

Algorytmy Genetyczne

Przypisywanie do grup za pomocą algorytmu genetycznego

Idea

Bierzemy pod uwagę dwa warunki. Warunek braku kolizji pomiędzy terminami dla każdego studenta jest warunkiem twardym i każde rozwiązanie musi je spełniać. Warunek ilości osób w klasach jest warunkiem miękkim i bierze udział w wyliczaniu funkcji **fitness**.

Przyjęliśmy reprezentację przedmiotów jako listę list. Cała siatka godzin to macierz $S \times P$, gdzie S to liczba studentów, a P to liczba przedmiotów. Algorytm ma za zadanie znajdować jak najlepszą siatkę godzin wg przyjętej metryki opisanej jako funkcja **fitness** wykorzystując krzyżowanie, mutowanie.

Krzyżowanie

Proces krzyżowania zachowuje poprawność rozwiązania (jeśli chodzi o kolizje między terminami) i odbywa się wg schematu:

Dla każdego studenta:

1. Próbuje kilka razy znaleźć jakiś podział, że część przedmiotów bierzemy z **rodzic1**, a część z **rodzic2**, tak żeby nie było kolizji godzinowych dla tego studenta.
2. Jeśli nie ma takiego podziału po kilku próbach to wziąć w ogóle całego studenta z **rodzic1** albo z **rodzic2** losowo.

Mutowanie

Proces mutacji odbywa się z kolei wg schematu:

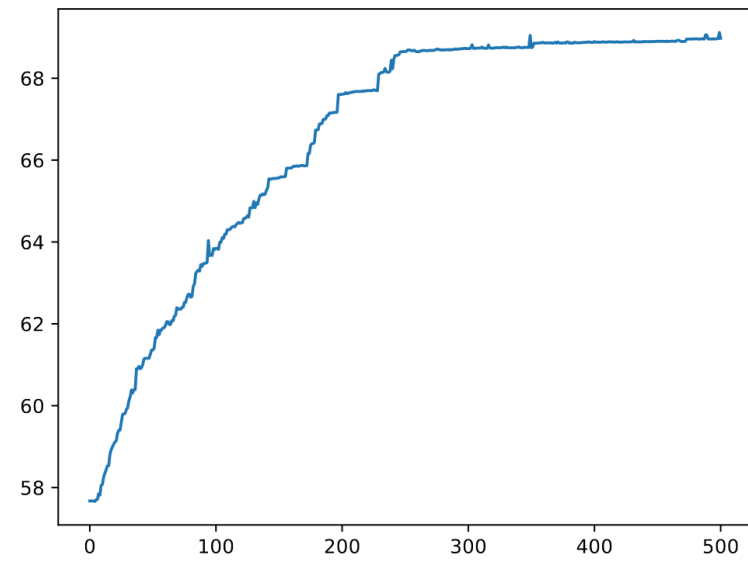
Dla jakiejś losowej części przedmiotów: bierzemy dwóch studentów i zamieniamy im terminy (oczywiście tylko jeśli nie ma kolizji).

Dopasowanie

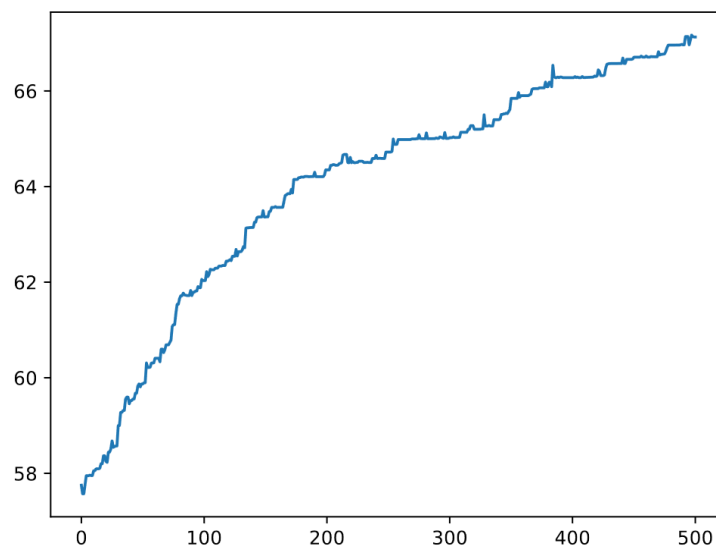
Funkcja fitness oblicza wskaźnik dopasowania danego rozwiązania, bierze ona pod uwagę przede wszystkim kwestię niepotrzebnych bloków przerw w planie zajęć - nikt nie lubi 5 godzinnych okienek między zajęciami, szczególnie jeżeli mieszka daleko. Problem jest o tyle skomplikowany, że nie jesteśmy w stanie w pełni usatysfakcjonować każdego studenta - często niemożliwym jest utworzenie dla każdego planu godzin bez okienek ze względu na ograniczenie do ilości osób w grupach. Funkcja fitness liczyła dla każdego studenta dopasowanie jako kwadrat sumy dopasowań w każdym dniu (które natomiast było liczone jako iloraz $\frac{\text{czas trwania zajęć}}{\text{czas trwania zajęć} + \text{okienka}}$), a następnie od sumy tych wartości odejmowaliśmy odpowiedni współczynnik związany z ograniczeniem ilości osób - suma wartości funkcji eksponencjalnych z modułu różnicy oczekiwanej ilości osób na zajęciach i jej rzeczywistej ilości.

Wyniki

Poniżej przedstawiamy wykresy wyników z niektórych z przeprowadzonych eksperymentów. Oś pionowa to wynik naszej funkcji **fitness**, a oś pozioma to iteracje. W obu przypadkach liczba iteracji wynosiła 500.



Prawdopodobieństwo mutacji: 15%



Prawdopodobieństwo mutacji: 5%

Wnioski

- Choć warunek liczby studentów w danym terminie jest warunkiem miękkim, to po niewielu iteracjach wszystkie rozwiązania go optymalizują, i maksymalnie 1 lub 2 studentów wychodzi ponad limit (co nie jest problemem, bo można go ustawić na trochę mniejszy od prawdziwego limitu).
- Z przeprowadzonych testów wynika, że mutacja na poziomie 15% daje lepsze wyniki od mutacji na poziomie 5%.