, natomiast dla Pc=70% stan ustalony populacje osiągają już między 18 - 32 generacją. Minimalna wartość kosztów, jaka została osągnięta w tym przypadku, była równa 300.



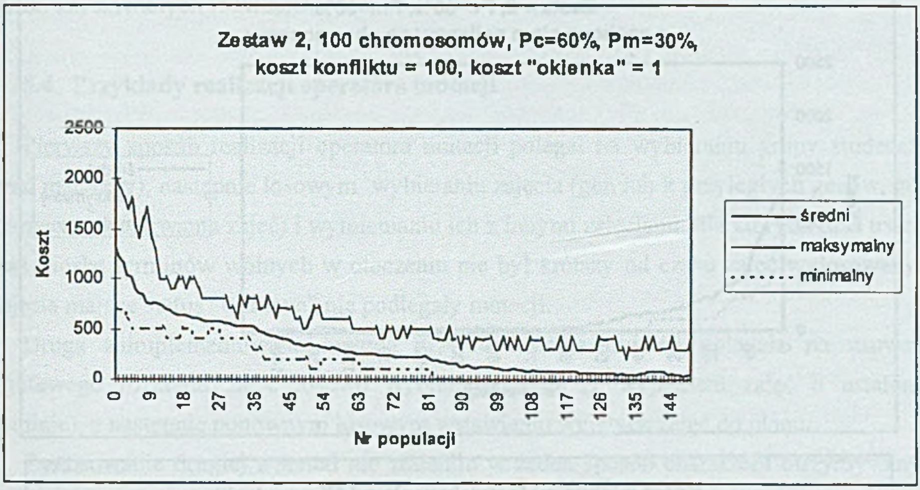


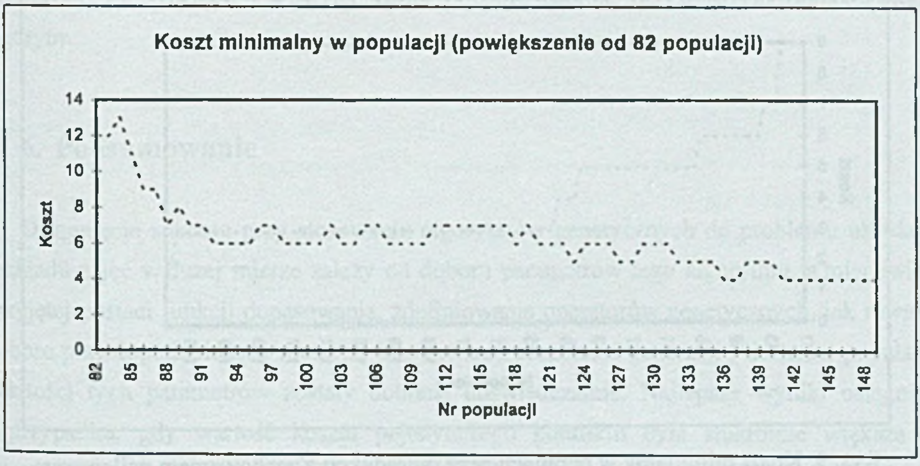
Rys. 7. Zależność zmian maksymalnego, średniego i minimalnego kosztu w populacji od prawdopodobieństawa mutacji Pm

Fig 7. Change of maximum, average and minimum cost with mutation probability Pm

Przy zastosowaniu mutacji pojedynczych zajęć dla losowo wybranych osobników osiągnięto różnorodność wyników aż do ostatniej generacji. Przy czym im wyższe było prawdopodobieństwo mutacji, tym szybciej otrzymywano chromosomy o zerowej wartości kosztów, tzn. plany zajęć bez konfliktów. W przypadku gdy prawdopodobieństwo mutacji wynosiło 20-30%, minimalny koszt został osiągnięty już w 25 generacji.

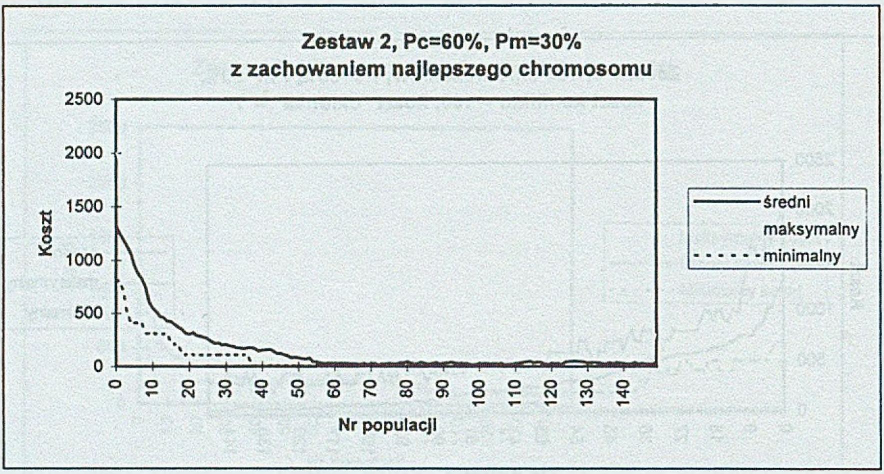
Kolejna seria przeprowadzonych testów miała na celu badanie wpływu postaci funkcji dopasowania na jakość uzyskiwanych rozwiązań. Przyjęto, że koszt pojedynczego konfliktu α wynosi 100, natomiast koszt nieciągłości ("okienka" w rozkładzie zajęć) β wynosi 1.

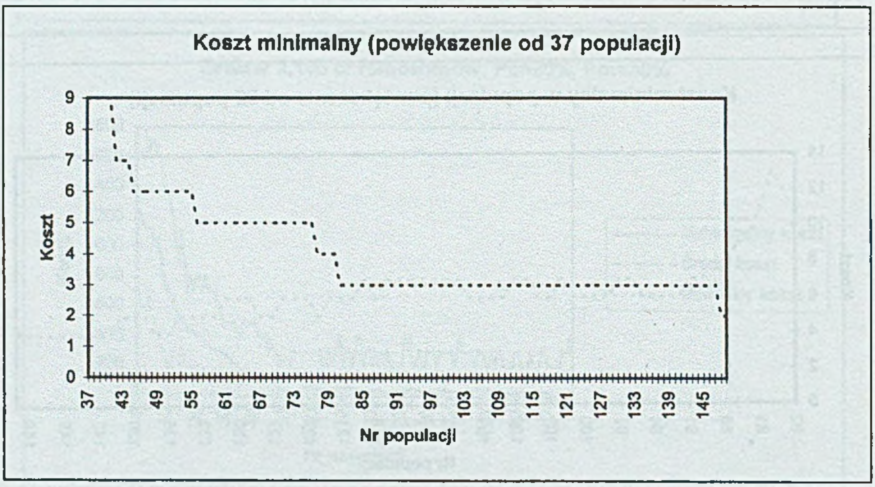




Rys. 8. Minimalny koszt w populacji po uwzględnieniu kosztów nieciągłości
Fig. 8. Minimum population cost concerning uncontinuity costs

Dobór stukrotnie mniejszej wartości kosztu nieciągłości zajęć w stosunku do wartości kosztu konfliktu spowodował, że nawet duża liczba nieciągłości w rozkładzie zajęć praktycznie nie wpływała na reprodukcję chromosomu, dopóki w populacji nie występowały chromosomy nie zawierające konfliktów. Natomiast w sytuacji zaistnienia w populacji chromosomów bez konfliktów liczba nieciągłości w rozkładzie zajęć zaczynała decydować o wyborze chromosomu do reprodukcji. Sytuację taką przedstawia rys. 8, gdzie od 82 populacji występują chromosomy, dla których wartość funkcji kosztów jest mniejsza od 100 (nie zawierające konfliktów).





Rys. 9. Koszt minimalny w populacji przy reprodukcji z zachowaniem najlepszego chromosomu

Fig. 9. Minimum population cost with best chromosome preserve reproduction

5.3. Przykłady realizacji operatora reprodukcji

Operator reprodukcji został zdefiniowany w następujący sposób: osobniki umieszczane były w populacji z prawdopodobieństwem wprost proporcjonalnym do wartości funkcji dopasowania z zastosowaniem metody ruletki [3]. Takie zdefiniowanie reprodukcji powodowało, że osobniki o najwyższej funkcji dopasowania nie zawsze wybierane były do kolejnej populacji. Drugi zaimplementowany operator reprodukcji działał również na zasadzie ruletki, natomiast miał wbudowany mechanizm zachowania najlepszego rozwiązania. Rysunki 8 i 9 pokazują, jak zmiana sposobu realizacji reprodukcji wpływa na jakość uzyskiwanych rozwiązań.

5.4. Przykłady realizacji operatora mutacji

Pierwszy sposób realizacji operatora mutacji polegał na wybieraniu grupy studenckiej (rząd macierzy), następnie losowym wybieraniu zajęcia (gen lub k przyległych genów, gdzie k oznacza czas trwania zajęć) i wymienianiu ich z innymi zajęciami, dla których czas trwania oraz liczba terminów wolnych w otoczeniu nie był krótszy od czasu zajęć wylosowanych. Zajęcia mające status "ustalone" nie podlegały mutacji.

Druga zaimplementowana metoda przeprowadzenia mutacji polegała na usuwaniu z gotowego rozkładu zajęć losowo wybranych zajęć (z wyjątkiem zajęć o ustalonym terminie), a następnie ponownym losowym wstawianiu wyjętych zajęć do planu.

Zastosowanie drugiej z metod nie zmieniło w żaden sposób charakteru otrzymywanych rozwiązań. Ze względu jednak na swoją prostotę istotnie przyspieszyło opracowywany algorytm.

6. Podsumowanie

Osiągnięcie sukcesu przy stosowaniu algorytmów genetycznych do problemu układania rozkładu zajęć w dużej mierze zależy od doboru parametrów tego algorytmu, a mianowicie przyjętej postaci funkcji dopasowania, zdefiniowania operatorów genetycznych, jak również doboru prawdopodobieństw mutacji i krzyżowania oraz ilości generacji i liczności populacji. Wartości tych parametrów zostały dobrane doświadczalnie. Najlepsze wyniki osiągnięto w przypadku, gdy wartość kosztu pojedynczego konfliktu była stukrotnie większa od wartości kosztu pojedynczej nieciągłości w rozkładzie zajęć. Ponadto prawdopodobieństwo krzyżowania powinno wynosić 60-70%, a mutacji 20-30%. Ograniczenie liczby chromosomów w populacji do 100, a liczby generacji do 150 pozwalało na osiąganie zadowalających wyników. Przeprowadzone eksperymenty wykazały, że przyjęcie zbyt restrykcyjnych ograniczeń co do postaci chromosomu w fazie inicjalizacji znacznie wydłużało proces znajdowania optymalnych rozwiązań. Zastosowanie reprodukcji z wbudowanym mechanizmem zachowania najlepszego chromosomu dawało lepsze jakościowo rozwiązania. Ponadto przeprowadzone eksperymenty wykazały, że w przypadku omawianego problemu konieczne jest stosowanie mutacji oraz zdefiniowanie operatorów mutacji i krzyżowania w taki sposób, aby nie powodowały zmian wejściowego zbioru danych (operatory doskonałe).

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-Łódź 1986.

[4] Colorni A., Dorigo M., Maniezzo V.: Genetic Alghorithms: New Approach to the Timetable Problem. NATO ASI Series, vol. F 82, Combinatorial Optimization 1992.

[5] Goldberg D. E.: Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.

[6] Holland J.H.: Adaptation in natural and artificial systems, Ann Arbor: The University of Michigan Press, 1975.

[7] Colorni A. , Dorigo M. , Maniezzo V.: Genetic Alghorithms and highly constrained problems. Proceeding of the First International Workshop on Parallel Problem Solving from Nature, Dortmund, 1993.

[8] Colorni A., Dorigo M., Maniezzo V.: A Genetic Alghorithm to solve the timetable problem. Submited to computational Optimization and applications journal, 1994.

Recenzent: Dr Urszula Boryczka

Wpłynęło do Redakcji 3 grudnia 1996 r.

Abstract

The Genetic Algorithm is a model of machine learning which derives its behavior from a metaphor of the processes of evolution in nature. Genetic Algorithms are used for a number of multidimensional optimization problems. In practice we can implement this genetic model of computation by having arrays of bits or characters (alphabet) to represent the chromosomes. Simple bit manipulation operations allow the implementation of crossover, mutation and other genetic operations. Implemented Genetic Algorithm involves the following cycle: init the first population, evaluate the fitness of all of the chromosomes in the population, create a new population by performing reproduction, apply crossover and mutation.

The timetable problem (TTP) one of the NP-hard optimization problems [2,3] can be solved by genetic algorithm. Chapter 2 describes the primitives of this problem. Chapter 3 defines genetic operators and structure of the fitness function. One of the posible representation of the chromosome in timetable problem has been presented at fig.2. Next chapter explains in general genetic algorithm engine. Initialization process has been described next in detailes. Different probability of crossover and mutation has been tested (fig. 5, 6, ,7). Two methods of reproduction and mutation has been compared (fig. 8, 9). The best results have been obtain with genetic operators which preserves input data structure.