

STUNDENPLANUNG mit genetischen Algorithmen

Deniz Özdemirli

Fitness

```
def calculate_fitness(self):
    self._numOfConflicts = 0
    classes = self.get_classes()

    for i in range(len(classes)):
        for j in range(len(classes)):
            if j >= i:
                if classes[i].get_meetingTime() ==
                    classes[j].get_meetingTime() \
                    and classes[i].get_id() !=
                        classes[j].get_id():
                    if classes[i].get_dept() ==
                        classes[j].get_dept():
                        self._numOfConflicts += 1
                    if classes[i].get_room() ==
                        classes[j].get_room():
                        self._numOfConflicts += 1

                if classes[i].get_instructor() ==
                    classes[j].get_instructor():
                        self._numOfConflicts += 1

    return 1 / (1.0 * (self._numOfConflicts + 1))
```

Evolve

```
def evolve(self, population):
    p1,p2 = self.tournament_selection(population)
    crossovered1, crossovered2 =
        self.crossover(p1,p2)
    mutatedSchedule1 = self.mutation(crossovered1)
    mutatedSchedule2 = self.mutation(crossovered2)
    population.get_schedules().append(mutatedSchedule1)
    population.get_schedules().append(mutatedSchedule2)
    schedules = sorted(population.get_schedules(),
        key=lambda agent: agent.get_fitness(),
        reverse=True)
    population.set_schedules(schedules[:POPULATION_SIZE])
    return population
```

Der genetische Algorithmus ist eine von vielen Methoden, die zur Erstellung eines Stundenplans für die Uni verwendet werden können.

Class #	Jahrgang	Courses	Room	Instructor	Meeting Time
0	1.Jahrgang	INF101	MD B-ED-2-1	Volkan Gezer	Freitag 08:30 - 10:30
1	1.Jahrgang	INF103	MD B-ED-1-14	Faruk Bağcı	Montag 08:30 - 10:30
2	1.Jahrgang	INF107	MD B-CS-2-15	Faruk Bağcı	Mittwoch 08:30 - 10:30
3	1.Jahrgang	ENG101	MD A-ED-2-3	Vahap Sümer ÖZSÜER	Donnerstag 12:30 - 14:30
4	1.Jahrgang	MAT103	SIEMENS Otomasyon LAB	İzzet Göksel	Donnerstag 10:30 - 12:30
5	1.Jahrgang	TUR001	MD B-ED-2-17,0	Gül Ayşe Akar	Freitag 14:30 - 16:30
6	1.Jahrgang	DEU121	FIZIK-LAB	Selahaddin Soyudoğru	Montag 10:30 - 12:30
7	2.Jahrgang	INF201	MD B-ED-2-17,0	Canan Yıldız	Mittwoch 14:30 - 16:30
8	2.Jahrgang	INF209	FIZIK-LAB	Faruk Bağcı	Donnerstag 14:30 - 16:30
9	2.Jahrgang	INF205	MD B-ED-2-17,0	Canan Yıldız	Dienstag 10:30 - 12:30
10	2.Jahrgang	INF203	SIEMENS Otomasyon LAB	Emel Maden Yılmaz	Freitag 14:30 - 16:30
11	2.Jahrgang	ENG201	MD A-ED-1-12	Vahap Sümer ÖZSÜER	Donnerstag 10:30 - 12:30
12	2.Jahrgang	AIT001	MD B-ED-2-2	Güneş Çap	Montag 10:30 - 12:30
13	2.Jahrgang	INF211	MD A-ED-1-12	Ali Gökhan Yavuz	Montag 12:30 - 14:30
14	3.Jahrgang	MEC313	MD B-ED-2-15	Prof Dr. Hasan Smajić	Montag 12:30 - 14:30
15	3.Jahrgang	INF506	PC LAB	Emre Işık	Mittwoch 12:30 - 14:30
16	3.Jahrgang	ENG301	MD B-ED-2-15	Vahap Sümer ÖZSÜER	Montag 14:30 - 16:30
17	3.Jahrgang	INF303	MD B-ED-2-15	Ömer Karacan	Dienstag 08:30 - 10:30
18	3.Jahrgang	INF701	SIEMENS Otomasyon LAB	Carsten Gips	Donnerstag 12:30 - 14:30
19	3.Jahrgang	EFE091	PC LAB	Mehmet Gökhan Habiboğlu	Donnerstag 10:30 - 12:30
20	3.Jahrgang	ISG001	MD B-ED-2-2	J. Kuntze	Freitag 12:30 - 14:30
21	3.Jahrgang	INF517	MD B-ED-2-14	Dr. Selcan İpek-Uğay	Freitag 08:30 - 10:30
22	4.Jahrgang	INF523	MD B-ED-2-15	Dilek Göksel Duru	Mittwoch 14:30 - 16:30
23	4.Jahrgang	MAT201	PC LAB	Doç. Dr. Haydar Uncu	Donnerstag 12:30 - 14:30
24	4.Jahrgang	INF714	MD A-ED-2-3	Emre Işık	Donnerstag 14:30 - 16:30
25	4.Jahrgang	INF905	MD B-CS-2-15	MSc. Berkant Bostan	Dienstag 10:30 - 12:30
26	4.Jahrgang	INF499	MD B-ED- 1-13	Emre Işık	Dienstag 08:30 - 10:30
27	4.Jahrgang	INF401	MD B-ED-1-2	Emre Işık	Montag 12:30 - 14:30

Fitness Score: 1.0, Number of Conflicts: 0

ein konfliktfreier Lehrplan

Die 5 **Hauptphasen** vom genetischen Algorithmus sind:

1. Anfängliche Population
2. Fitness-Funktion
3. Auswahl (Tournament Selektion)
4. Kreuzung (Single Point Crossover)
5. Mutation

Modellierung

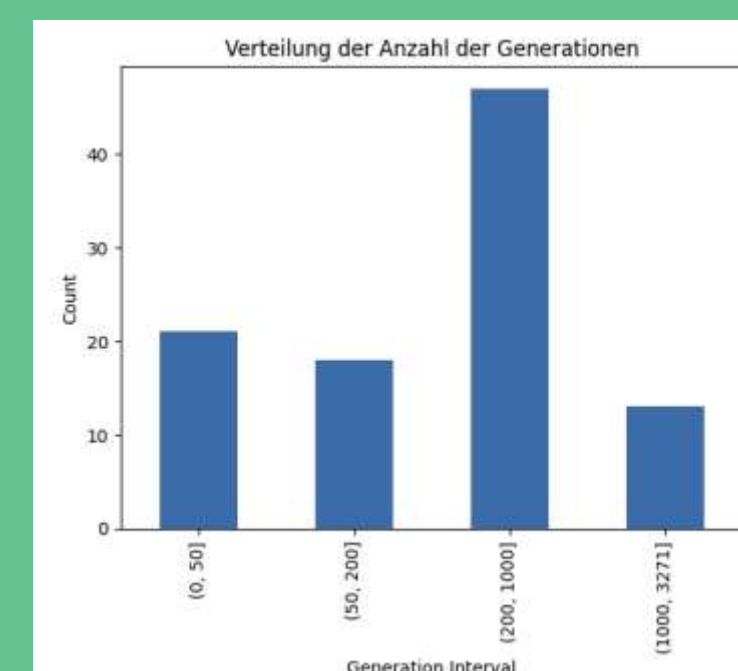
```
ROOMS = [['MD B-ED-2-2', 500],...]
MEETING_TIMES = [['MT1', 'Montag 08:30 - 10:30'],...]
INSTRUCTORS = [['I1', 'Ömer Faruk AYDIN'],...]
course1 = Course("C1", "MEC313", [INSTRUCTORS[8],...], 45)
COURSES = [course1,...]
dept4 = Department("4.Jahrgang", [course23,...])
DEPTS = [dept1, dept2, dept3, dept4]
data = Data(ROOMS, MEETING_TIMES, INSTRUCTORS, COURSES, DEPTS)
population = Population(POPULATION_SIZE,data)
geneticAlgorithm = GeneticAlgorithm()
new_population = geneticAlgorithm.evolve(population)
```

Ergebnisse von Experiment

Um die Effizienz des Algorithmus zu messen, wurde das Experiment 100 Mal wiederholt. Der konfliktfreie Lehrplan wurde in durchschnittlich 446 Generationen gefunden.

Parametern

```
POPULATION_SIZE = 9
TOURNAMENT_SELECTION_SIZE = 2
MUTATION_RATE = 0.1
```



	Value
Mean	446.490000
Std. Deviation	482.737703
Min	0.000000
Max	3271.000000