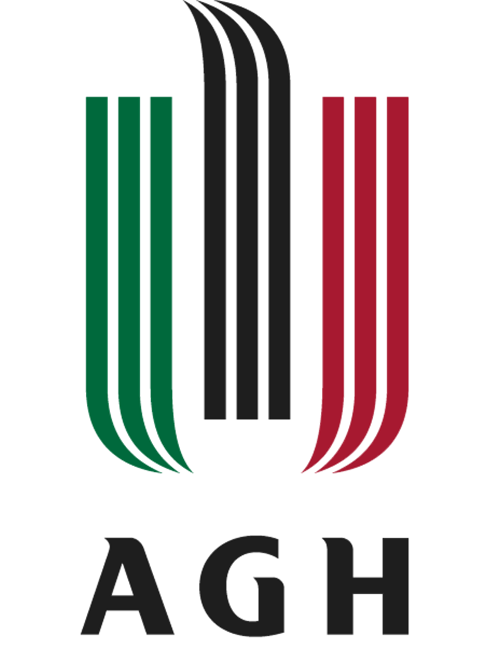
**Akademia Górniczo-Hutnicza**

**im. Stanisława Staszica w Krakowie**

Wydział Inżynierii Metali i Informatyki Przemysłowej



**Praca dyplomowa inżynierska**

„Generator planu zajęć lekcyjnych dla szkół”

Autor: Gabriel Rogala

Kierunek studiów: Informatyka stosowana

Opiekun pracy: prof. dr hab. inż. Jan Kusiak

Kraków, 2017

Uprzedzony o odpowiedzialności karnej na podstawie art. 115 ust. 1 i 2 ustawy z dnia   
4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.): „ Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór   
w wersji oryginalnej albo w postaci opracowania, artystyczne wykonanie albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, wideogram lub nadanie.”, a także uprzedzony o odpowiedzialności dyscyplinarnej na podstawie art. 211 ust. 1 ustawy z dnia   
27 lipca 2005 r. Prawo o szkolnictwie wyższym (t.j. Dz. U. z 2012 r. poz. 572, z późn. zm.) „Za naruszenie przepisów obowiązujących w uczelni oraz za czyny uchybiające godności studenta student ponosi odpowiedzialność dyscyplinarną przed komisją dyscyplinarną albo przed sądem koleżeńskim samorządu studenckiego, zwanym dalej „sądem koleżeńskim”, oświadczam, że niniejszą pracę dyplomową wykonałem(-am) osobiście, samodzielnie i że nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.

…….…………………………………

Spis treści :

1 Wstęp 3

1.1 Cel i założenia projektu 4

1.2 Struktura pracy 4

2 Zasada działania algorytmu genetycznego 5

2.1 Podstawowe pojęcia 5

2.2 Podstawowe operacje 6

2.2.1 Generowanie populacji 6

2.2.2 Krzyżowanie 7

2.2.3 Mutacja 7

2.2.4 Ocena populacji 8

2.2.5 Selekcja 8

2.3 Przebieg ewolucji 9

3 Specyfikacja Problemu 10

3.1 Cel problemu 10

3.2 Wymagania funkcjonalne 11

4 Projekt i implementacja 12

4.1 Architektura aplikacji 12

4.1.1 Baza danych 12

4.1.1.1 Opis tabel bazy danych 12

4.1.1.2 Diagram bazy danych 13

4.1.2 Algorytm 14

4.1.2.1 Diagram klas 14

4.1.2.2 Opis klas algorytmu 15

4.1.3 Interfejs użytkownika 17

4.2 Opis najważniejszych funkcji algorytmu genetycznego 21

4.2.1 Tworzenie osobnika 21

4.2.1.1 Funkcja findAndSetLessonsInBestPosition 22

4.2.1.2 Funkcja removeLessonAndFindNewPosition 23

4.2.2 Krzyżowanie 25

4.2.3 Mutacja 27

4.2.4 Ocena populacji 28

4.2.5 Selekcja 29

5 Testy 30

6 Podsumowanie 35

7 Literatura 36

# Wstęp

Szkoła jest jedną z instytucji z którą każdy z nas miał, bądź będzie miał, do czynienia przez około 12 lat naszego życia (nie wliczając w to szkoły wyższej). W główny skład każdej ze szkół wchodzą uczniowie oraz nauczyciele, których liczebność czasem przekracza tysiące osób. Każda z miejscowości posiada więcej niż jedna szkołę. Takich instytucji w całym naszym kraju jest około 20 tysięcy. Do prawidłowego prowadzenia zajęć wymagane są informacje o tym kiedy i gdzie odbywa się dana lekcja. Lekcja to interakcja nauczyciela z uczniami w określonym przedziale czasowym. Takie informacje zawarte są w planie zajęć. Plan zajęć jest jedną z najważniejszych rzeczy w szkole, ponieważ bez odpowiednio przygotowanego rozkładu zajęć prowadzenie lekcji w szkołach było by niemożliwe.

Każda ze szkół co pół roku potrzebuje nowego planu zajęć, którego ułożenie zajmuje nawet kilka tygodni. Rocznie ilość nowostworzonych planów osiąga 40 tysięcy. Każda osoba korzystała w swoim życiu z około 24 planów zajęć, na których ułożenie ktoś poświęcił wiele czasu.

Niniejsza praca inżynierska ma na celu stworzenie aplikacji, która będzie pozwalała wyręczyć ludzi w układaniu planu zajęć, co za tym idzie, zmniejszyć czas poświęcony na układanie planu zajęć z kilku tygodni do kilku minut (pomijając czas poświęcony na wprowadzenie danych do systemu). Do rozwiązania takiego problemu, jakim jest ułożenie planu, idealnie będzie nadawał się algorytm genetyczny, który dzięki specyficznemu sposobie działania, pozwoli wyewoluować planu na miarę naszych potrzeb. Dzięki zastosowaniu algorytmu genetycznego aplikacja nie musi znać dokładnego procesu układania planu zajęć, który jest dość skomplikowany, jedynie musi potrafić ocenić wygenerowany plan.

## Cel i założenia projektu

Celem niniejszej pracy dyplomowej jest stworzenie aplikacji generującej plan zajęć dla szkół. Aplikacja ma za zadanie wygenerowanie planu zajęć nie posiadającego żadnych defektów, takich jak:

* więcej niż jedna lekcja tej samej grupy w jednym slocie czasowym,
* brak lekcji w planie,
* lekcja w nieodpowiedniej sali lekcyjnej.

## Struktura pracy

W pierwszym rozdziale zawarto wstęp, cel i założenia projektu oraz strukturę pracy. W rozdziale drugim została opisana zasada działania algorytmu genetycznego oraz charakterystyczne, dla tego algorytmu, operacje dokonywane na obiektach. Trzeci rozdział z kolei opisuje specyfikacje problemu, fakty na które trzeba zwrócić uwagę podczas układania planu zajęć oraz wymagania funkcjonalne aplikacji. W kolejnym czwartym rozdziale została opisana architektura aplikacji, między innymi baza danych, interfejs użytkownika oraz to co najważniejsze czyli implementacja algorytmu genetycznego. Piąty rozdział jest poświęcony testom generatora dla średniego planu zajęć szkoły średniej. Test obejmuje kilka różnych konfiguracji algorytmu. Ostatnie dwa rozdziały to kolejno podsumowanie oraz literatura.

# Zasada działania algorytmu genetycznego

Algorytm genetyczny (AG) jest jedną z wielu metod zaliczanych do nurtu ewolucyjnego. Wykorzystuje się go głównie do znalezienia optymalnego rozwiązania danego problemu. Algorytm ten przypomina proces ewolucji z biologicznego punktu widzenia, ponieważ dany proces nie polega na stworzeniu właściwego rozwiązania (wykorzystując przy tym szeregu, często bardzo skomplikowanych, operacji), lecz na „wyhodowaniu” go z pośród wielu alternatywnych rozwiązań. Taki proces polega na porównywaniu ze sobą wielu rozwiązań, które poddawane są niewielkim zmianą, w celu stworzenia coraz to lepszego rozwiązania.

## Podstawowe pojęcia

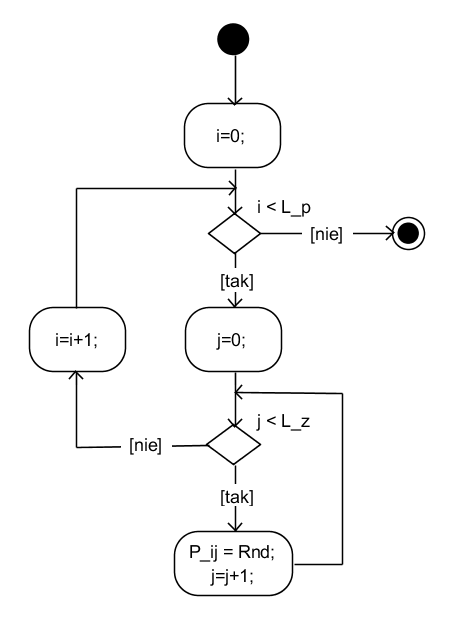
**Osobnik** jest to pojedynczy obiekt podlegający ewolucji. Celem ewolucji jest jak najlepsze przystosowanie osobnika do otaczającego go środowiska. W przypadku AG osobnikiem możemy nazwać rozwiązanie danego problemu.

**Populacja** jest to zbiór osobników o stałym rozmiarze, która w kolejnych cyklach ewolucji zastępuje obecne osobniki nowo stworzonymi.

## Podstawowe operacje

### Generowanie populacji

Generowanie populacji jest to proces polegający na wygenerowaniu danej ilości osobników danej populacji w sposób losowy bądź deterministyczny.



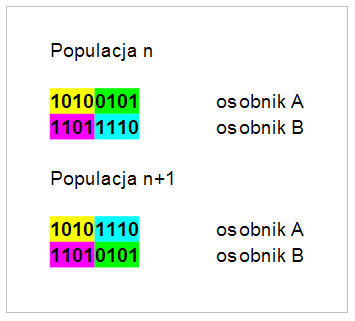
Rysunek Diagram aktywności - generowanie populacji

Gdzie:

* L\_p – ilość osobników w populacji
* L\_z – ilość zmiennych w osobniku
* Rnd – losowa wartość zmiennej
* P – populacja
* P\_i – osobnika
* P\_ij – zmienna

### Krzyżowanie

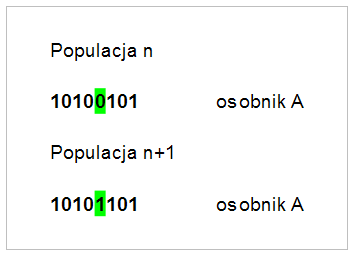
Krzyżowanie to proces polegający na stworzeniu nowego osobnika bądź nowych dwóch osobników, wykorzystując do tego dwóch innych osobników. Metod krzyżowania jest bardzo wiele. Jedną ze standardowych oraz najmniej skomplikowanych jest krzyżowanie jednopunktowe (1-point Crossover). Przykładowa metoda krzyżowania polega na wylosowaniu punktu krzyżowania. Tworzenie nowego osobnika polega na skopiowaniu danych do punktu krzyżowania od rodzica A, resztę od rodzica B.



Rysunek Algorytm genetyczny - krzyżowanie

### Mutacja

Mutacja to proces polegający na dokonaniu niewielkich zmian na istniejącym osobniku. Metod mutacji jest bardzo wiele. Jedną ze standardowych oraz najmniej skomplikowanych odwrócenia bitu (Invert a Bit). Przykładowa metoda mutacji polega na wylosowaniu punktu mutacji w którym będzie dokonana operacja zanegowania bitu.



Rysunek Algorytm genetyczny - mutacja

### Ocena populacji

Ocena populacji jest to indywidualny stopień przystosowania każdego z osobników w populacji. Za ocenę odpowiada funkcja przystosowania, która podczas oceny bierze pod uwagę kilka czynników.

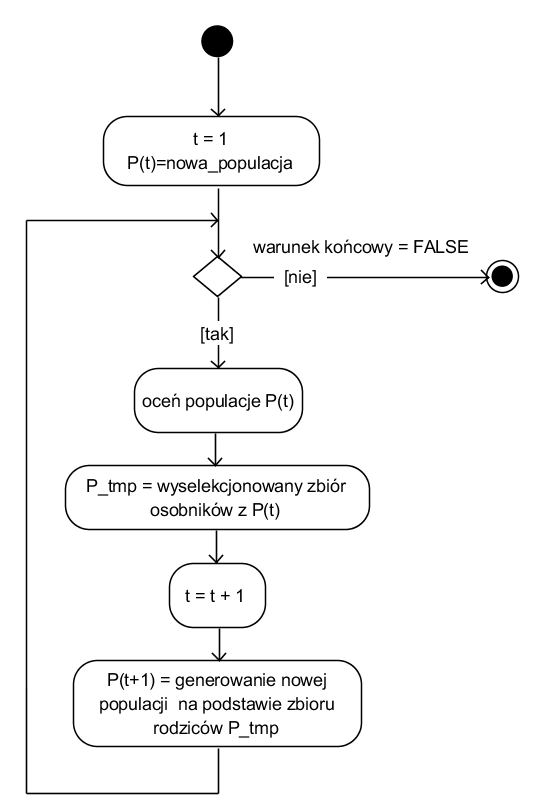
### Selekcja

Selekcja jest to proces tworzenia nowej populacji z obecnie istniejącej, wykorzystując do tego proces mutacji oraz krzyżowania. Każdą populacje można podzielić na osobników silnie, średnio i słabo przystosowanych. W przypadku AG taki podział jest istotny, ponieważ przynależność osobnika do danej grupy określa jego stopień przydatności do powstania nowej populacji.

Wykorzystanie osobników z grupy:

1. Silnie przystosowani:
   * reprodukcja,
   * krzyżowanie z osobnikami ślinie i średnio przystosowanymi,
   * mutacja.
2. Średnio przystosowani:
   * krzyżowanie z osobnikami ślinie i średnio przystosowanymi,
   * mutacja.
3. Słabo przystosowani:
   * porzucenie.

## Przebieg ewolucji



Rysunek Diagram aktywności - przebieg ewolucji

Gdzie:

* t – krok czasowy (epoka)
* P – populacja
* P\_tmp – lista wyselekcjonowanych osobników
* warunek końcowy – warunek ograniczający czas trwania ewolucji

# Specyfikacja Problemu

## Cel problemu

Podczas ręcznego układania planu zajęć, osoba układająca najczęściej stara się ułożyć plan w taki sposób aby posiadał:

* minimalna ilość okienek w planie zajęć grup,
* określona oraz minimalna ilość okienek w planie zajęć nauczycieli,
* minimalna ilość lekcji tego samego przedmiotu w dniu tygodnia,
* równomierny rozkład zajęć różnego typu w dniu tygodnia,
* równomierny rozkład okienek w planach zajęć nauczycieli (minimalna ilość nauczycieli posiadających okienko w tym samym slocie czasowym),
* minimalna ilość lekcji o wysokim priorytecie w ostatnich godzinach dni.

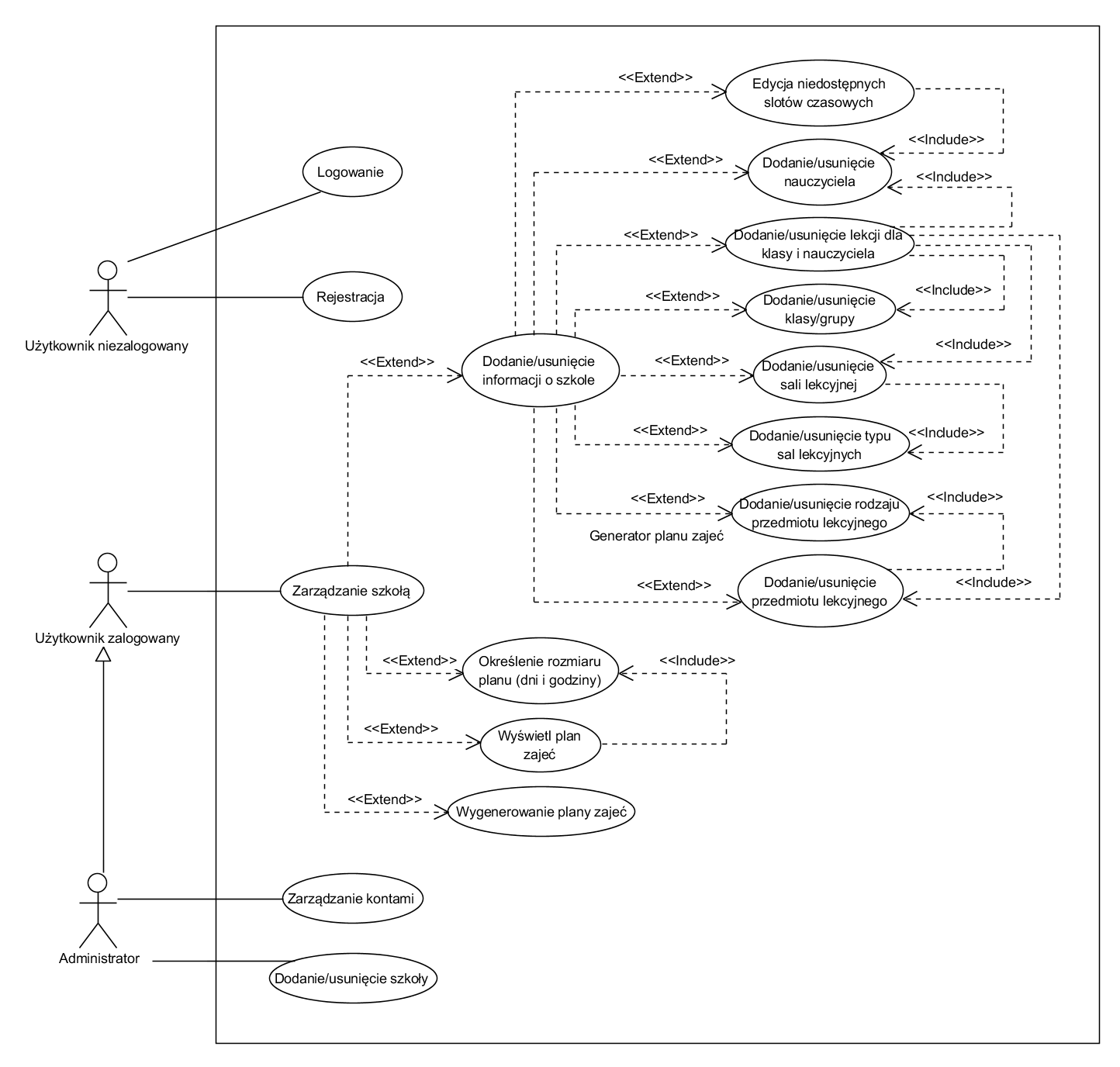
Aplikacja ma za zadanie wygenerowanie planu zajęć składającego się z planów klas, nauczycieli oraz sal lekcyjnych. Wygenerowany plan nie może posiadać defektów takich jak:

* więcej niż jednej lekcji tej samej grupy w jednym slocie czasowym,
* braku lekcji w planie,
* lekcji w nieodpowiedniej sali lekcyjnej (zły typ sali oraz niewystarczająca ilość miejsc).

Głównym celem jest zoptymalizowanie planu pod względem:

* minimalna ilości wolnych slotów czasowych pomiędzy zajęciami na planie nauczycieli i grup,
* równomierne rozłożenie lekcji różnego typu w dniu,
* minimalna ilość powtórzeń tych samych lekcji w jednym dniu,

## Wymagania funkcjonalne



Rysunek Diagram przypadków użycia

# Projekt i implementacja

Analiza możliwych sposobów implementacji wykazała, że najlepszym rozwiązaniem takiego problemu, będzie zaprojektowanie aplikacji jako aplikacja internetowa. Do implementacji został wykorzystany język C# w oparciu o Framework ASP.NET MVC.

## Architektura aplikacji

### Baza danych

#### Opis tabel bazy danych

**AspNetUser** – informacje dotyczące konta użytkownika,

**Groups** – informacje dotyczące grupy, ilość uczniów, kategoria podziału na podgrupy, niedostępne godziny w tygodniu,

**Lessons** – informacje dotyczące lekcji, nauczyciel prowadzący, grupa, rodzaj sali lekcyjnej, przedmiot szkolny, rozkład zajęć,

**Rooms** – informacje dotyczące sali lekcyjnej, ilość miejsc, rodzaj sali, niedostępne godziny w tygodniu,

**RoomTypes** – informacje dotyczące rodzaju sal lekcyjnych,

**Schools** – informacje dotyczące szkoły oraz rozmiaru rozkładu godzin w tygodniu, przypisanie szkoły do konta użytkownika,

**STGConfig** – informacje dotyczące konfiguracji algorytmu generującego plan zajęć,

**SubGroupTypes** – informacje dotyczące kategorii podgrupy (np.: podział klasy na dwie podgrupy na zajęciach z wychowania fizycznego),

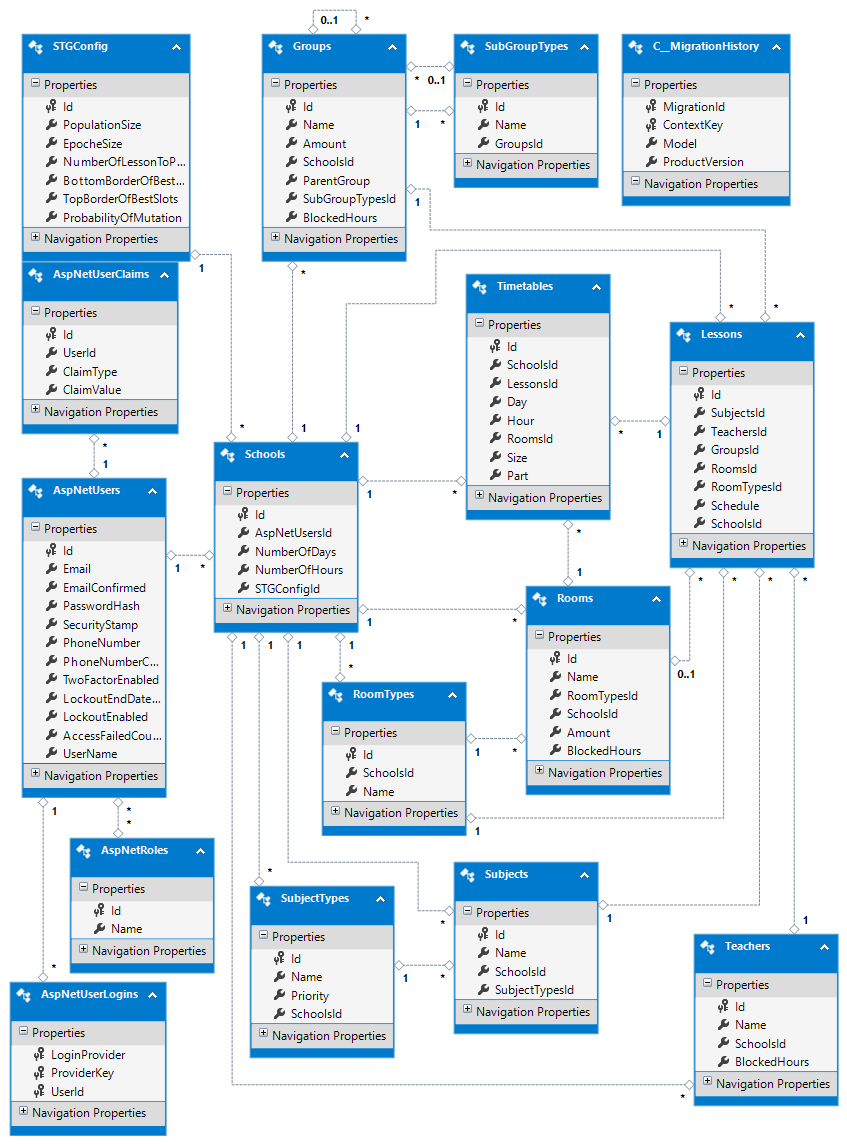
**Subjects** – informacje dotyczące przedmiotu szkolnego, rodzaju przedmiotu szkolnego,

**SubjectTypes** – informacje dotyczące rodzaju przedmiotu szkolnego,

**Teachers** – informacje dotyczące nauczyciela, niedostępne godziny w tygodniu,

**Timetables** – informacje dotyczące planu zajęć, dnia, godziny, lekcji, sali lekcyjnej, długości lekcji,

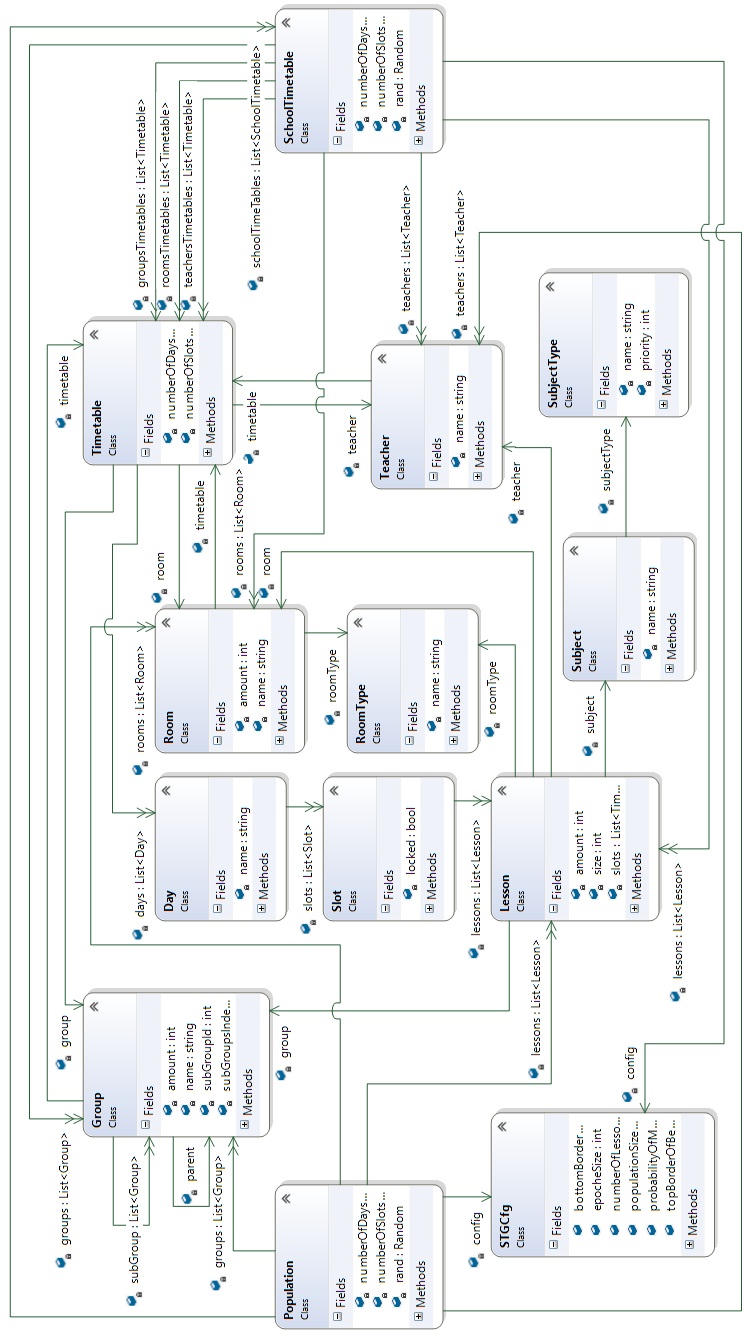
#### Diagram bazy danych



Rysunek Diagram ERD bazy danych

### Algorytm

#### Diagram klas



Rysunek Diagram klas algorytmu

#### Opis klas algorytmu

Klasa **Day** przechowuje listę slotów czasowych danego dnia.

Klasa **Group** przechowuje informacje o klasie/grupie takie jak nazwa, ilość uczniów, identyfikator grupy bazowej (w przypadku gdy opisuje ona podgrupę), listę identyfikatorów podgrup (w przypadku gdy opisuje ona grupę bazową), identyfikator kategorii podgrupy oraz odnośnik do planu zajęć.

Opis najważniejszych funkcji:

* **getSubGroupFreeSlotToLesson** - odpowiada za wygenerowanie list slotów czasowych w których jest możliwość wstawienia przekazywanej lekcji w każdej z podgrup.

Klasa **Lesson** przechowuje informacje o lekcji takie jak identyfikator przedmiotu szkolnego, identyfikator grupy, identyfikator nauczyciela prowadzonego daną lekcje, typ sali lekcyjnej w której powinna odbyć się dana lekcja, identyfikator sali lekcyjnej w której będzie odbywała się dana lekcja, listę slotów czasowych w których znajduję się dana lekcja, rozmiar danej lekcji, ilość takich samych lekcji w planie zajęć.

Opis najważniejszych funkcji:

* **isDifferent** – zwraca informacje czy dana lekcja jest tego samego przedmiotu szkolnego co lekcja przekazywana.

Klasa **Population** odpowiada za przeprowadzenie całego procesu generowania planu zajęć. Przechowuje listę osobników (planów zajęć dla szkoły), rozmiar planu zajęć (ilość dni i godzin w tygodniu), konfigurację dla procesu generowania planu zajęć oraz listy lekcji, nauczycieli, grup, sal lekcyjnych dla danej szkoły.

Opis najważniejszych funkcji:

* **generatePopulation** – generuje populacje początkową,
* **generateNewPopulation** – generowanie nowej populacji na podstawie obecnie istniejącej,
* **start** – rozpoczęcie procesu generowania planu zajęć.

Klasa **Room** przechowuje informacje o sali lekcyjnej takie jak nazwa, ilość miejsc, typ sali lekcyjnej oraz odnośnik do planu zajęć.

Klasa **RoomType** przechowuje informacje o typie sali lekcyjnej.

Klasa **SchoolTimetable** jest najbardziej rozbudowaną klasą ponieważ odpowiada za wygenerowanie, mutacje oraz krzyżowanie planu zajęć

Opis najważniejszych funkcji:

* **genereteTimetable** – generowanie planu zajęć,
* **findAndSetLessonsInBestPosition** – wstawianie lekcji w najlepszy slot,
* **getBestTimeSlot** – znalezienie najlepszego slotu do wstawienia danej lekcji,
* **findDifferentSubjectTheSameGroup** – znajduje lekcje z różnych przedmiotów tej samej grupy,
* **removeLessonsAndFindNewPosition** – zwalnia zajęty slot,
* **sortLessons** – sortuje lekcje pod względem podgrup, długości trwania lekcji, ilość lekcji w tygodniu, typu przedmiotu szkolnego,
* **fitness** – zwraca sumaryczny stopień przystosowania planów zajęć nauczycieli i grup,
* **crossover** – przeprowadza proces krzyżowania,
* **mutate** – przeprowadza proces mutacji,
* **isCorrect** – sprawdza czy wszystkie lekcje zostały wstawione do planu i nie ma żadnych powtórzeń.

Klasa **Slot** przechowuje informacje o slocie czasowym takie jak blokada slotu oraz lista lekcji występujących w danym slocie czasowym.

Klasa **STGCfg** przechowuje informacje o konfiguracji procesu generowania plany zajęć takie jak rozmiar populacji, ilość epok, ilość lekcji do jednoczesnego wstawienia, dolna i górna granica przedziału godzin do którego lekcje mają być wstawiane w pierwszej kolejności oraz prawdopodobieństwo do zajścia mutacji.

Klasa **Subject** przechowuje informacje o przedmiocie szkolnym takie jak nazwa oraz typ przedmiotu szkolnego.

Klasa **SubjectType** przechowuje informacje o typie przedmiotów szkolnych takie jak nazwa oraz priorytet.

Klasa **Teacher** przechowuje informacje o nauczycielu takie jak nazwa oraz odnośnik do planu zajęć.

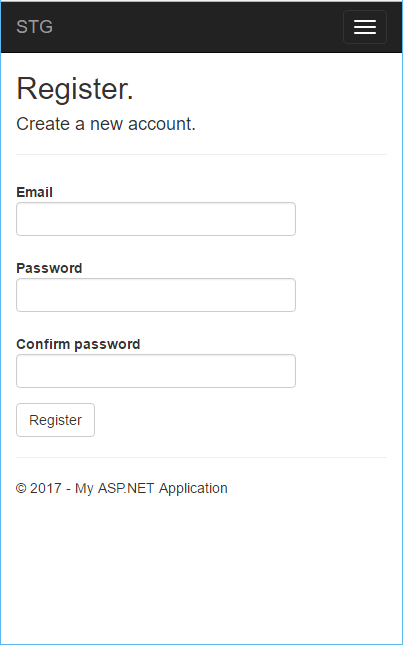
Klasa **Timetable** przechowuje informacje o planie zajęć takie jak lista dni, odnośnik do grupy, nauczyciela bądź sali lekcyjnej do której dany plan należy.

Opis najważniejszych funkcji:

* **fitness** – zwraca sumaryczny stopień przystosowania planu zajęć,
* **fitnessFreeSlots** – zwraca stopień przystosowania pod względem ilości wolnych slotów pomiędzy lekcjami,
* **fitnessType** – zwraca stopień przystosowania pod względem rozkładu rodzajów zajęć w dniu,
* **fitnessSubject** – zwraca stopień przystosowania pod względem ilości wystąpień tej samej lekcji w jednym dniu,
* **getFreeSlotsToLesson -** odpowiada za wygenerowanie list slotów czasowych w których jest możliwość wstawienia przekazywanej lekcji.

### Interfejs użytkownika

Aplikacja do działania wymaga założenia konta do którego zostaje przypisana szkoła.

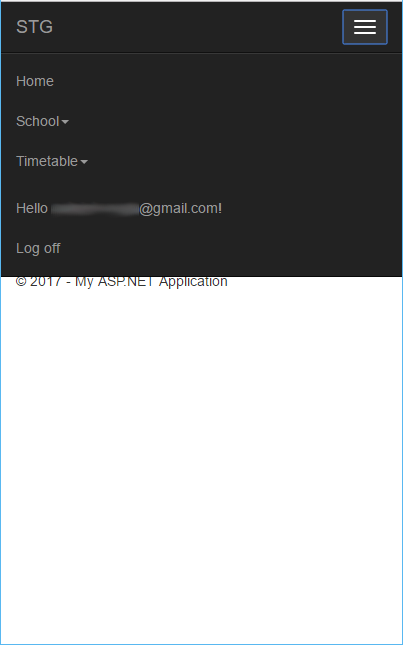


Rysunek Aplikacja - panel rejestracji

Po zalogowaniu użytkownik ma możliwość wprowadzenia informacji potrzebnych do stworzenia plany zajęć, takich jak:

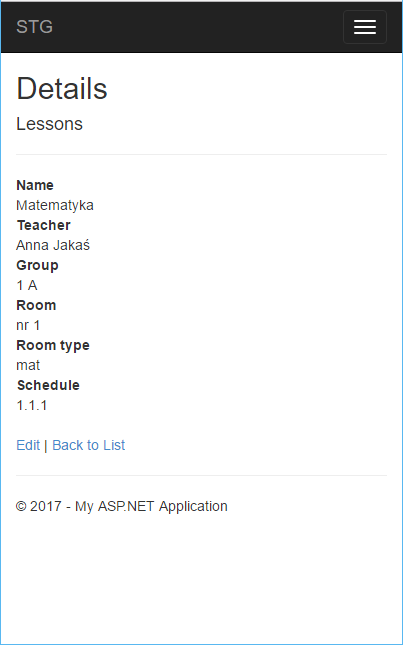
* Grupy,
* Rodzaje podgrup,
* Nauczyciele,
* Sale lekcyjne,
* Rodzaje sal lekcyjnych,
* Przedmioty szkolne,
* Lekcje dla grup.

Wszystkie informacje potrzebne do dodania lekcji znajdują się w menu pod przyciskiem „School”. Informacje odnośnie lekcji oraz widok całego planu zajęć znajdują się pod przyciskiem „Timetable”.



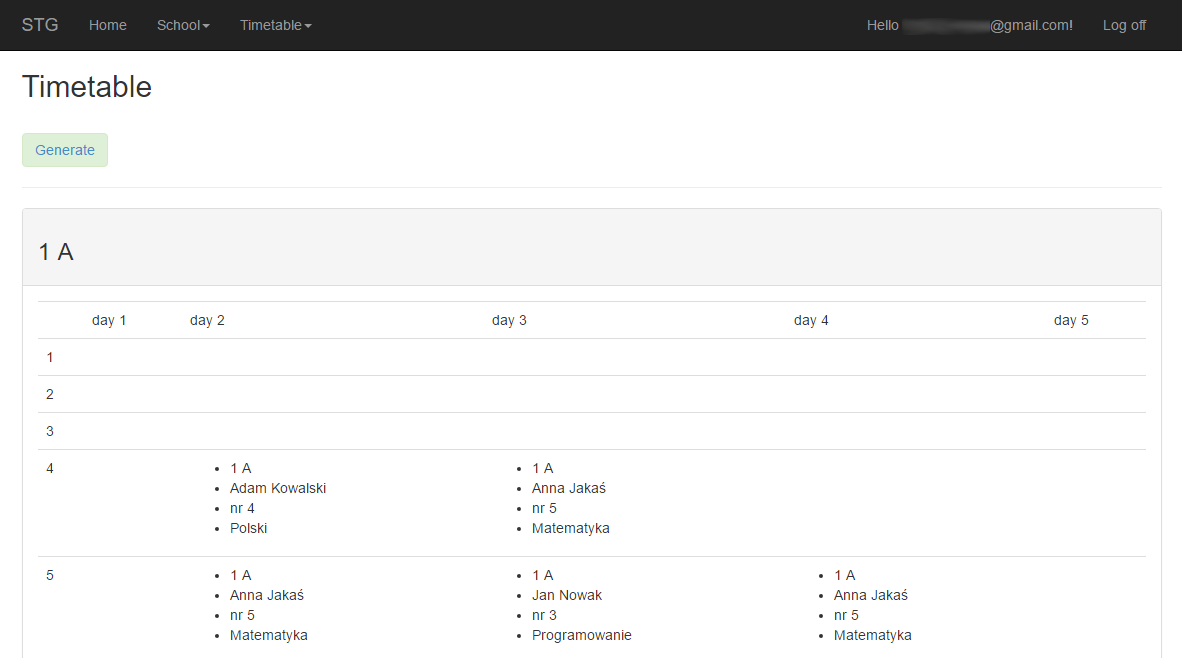
Rysunek Aplikacja - menu

Dodanie lekcji wymaga podania rozkładu zajęć, identyfikatora przedmiotu szkolnego, nauczyciela, grupy oraz typy sali lekcyjnej. Rozkład zajęć należy wprowadzać w następujący sposób. Długości trwania zajęć należy podać kolejno oddzielając je kropkami (np.: „2.1.1” - jedna lekcja dwu godzinna oraz dwie lekcje jednogodzinne).



Rysunek Aplikacja - panel ze szczegółowymi danymi lekcji

Chcąc wygenerować plan zajęć należy wejść w plan zajęć („Timetable” > „Timetables”) a następnie kliknąć przycisk . Po wygenerowaniu zostaje wyświetlony plan zajęć. Plan składa się z listy grup, nauczycieli oraz sal lekcyjnych. W każdym z planów wyświetlane są informacje o grupie, nauczycielu, sali lekcyjnej oraz lekcji w każdym ze slotów.

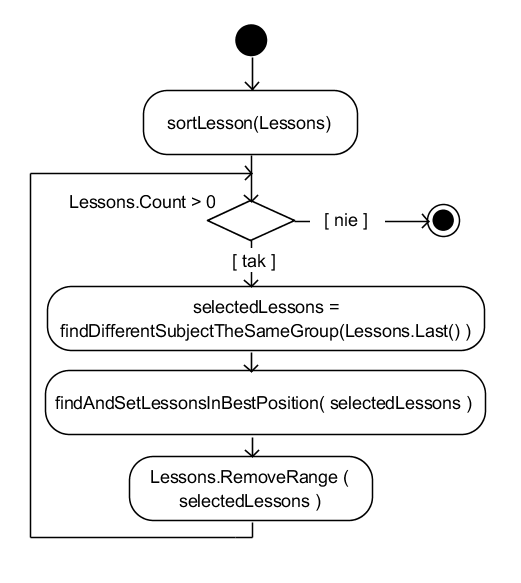


Rysunek Aplikacja - plan zajęć

## Opis najważniejszych funkcji algorytmu genetycznego

### Tworzenie osobnika

Proces tworzenia osobnika polega na wstawianiu do planu jednocześnie określonej ilości lekcji tej samej grupy. Poniżej został przedstawiony przepływ aktywności procesu generowania planu zajęć.



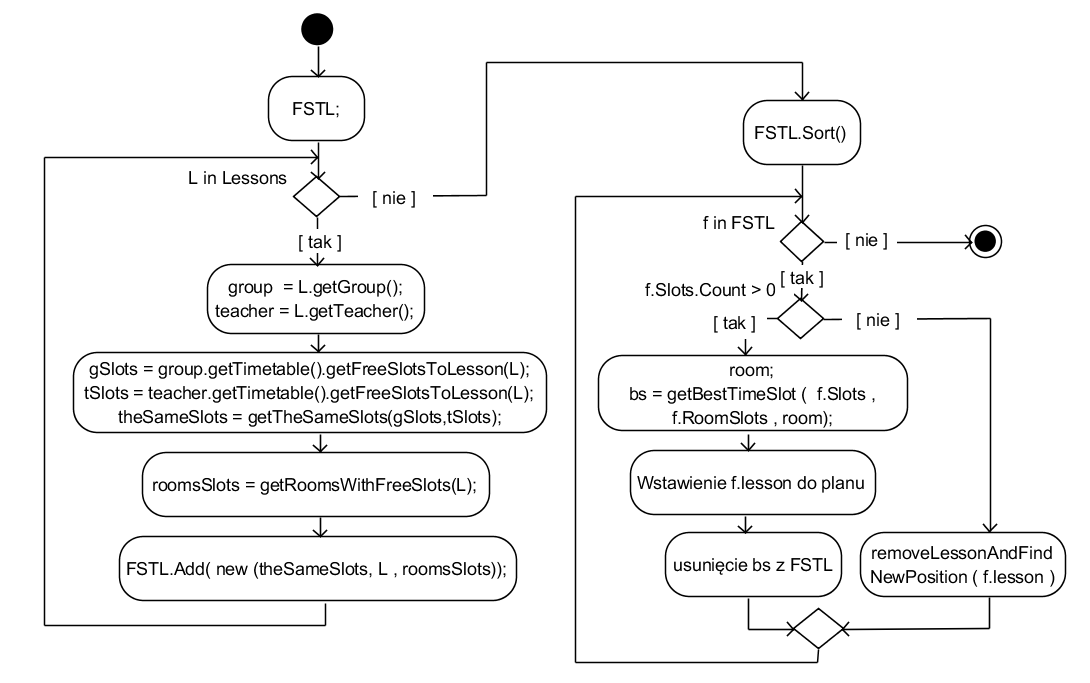
Rysunek Diagram aktywności - tworzenie osobnika

Gdzie:

* Lessons – lista wszystkich lekcji do wstawienia,
* selectedLessons – lista lekcji tej samej grupy.

#### Funkcja findAndSetLessonsInBestPosition

Funkcja ma za zadanie wstawienie każdą z lekcji w znaleziony slot oraz przydzielenie jej odpowiedniej sali.



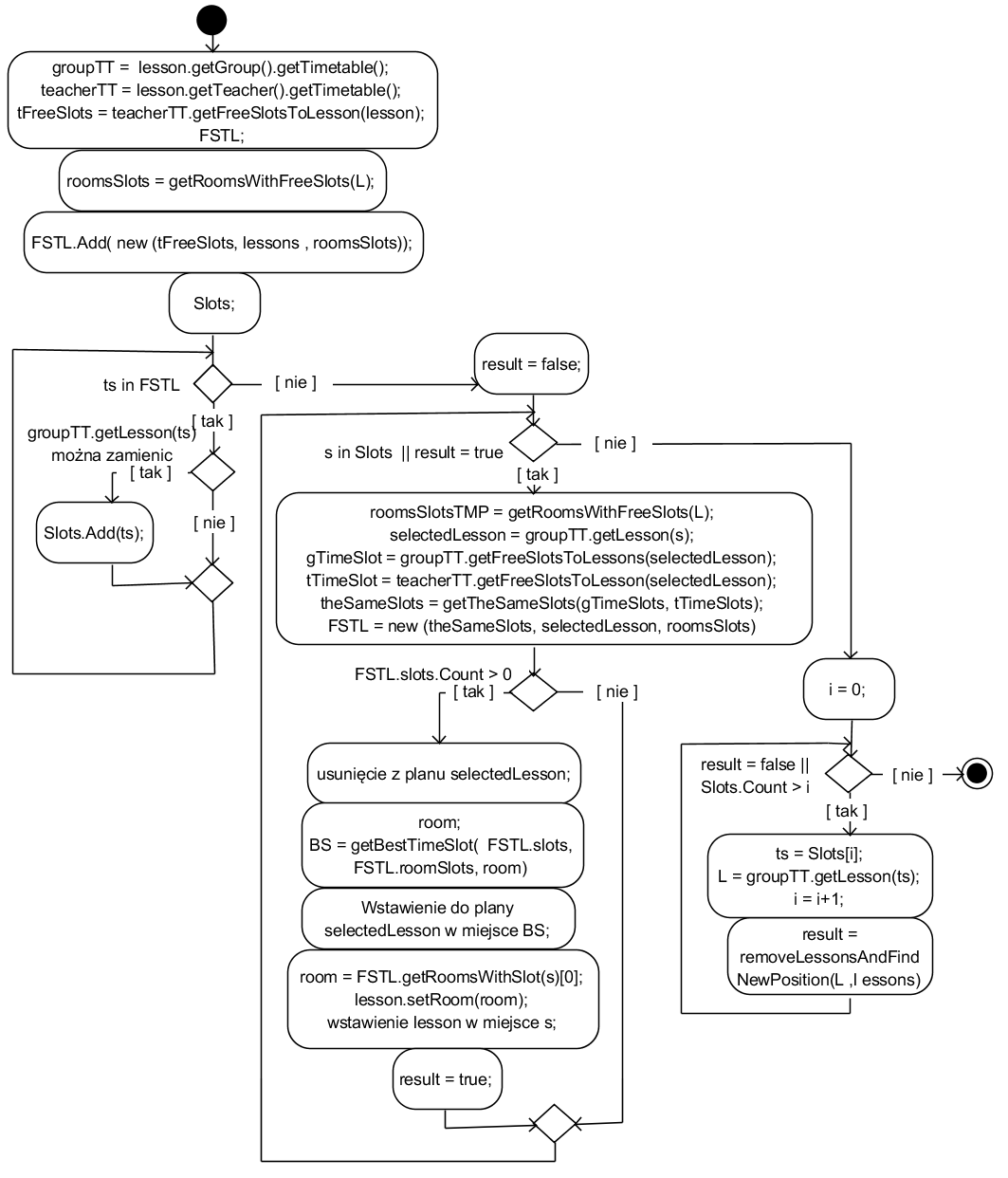
Rysunek Diagram aktywności - funkcja findAndSetLessonsInBestPosition

Gdzie:

* FSTL – Lista obiektów posiadających informacje o lekcji, wolnych slotach, wolnych slotach sal lekcyjnych oraz dostępnych salach,
* group – grupa która ma mieć daną lekcje,
* teacher – nauczyciel prowadzący daną lekcje,
* gSlots – lista wolnych slotów w planie grupy,
* tSlots – lista wolnych slotów w planie nauczyciela,
* roomsSlots – lista obiektów posiadających informacje o sali lekcyjnej i wolnych slotach danej sali,
* room – sala lekcyjna wybrana dla danej lekcji,
* bs – slot wybrany dla danej lekcji,
* Lessons – lista lekcji przekazanych do wstawienia.

#### Funkcja removeLessonAndFindNewPosition

Funkcja ma za zadanie zwolnienie miejsca dla danej lekcji (nie posiadającej wolnego slotu czasowego, w którym można ją umieścić). Poniżej został przedstawiony przepływ aktywności procesu zwalniania slotu.



Rysunek Diagram aktywności - funkcja removeLessonAndFindNewPosition

Gdzie :

* FSTL – Lista obiektów posiadających informacje o lekcji, wolnych slotach, wolnych slotach sal lekcyjnych oraz dostępnych salach,
* groupTT - plan zajęć grupy,
* teacherTT – plan zajęć nauczyciela,
* tFreeSlots – wolne sloty nauczyciela,
* roomsSlots – lista obiektów posiadających informacje o sali lekcyjnej i wolnych slotach danej sali,
* Slots – lista slotów z których można usunąć lekcję,
* selectedLesson – lekcja wybrana do przestawienia,
* BS – slot czasowy do którego zostaje wstawiona lekcja selectedLesson,
* room – sala lekcyjna do wstawienia lekcji,
* L – lekcja wybrana do wstawienia w nowe miejsce.

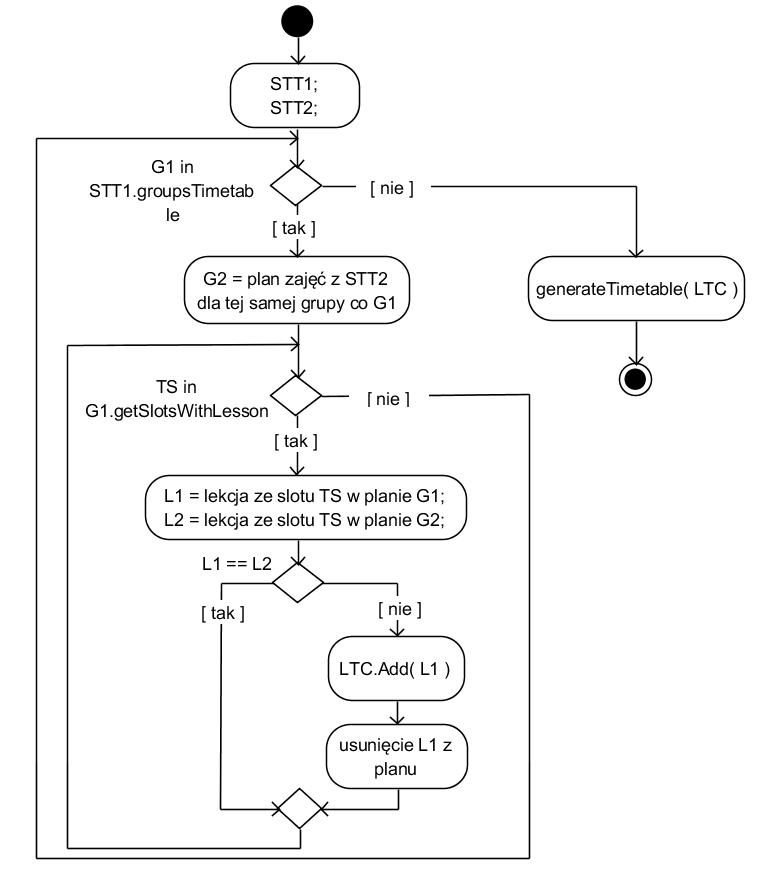
Podczas przeszukiwania brane są pod uwagę jedynie te lekcje, których zmiana miejsca zwolni potrzebny slot. Dla każdej lekcji szukane jest miejsce do którego można ją wstawić, jeżeli istnieje takie miejsce, lekcja zostaje przeniesiona. W przypadku gdy żadna z lekcji nie posiada alternatywnego miejsca, zostaje uruchomiony rekurencyjnie proces zwalniania miejsca dla jednej ze znalezionych lekcji.

### Krzyżowanie

Proces krzyżowania polega na porównaniu dwóch osobników i zachowaniu tych lekcji które występują u oby osobników w tych samych slotach czasowych. Do porównania dopuszczane są jedynie lekcje posiadające odpowiednie cechy, takie jak:

* długość lekcji równą 1,
* jedynie lekcję dla całej klasy.

Poniżej został przedstawiony przepływ aktywności procesu krzyżowania.



Rysunek Diagram aktywności - krzyżowanie

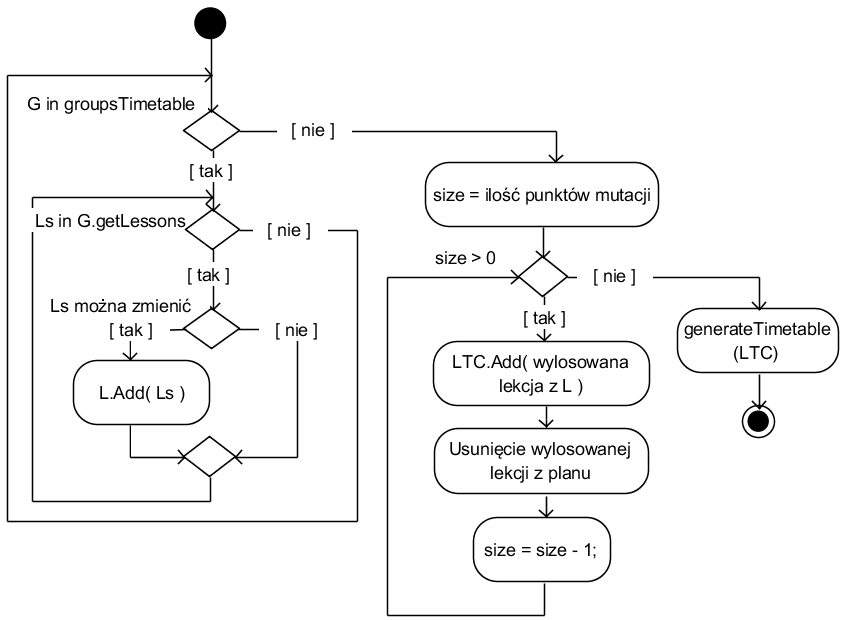
Gdzie:

* STT1 – wzorcowy plan zajęć,
* G1 – plan zajęć grupy z STT1,
* STT2 – plan zajęć do porównania,
* G2 – plan zajęć grupy z STT2,
* TS – slot czasowy (dzień i godzina),
* LTC – lista lekcji, które zostaną na nowo wstawione do planu.

Plan zajęć, na którym dokonywany jest proces krzyżowania, ustawia lekcje tak jak w planie wzorcowym, czyli staję się kopią. Przeszukiwanie polega na przejściu po wszystkich slotach wszystkich planów zajęć grup i usunięciu z planu zajęć tych lekcji, które nie występują w tych samych slotach czasowych w oby planach. Usunięte lekcje zostają na nowo wstawione do planu przy pomocy metody generującej plan zajęć.

### Mutacja

Proces mutacji polega na ponownym wstawieniu określonej ilości lekcji. Poniżej został przedstawiony przepływ aktywności procesu mutacji.



Rysunek Diagram aktywności - mutacja

Gdzie:

* G – plan zajęć grupy,
* Ls – lekcja z planu zajęć G,
* L – lista lekcji, które można poddać zamianie,
* LTC – lista lekcji, które zostaną poddane ponownemu wstawieniu.

W początkowej fazie zostają sprawdzone wszystkie lekcje z wszystkich planów zajęć grup, po to by wyselekcjonować lekcje, które mają największą szanse na znalezienie nowego miejsca. Takie lekcje powinny posiadać odpowiednie cechy, takie jak:

* długość lekcji równą 1,
* jedynie lekcję dla całej klasy.

W kolejnej fazie zostaje wylosowana określona ilość lekcji, która zostaje usunięta z planu zajęć. Ostatnim krokiem jest ponowne wstawienie lekcji do planu przy pomocy metody generującej plan zajęć.

### Ocena osobnika

Ocena osobnika składa się z sumy ocen wszystkich rozkładów zajęć. Na ocenę rozkładu zajęć wpływa:

1. Odchylenie ilości zajęć danego typu w danym dniu, od średniej ilości zajęć tego typu w tygodniu. Odchylenie jest czynnikiem o najmniejszym wpływie na ocenę rozkładu. Chcąc podkreślić istotność odchyleń zastosowano poniższy wzór, w postaci funkcji liniowej, do wyliczenia oceny.
2. Okienko (wolne sloty między zajęciami) jest czynnikiem o większym wpływie na ocenę niż odchylenie. Chcąc uwzględnić fakt iż bardziej negatywny wpływ na ocena mają dłuższe okienka niż większa ich ilość zastosowano poniższy wzór, w postaci funkcji kwadratowej. Zwiększenie wartości funkcji dwukrotnie pozwala zwiększyć istotność dla pojedynczego okienka w porównaniu do odchylenia
3. Powtórzenie lekcji w jednym dniu jest czynnikiem o największym wpływie na ocenę rozkładu, ponieważ lekcje danego przedmiotu powinny być równomiernie rozłożone w ciągu tygodnia. Chcąc podkreślić istotność powtórzeń zastosowano poniższy wzór, w postaci funkcji wykładniczej, do wyliczenia oceny.

Najważniejszym z czynników wpływających na dobrze ułożony plan jest brak powtórzeń tej samej lekcji w danym dniu, dlatego takie wystąpienie jest najwyżej oceniane.

Nie każdy plan zajęć oceniany jest pod względem wszystkich czynników. Jedynie w planie dla grup interesuje nas minimalna ilość lekcji tego samego przedmiotu w jednym dniu tygodnia oraz równomierny rozkład zajęć różnego typu w jednym dniu tygodnia. W przypadku planu zajęć nauczycieli interesuje nas jedynie ilość wolnych godzin między zajęciami. Rozkład zajęć dla sali lekcyjnej nie jest oceniany.

### Selekcja

Proces selekcji polega na podzieleniu populacji na trzy grupy:

* Silnie przystosowanych 20%,
* Średnio przystosowanych 60%,
* Słabo przystosowanych 20%.

Osobnicy silnie przystosowani są reprodukowani oraz poddawani możliwości mutacji. Osobnicy średnio przystosowani są krzyżowani z losowym osobnikiem silnie przystosowanym oraz poddani możliwości mutacji. Osobnicy słabo przystosowani są tworzeni na nowo.

# Testy

Do testów została przygotowana szkoła składająca się z 9 klas, 16 nauczycieli i 17 sal lekcyjnych. Każda z grup posiada 2 podgrupy z zajęć wychowania fizycznego. Na plan zajęć szkoły składa się 252 lekcji. Na każdą z klasy przypada średnio 28 lekcji na tydzień czyli około 6 lekcji w dniu.

Test został przeprowadzony dla konfiguracji:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| L. p. | Ilość epok | Wielkość populacji | Prawdopodobieństwo mutacji | Ilość lekcji do wstawienia |
| 1 | 100 | 100 | 5 | 5 |
| 2 | 500 | 100 | 5 | 5 |
| 3 | 100 | 100 | 25 | 5 |
| 4 | 100 | 100 | 5 | 1 |

Z wyników przeprowadzonych testów można zaobserwować fakt, iż w większości przypadków wartość minimalna ocen w epoce maleje w sposób logarytmiczny wraz z kolejnym krokiem czasowym. Najlepsze wyniki otrzymały plany zajęć wygenerowane z konfiguracjami 2 i 3. Największy wpływ na polepszenie oceny planu ma odpowiednio długi proces generacji oraz odpowiednio wysokie prawdopodobieństwo mutacji. Dzięki blokadzie niepozwalającej wygenerować gorszego planu, w procesie mutacji, funkcja minimum jest malejąca w całym przedziale.

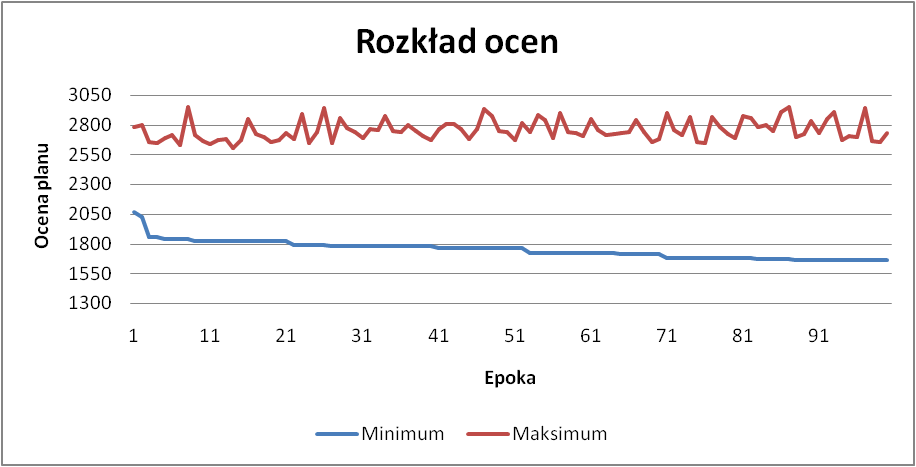
Poniżej zostały przedstawione szczegółowe wyniki testów dla danych konfiguracji.

1. Konfiguracja nr 1

Czas generowania planu zajęć: 03:53,3 min

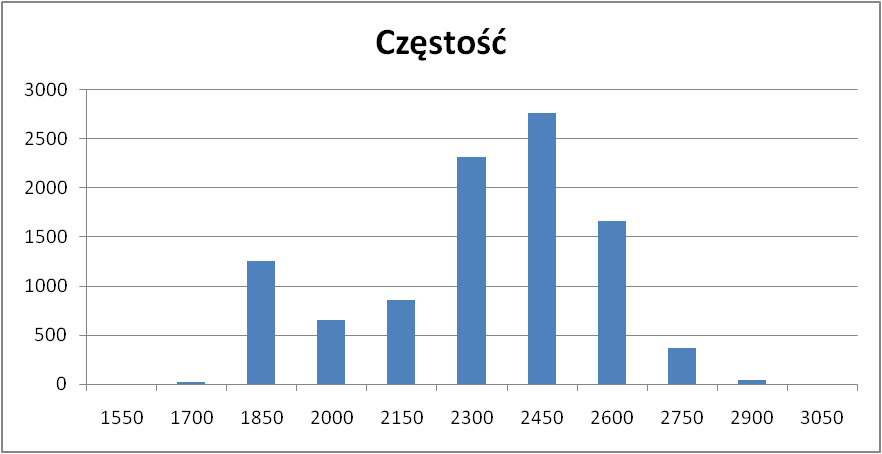
Ocena funkcji przystosowania planu wynikowego: 1660

Rozkład ocen planów podczas procesu generowania:



Rysunek Wykres rozkładu ocen - konfiguracja 1

Histogram ocen planów zajęć:



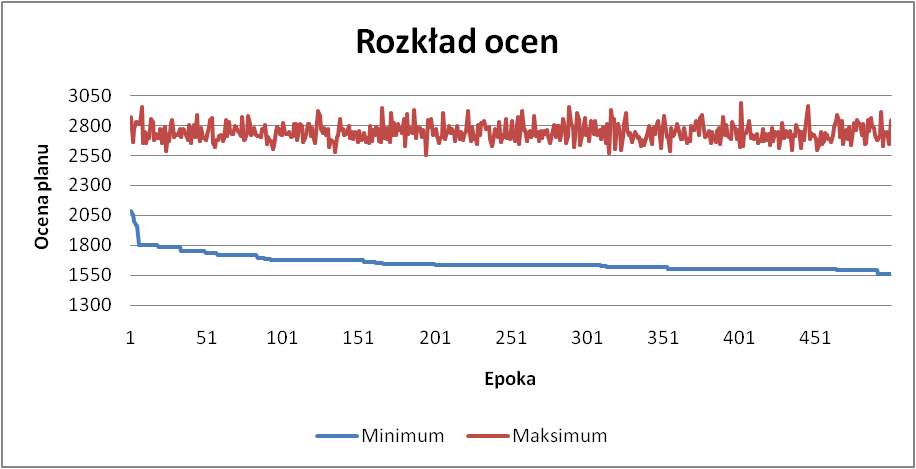
Rysunek Histogram ocen - konfiguracja 1

1. Konfiguracja nr 2

Czas generowania planu zajęć: 18:19,4 min

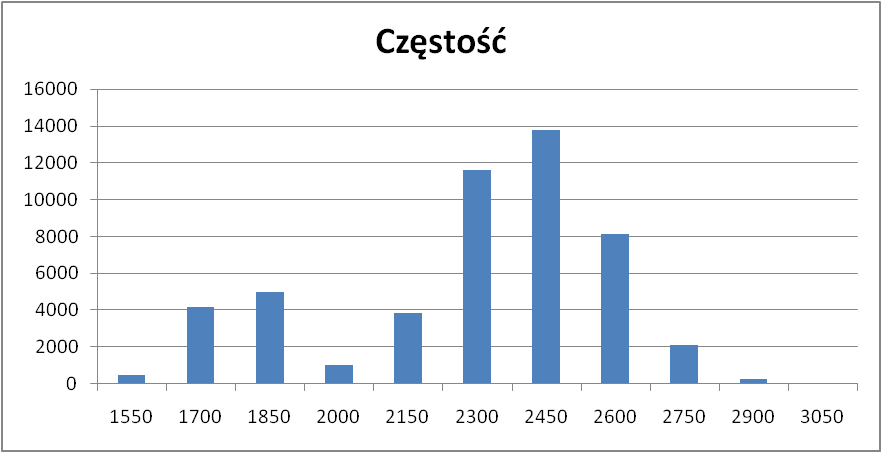
Ocena funkcji przystosowania planu wynikowego: 1556

Rozkład ocen planów podczas procesu generowania:



Rysunek Wykres rozkładu ocen - konfiguracja 2

Histogram ocen planów zajęć:



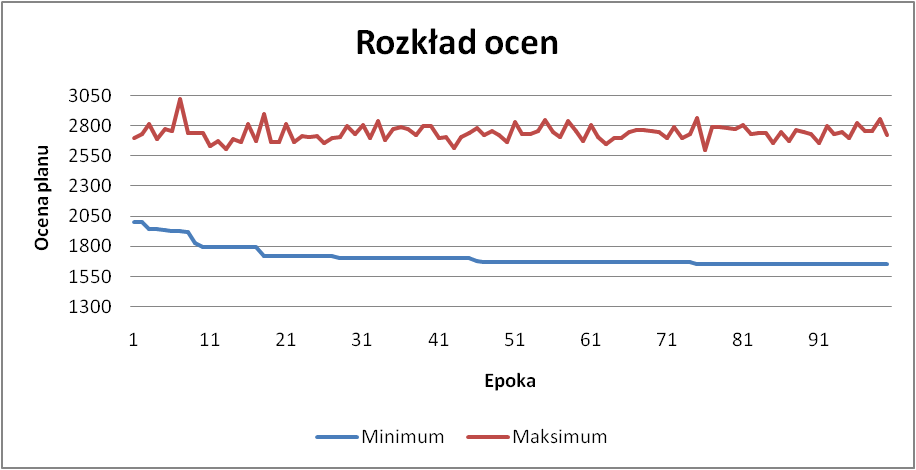
Rysunek Histogram ocen - konfiguracja 2

1. Konfiguracja nr 3

Czas generowania planu zajęć: 04:23,0 min

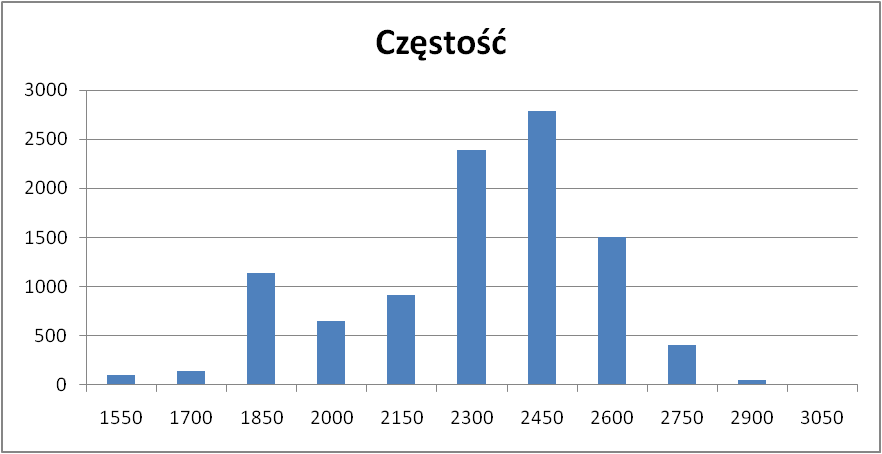
Ocena funkcji przystosowania planu wynikowego: 1652

Rozkład ocen planów podczas procesu generowania:



Rysunek Wykres rozkładu ocen - konfiguracja 3

Histogram ocen planów zajęć:



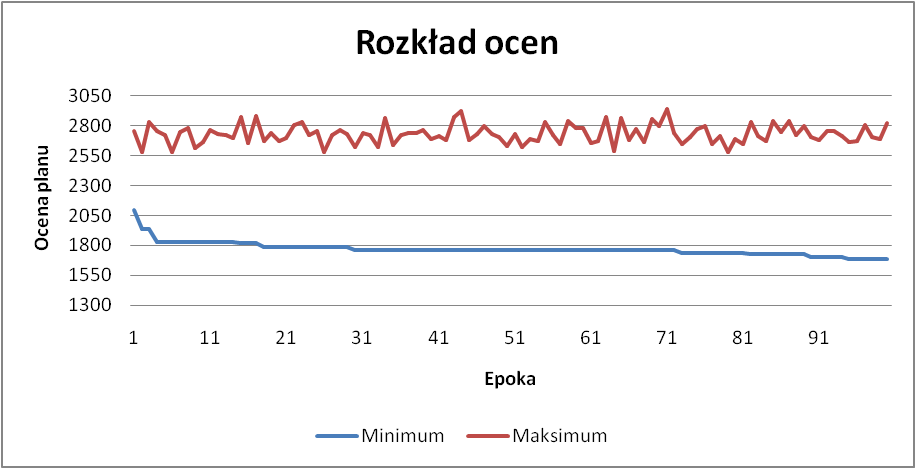
Rysunek Histogram ocen - konfiguracja 3

1. Konfiguracja nr 4

Czas generowania planu zajęć: 03:48,8 min

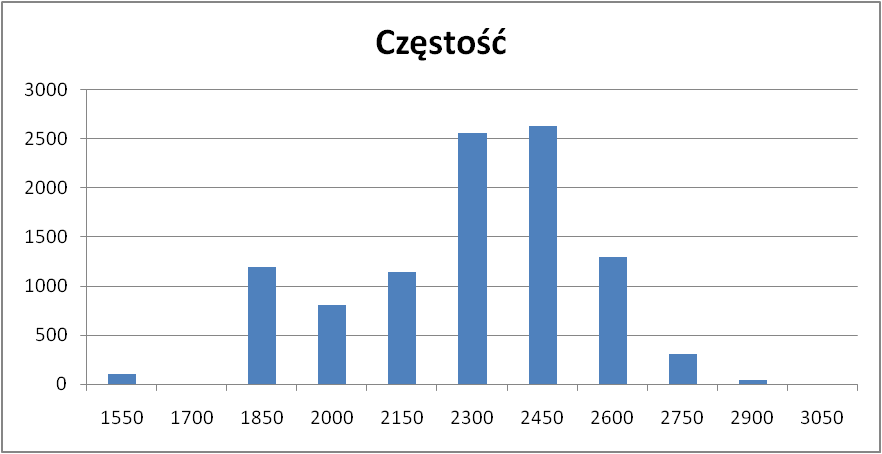
Ocena funkcji przystosowania planu wynikowego: 1684

Rozkład ocen planów podczas procesu generowania:



Rysunek Wykres rozkładu ocen - konfiguracja 4

Histogram ocen planów zajęć:



Rysunek Histogram ocen - konfiguracja 4

# Podsumowanie

Celem niniejszej pracy, było stworzenie aplikacji generującej plan zajęć dla szkół. Aplikacja internetowa, która została stworzona pozwala na wprowadzenia danych dotyczących szkoły, nauczycieli, grup, sal lekcyjnych oraz lekcji. Użytkownik ma możliwość dostosowania do własnych potrzeb konfiguracji algorytmu w której skład wchodzi:

* wielkości populacji,
* ilości epok,
* prawdopodobieństwo mutacji,
* ilość lekcji do wstawienia w pojedynczym kroku,
* dolne i górne ograniczenie przedziału godzin o wyższym priorytecie.

Wygenerowany plan zajęć można wyświetlić w postaci tabeli z podziałem na dni i godziny dla każdego nauczyciela, grupy oraz sali lekcyjnej.

Generator planów zajęć może być rozwijany poprzez dodanie możliwości ręcznego ustawiania lekcji w planie w sposób przyjazny użytkownikowi. Typy sal lekcyjnych można rozszerzyć o listę typów alternatywnych. Funkcję oceniającej przystosowanie można rozszerzyć o ocenę kolejnych cech charakteryzujących „dobry” plan zajęć, takich jak: równomierny rozkład okienek w dniach tygodnia w planach zajęć nauczycieli, minimalna ilość lekcji o wysokim priorytecie w ostatnich godzinach dnia. Do funkcje szukającą najlepszego slotu można dodać nakaz wybór slotu z dnia w którym nie występuje dana lekcja.

# Literatura

1. Gwiazda, T. D. (2007). *Algorytmy genetyczne - kompedium TOM II Operator mutacji dla problemów numeryczncyh.* Warszawa: PWN.
2. Gwiazda, T. D. (2007). *Algorytmy genetyczne - kompedium, TOM I, Operator krzyżowania dla problemów numeryczncyh.* Warszawa: PWN.