

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN CUỐI KỲ MÔN NHẬP MÔN TRÍ TUỆ NHÂN TẠO

XÂY DỰNG CHƯƠNG TRÌNH SẮP XẾP THỜI KHÓA BIỂU CỦA MỘT HỌC KỲ

Người hướng dẫn: **LÊ ANH CƯỜNG**

Người thực hiện: **TRẦN HƯNG TRỌNG – 51800943**

NGUYỄN MINH ĐĂNG KHOA – 51800882

Lớp : 18050302

Khoá : 22

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN CUỐI KỲ MÔN NHẬP MÔN TRÍ TUỆ NHÂN TẠO

XÂY DỰNG CHƯƠNG TRÌNH SẮP XẾP THỜI KHÓA BIỂU CỦA MỘT HỌC KỲ

Người hướng dẫn: **LÊ ANH CƯỜNG**
Người thực hiện: **TRẦN HÙNG TRỌNG – 51800943**
NGUYỄN MINH ĐĂNG KHOA – 51800882
Lớp : **18050302**
Khoá : **22**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành và sâu sắc đến thầy Lê Anh Cường – người trực tiếp hướng dẫn tận tình, tận tâm và những lời góp ý quý báu của thầy trong suốt quá trình học giúp chúng em thuận lợi trong việc hoàn thành bài báo cáo cuối kỳ.

MỤC LỤC

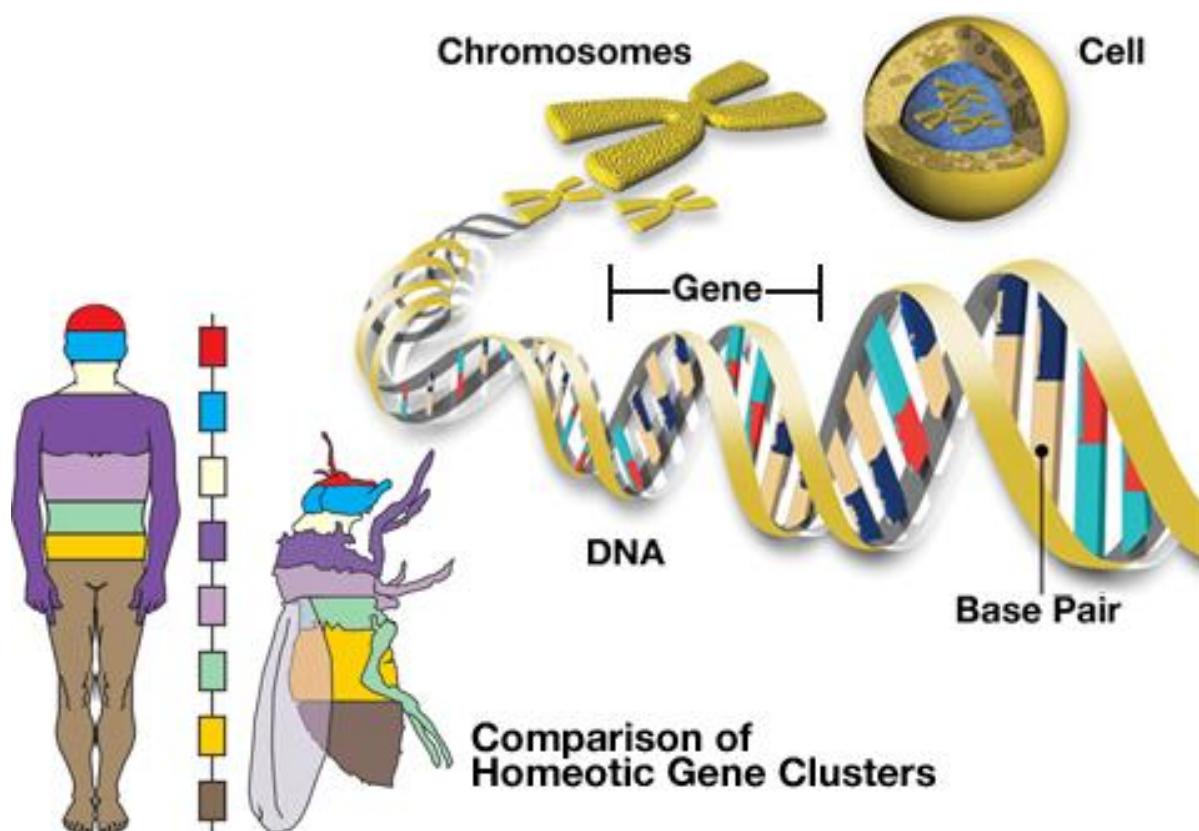
LỜI CẢM ƠN	i
MỤC LỤC.....	1
DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ	2
CHƯƠNG 1 – LÝ THUYẾT	3
1.1 Di truyền	3
1.2 Tiến hóa.....	4
1.3 Thuật toán di truyền – Genetic Algorithm (GA)	5
CHƯƠNG 2 – THỰC HÀNH.....	9
2.1 Giới thiệu bài toán.....	9
2.2 Tạo ngẫu nhiên quần thể	10
2.3 Khởi tạo một gen của cá thể.....	10
2.4 Đánh giá thích nghi	11
2.5 Chọn lọc	13
2.6 Lai ghép.....	14
2.7 Đột biến.....	15
CHƯƠNG 3 – KẾT QUẢ	18
TÀI LIỆU THAM KHẢO.....	21

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

DANH MỤC HÌNH

Hình 1: Minh họa trong sinh học	3
Hình 2: Sơ đồ tiến hóa từ loài vượn cổ	4
Hình 3: Minh họa population, chromosome, gene.....	6
Hình 4: Sơ đồ thuật toán di truyền	7
Hình 5: Dạng chuỗi gen của một cá thể	9
Hình 6: Hàm tạo quần thể	10
Hình 7: Hàm tạo gen	10
Hình 8: Ví dụ ma trận để lưu lịch sử dụng phòng.....	11
Hình 9: Đánh giá tiêu chí số lớp được mở	11
Hình 10: Đánh giá độ chênh lệch số ca dạy của giảng viên và khoảng cách giữa các phòng khi có các ca trong cùng một ngày.....	12
Hình 11: Đánh giá số phòng được dùng	13
Hình 12: Hàm chọn lọc	13
Hình 13: Hàm lai ghép	14
Hình 14: Random một cá thể	15
Hình 15: Random vị trí gen bị đột biến	15
Hình 16: Đột biến kiểu thêm vào	16
Hình 17: Đột biến kiểu xóa bớt.....	16
Hình 18: Đột biến kiểu thay đổi.....	17
Hình 19: Giao diện chính THỜI KHÓA BIỂU.....	18
Hình 20: Giao diện SỬA DỮ LIỆU.....	19
Hình 21: Giao diện THÊM DỮ LIỆU	20

CHƯƠNG 1 – LÝ THUYẾT

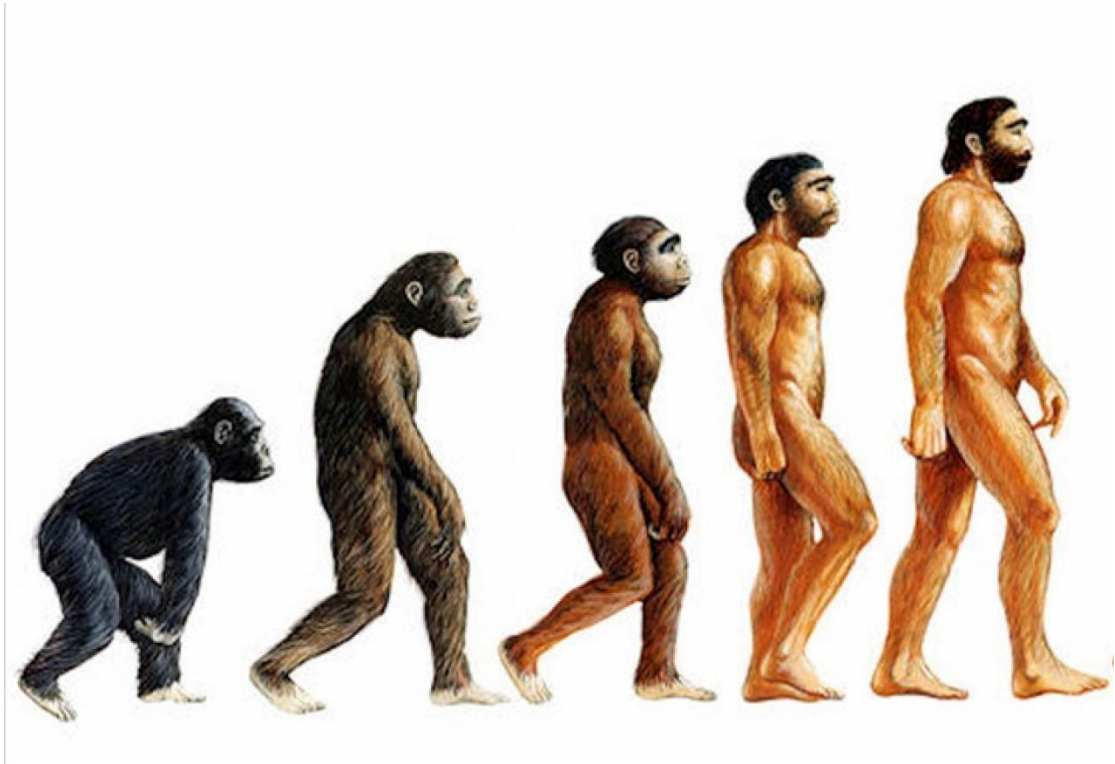


Hình 1: Minh họa trong sinh học

1.1 Di truyền

Di truyền là hiện tượng chuyển những tính trạng của cha mẹ, tổ tiên cho các thế hệ con, cháu thông qua gen. Trong sinh học và di truyền học, di truyền chuyển những đặc trưng sinh học từ một sinh vật cha mẹ đến con cái và nó đồng nghĩa với di chuyển gen, gen thừa nhận mang thông tin sinh học hay thông tin di truyền.

1.2 Tiến hóa



Hình 2: Sơ đồ tiến hóa từ loài vượn cổ

Tiến hóa nói đến quá trình hoàn thiện, biến đổi dần để hoàn thiện hơn các bộ phận, chức năng của các sinh vật để phù hợp hơn với điều kiện sinh tồn cũng đang dần thay đổi. Trong sinh học, tiến hóa là sự thay đổi đặc tính di truyền của một quần thể sinh học qua những thế hệ nối tiếp nhau. Các quá trình tiến hóa làm nảy sinh sự đa dạng ở mọi mức độ tổ chức sinh học bao gồm loài, các cá thể sinh vật và cả các phân tử như ADN và protein.

Tiến hóa do chọn lọc tự nhiên là một quá trình có thể suy ra từ ba thực kiện về các quần thể sinh học:

- Nhiều cá thể con được sinh ra hơn số lượng có thể sống sót.
- Các tính trạng khác nhau giữa các cá thể, dẫn tới tỉ lệ sinh tồn và sinh sản khác nhau.
- Những sự khác biệt về đặc điểm trên là có tính di truyền.

Do đó, khi những cá thể của một quần thể chết đi, chúng được thay thế bằng những hậu duệ của thế hệ cha mẹ nhưng có thể thích nghi tốt hơn để tồn tại và sinh sôi trong môi trường mà sự chọn lọc tự nhiên diễn ra. Quá trình này tạo ra và bảo tồn những đặc điểm được cho là phù hợp hơn cho chức năng mà chúng đảm nhiệm.

Cho đến nay, sự chọn lọc tự nhiên là nguyên nhân duy nhất cho sự thích nghi, tuy nhiên không phải là nguyên nhân duy nhất cho sự tiến hóa. Những nguyên nhân khác của tiến hóa bao gồm sự đột biến và dịch chuyển di truyền. Vào đầu thế kỷ 20, di truyền học kết hợp với lý thuyết tiến hóa nhờ chọn lọc tự nhiên của Darwin thông qua di truyền học quần thể. Tầm quan trọng của chọn lọc tự nhiên như một nguyên nhân tiến hóa đã được chấp nhận trong những nhánh khác của sinh học.

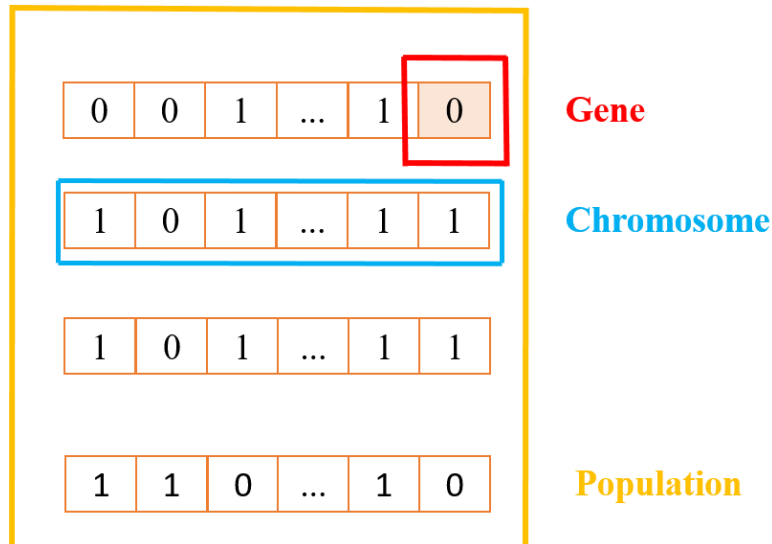
1.3 Thuật toán di truyền – Genetic Algorithm (GA)

Genetic Algorithm (GA) là kỹ thuật phỏng theo quá trình thích nghi tiến hóa của các quần thể sinh học dựa trên học thuyết Darwin. GA là phương pháp tìm kiếm tối ưu dựa vào ngẫu nhiên và phù hợp bằng cách mô phỏng theo sự tiến hóa của con người hay của sinh vật. Tư tưởng của thuật toán di truyền là mô phỏng các hiện tượng tự nhiên, là kế thừa và đấu tranh sinh tồn.

Genetic Algorithm thuộc lớp các giải thuật xuất sắc nhưng lại rất khác các giải thuật ngẫu nhiên vì chúng kết hợp các phần tử tìm kiếm trực tiếp và ngẫu nhiên. Khác biệt quan trọng giữa tìm kiếm của Genetic Algorithm và các phương pháp tìm kiếm khác là Genetic Algorithm duy trì và xử lý một tập các lời giải, gọi là một quần thể (population).

Trong Genetic Algorithm, việc tìm kiếm giả thuyết thích hợp được bắt đầu với một quần thể (population), hay một tập hợp có chọn lọc ban đầu của các giả thuyết. Các cá thể (chromosome) của quần thể hiện tại khởi nguồn cho quần thể thế hệ kế tiếp bằng các hoạt động lai ghép (crossover) và đột biến (mutation) ngẫu nhiên – được lấy mẫu sau các quá trình tiến hóa sinh học. Ở mỗi bước, các giả thuyết trong quần thể hiện tại được ước lượng liên hệ với đại lượng thích nghi, với các giả thuyết phù hợp nhất được chọn

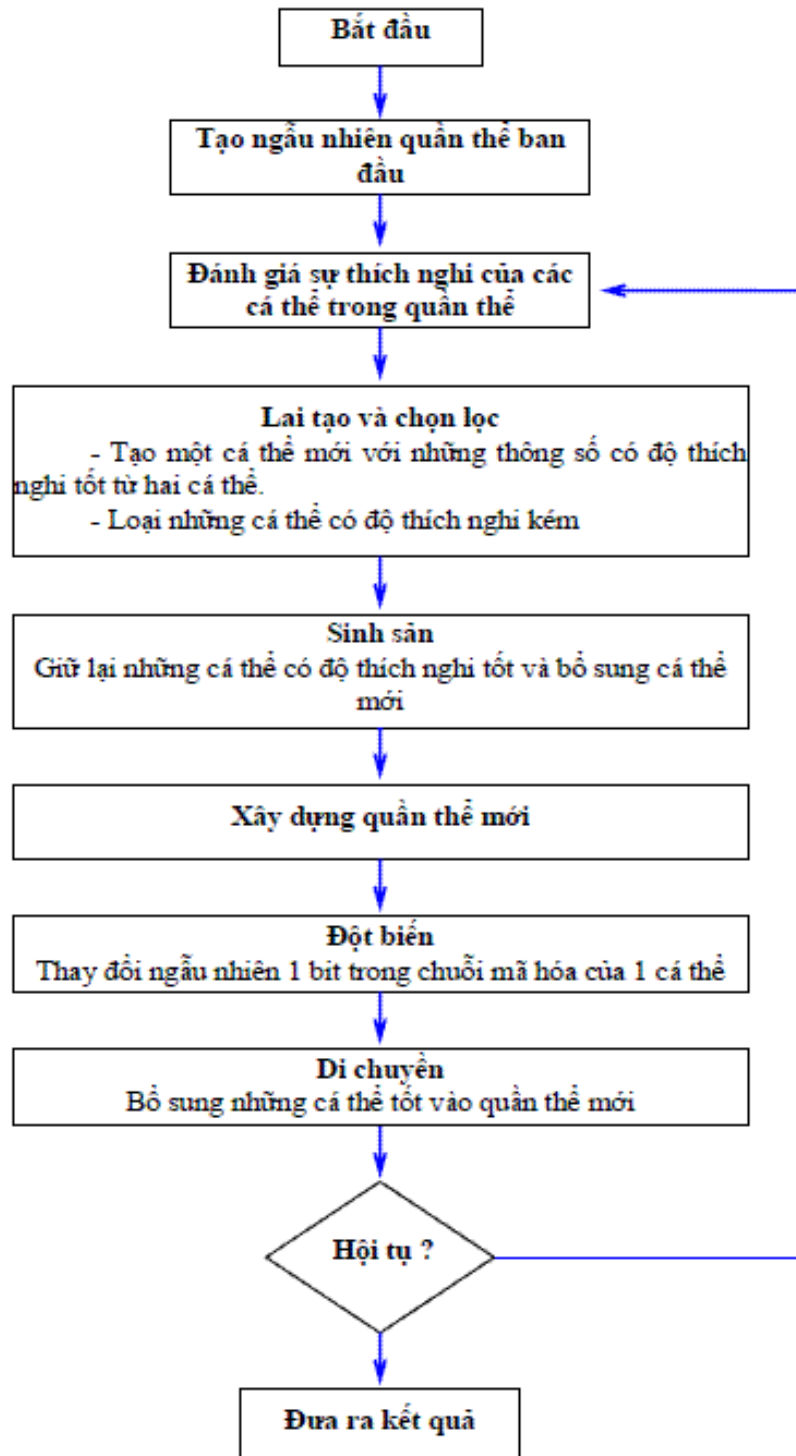
theo xác suất là các hạt giống cho việc sản sinh thế hệ kế tiếp, gọi là cá thể (chromosome). Cá thể nào phát triển hơn, thích ứng hơn với môi trường sẽ tồn tại và ngược lại sẽ bị đào thải. Genetic Algorithm có thể dò tìm thế hệ mới có độ thích nghi tốt hơn. Mỗi vị trí trong chromosome được gọi là gen và mỗi chromosome có n gen.



Hình 3: Minh họa population, chromosome, gene

Genetic Algorithm giải quyết các bài toán quy hoạch toán học thông qua các quá trình cơ bản: chọn lọc (selection), lai tạo (crossover) và đột biến (mutation) cho các cá thể trong quần thể. Dùng Genetic Algorithm đòi hỏi phải xác định được: khởi tạo quần thể ban đầu, hàm đánh giá các lời giải theo mức độ thích nghi – hàm mục tiêu, các toán tử di truyền tạo hàm sinh sản.

Thuật toán Genetic Algorithm hiện được dùng rộng rãi trong rất nhiều lĩnh vực để tối ưu giá trị cho bộ tham số của một hệ thống hay một phương pháp.



Hình 4: Sơ đồ thuật toán di truyền

Trong hình trên:

- Chọn lọc (selection): chọn lọc những cá thể có fitness (hoặc score) tốt để lai ghép và đột biến.
- Lai ghép (crossover): trao đổi gen giữa hai cá thể.
- Đột biến (mutation): đột biến gen của một cá thể.

Ngoài ra còn có bước khởi tạo quần thể nhằm tạo ngẫu nhiên quần thể ban đầu. Bước đánh giá thích nghi có mục đích tính giá trị fitness (hoặc score) cho các cá thể. Ba bước chính của Genetic Algorithm sẽ được lặp đi lặp lại cho đến khi nào điều kiện dừng được thỏa mãn. Điều kiện dừng có thể là số lần lặp (generation), tính bão hòa của giá trị `max_fitness`, hay sự hội tụ của quần thể (các cá thể có fitness gần giống nhau).

CHƯƠNG 2 – THỰC HÀNH

Viết chương trình sắp xếp thời khoá biểu của một học kỳ cho một trường đại học hoặc một trường phổ thông.

2.1 Giới thiệu bài toán

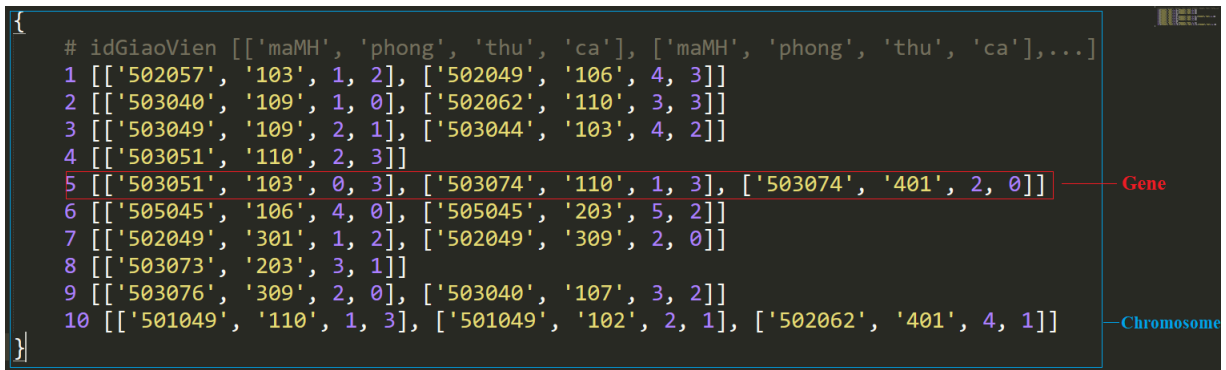
Các thông tin về thời khóa biểu bao gồm:

- Giáo viên và môn học: giáo viên có thể dạy những môn nào.
- Giáo viên và số tiết học tối thiểu, tối đa trong một tuần.
- Phòng học, ca học: có những phòng học nào; mỗi phòng học trong 1 ngày được chia làm mấy ca, mỗi ca bao nhiêu tiết.
- Trong học kỳ này, mỗi môn học cần được tổ chức bao nhiêu lớp.

Có 4 file csv (4 table) được tạo gồm:

- classroom.csv: 40 phòng (x01 – x10 với x: 1 – 4).
- lecturers.csv: id (idGV), monhoc (idMH).
- lecturers_info.csv: id (idGV), name (tenGV).
- subjects.csv: mamon (idMH), ten (tên MH), sotiet (số tiết), solop (số lớp).

Sắp xếp một chuỗi gen của một cá thể như sau:



```
{
  # idGiaoVien [['maMH', 'phong', 'thu', 'ca'], ['maMH', 'phong', 'thu', 'ca'],...]
  1 [['502057', '103', 1, 2], ['502049', '106', 4, 3]]
  2 [['503040', '109', 1, 0], ['502062', '110', 3, 3]]
  3 [['503049', '109', 2, 1], ['503044', '103', 4, 2]]
  4 [['503051', '110', 2, 3]]
  5 [['503051', '103', 0, 3], ['503074', '110', 1, 3], ['503074', '401', 2, 0]]
  6 [['505045', '106', 4, 0], ['505045', '203', 5, 2]]
  7 [['502049', '301', 1, 2], ['502049', '309', 2, 0]]
  8 [['503073', '203', 3, 1]]
  9 [['503076', '309', 2, 0], ['503040', '107', 3, 2]]
  10 [['501049', '110', 1, 3], ['501049', '102', 2, 1], ['502062', '401', 4, 1]]
}
```

Hình 5: Dạng chuỗi gen của một cá thể

2.2 Tạo ngẫu nhiên quần thể

```
# Hàm tạo quần thể
def createPopulations(self, n = 100):
    lecturerId = self.lecturers.keys()
    print("Create: ", lecturerId)

    totalClasses = 0 # Tổng số lớp
    for x in self.subjects.values():
        totalClasses += x[-1] # Col solop

    i = 0
    while (i < n):
        self.populations.append(self.createGen(lecturerId, totalClasses // len(lecturerId) + 1))
        i += 1
```

Hình 6: Hàm tạo quần thể

Tạo quần thể với số cá thể $n = 100$ và số gen của một cá thể là số giáo viên và số lớp của mỗi giáo viên.

2.3 Khởi tạo một gen của cá thể

```
# Hàm tạo gen
def createGen(self, lecturerId, n = 3):
    gen = {key: [] for key in lecturerId}
    scheduleRoom = [[[[] for i in range(7)] for j in range(4)]]

    for _id in lecturerId:
        numShifts = rd.randint(n - 1, n)
        haveShifts = [] # Chứa những ca được sắp xếp để tránh bị trùng lặp

        for i in range(numShifts):
            if (self.lecturers[_id] == [] or self.lecturers[_id] == None):
                continue

            subjectCode = rd.choice(self.lecturers[_id])
            shift = self.myRandom(0)
            weekDays = self.myRandom(1)

            while ([weekDays, shift] in haveShifts):
                shift = self.myRandom(0)
                weekDays = self.myRandom(1)

            room = rd.choice(self.classRoom)

            while (room in scheduleRoom[shift][weekDays]):
                room = rd.choice(self.classRoom)

            haveShifts.append([weekDays, shift])
            scheduleRoom[shift][weekDays].append(room)
            gen[_id].append([subjectCode, room, weekDays, shift])

    return gen
```

Hình 7: Hàm tạo gen

Đầu tiên khởi tạo gen: `gen = {key: [] for key in lecturerId}`

Khởi tạo ma trận để lưu lịch sử dụng phòng với mục đích để không dùng phòng trùng ca: `scheduleRoom = [[[for i in range(7)] for j in range(4)]`

```
[
  [
    [], [], [], [], [], [], []
    [404, 102], [], [], [], [], [], []
    [], [], [], [], [], [], []
    [], [], [], [109], [], [], []
    [], [], [], [], [], [], []
  ]
]
```

Hình 8: Ví dụ ma trận để lưu lịch sử dụng phòng

Tạo gen:

- Với mỗi `_id` (mỗi giáo viên) -> random số ca dạy trong tuần
- Với mỗi ca dạy -> random mã môn, phòng, thứ, ca -> tạo thành list -> append vào `gen[_id]` -> lưu trữ các ca dạy

2.4 Đánh giá thích nghi

Dựa vào giá trị rate (fitness hoặc score) của mỗi cá thể, bước này sẽ chọn ra một tập hợp các cá thể có rate tốt nhất. Nguyên tắc chọn là cá thể nào có rate càng cao thì khả năng cá thể đó sẽ được chọn càng nhiều lần.

Đánh giá tuyệt đối:

- Đánh giá tiêu chí số lớp được mở:
 - Đếm số lớp của từng môn trong GEN.
 - So sánh kết quả với dữ liệu và tính rate.

```
def rateGen(self, gen, scheduleRoom, type1 = 1):
    rate = 10000 # Để cho số nó ra dương cho đẹp
    test = {idMH: 0 for idMH in self.subjects.keys()}

    # Đánh giá tiêu chí số lớp được mở
    for val in gen.values(): # Đếm số lớp của từng môn trong GEN
        if (type(val) != int):
            for idMH in val:
                test[idMH[0]] += 1

    for idMH in self.subjects.keys(): # So sánh kết quả với dữ liệu và tính rate
        if (type1 == 0):
            print("%s - real = %d - new = %d - sub = %d" % (idMH, self.subjects[idMH][-1], test[idMH], abs(self.subjects[idMH][-1] - test[idMH])))
            # Dù chỉ lệch một lớp cũng bị trừ điểm cực nặng -> Không thể lên top
            rate -= abs(self.subjects[idMH][-1] - test[idMH]) * 10000

    if (type1 == 0):
        print("B1 = ", rate)
```

Hình 9: Đánh giá tiêu chí số lớp được mở

Đánh giá tương đối:

- Đánh giá độ chênh lệch số ca dạy của giảng viên.
- Đánh giá khoảng cách giữa các phòng khi có các ca trong cùng một ngày:
 - Sắp xếp lại lịch dạy của giảng viên theo thứ.
 - Khởi tạo: nowWeekDay, nowRoom là ngày, phòng của buổi dạy.
 - Nếu là cùng ngày thì tính khoảng cách giữa hai phòng, sau đó cập nhật lại nowWeekDay, nowRoom.
 - Lặp lại cho từng giảng viên.

```
# Đánh giá độ chênh lệch số ca dạy của giảng viên
maxNumShifts = -inf
minNumShifts = inf
distance = 0 # Tổng khoảng cách giữa các các phòng của một giảng viên trong cùng một ngày

for _id, val in gen.items():
    if (val is None or val == []):
        continue

    if (type(val) != int):
        # Tìm số lượng max, min
        if (len(val) > maxNumShifts):
            maxNumShifts = len(val)
        if (len(val) < minNumShifts):
            minNumShifts = len(val)

    # Đánh giá khoảng cách giữa các phòng khi có các ca trong cùng một ngày
    gen[_id] = sorted(val, key=lambda x: x[2]) # sắp xếp theo thứ
    nowWeekDay = gen[_id][0][2]
    nowRoom = gen[_id][0][1]

    for x in gen[_id][1:]:
        if (nowWeekDay == x[2]): # Cùng ngày thì tính khoảng cách
            distance += abs(int(x[1]) - int(nowRoom))
            nowWeekDay = x[2]
            nowRoom = x[1]

rate = rate - (maxNumShifts - minNumShifts) * 1000
if (type1 == 0):
    print("B2 Đánh giá ca dạy = ", rate)
    print("MAX ca = %d - MIN ca = %d"%(maxNumShifts, minNumShifts))

rate = rate - distance * 5
if (type1 == 0):
    print("B3 Đánh giá khoảng cách phòng = ", rate)
```

Hình 10: Đánh giá độ chênh lệch số ca dạy của giảng viên và khoảng cách giữa các phòng khi có các ca trong cùng một ngày

- Đánh giá số phòng được dùng:

- Duyệt hết lịch phòng trong scheduleRoom.
- Thêm tất cả các phòng list counterRoom.
- Chuyển list thành set -> lấy độ dài để tính rate.

```
# Đánh giá số lượng phòng được dùng
counterRoom = []
for shift in range(4):
    for weekday in range(7):
        counterRoom.extend(scheduleRoom[shift][weekday])

rate = rate - len(set(counterRoom)) * 10
if (type1 == 0):
    print("B4 = ", rate)
return rate
```

Hình 11: Đánh giá số phòng được dùng

2.5 Chọn lọc

```
# Hàm chọn lọc
def selectGen(self, percent):
    num = int(percent * (len(self.populations)))
    self.populations = sorted(self.populations, key=Lambda x: x["rate"], reverse=True)
    self.populations = self.populations[0:num]
```

Hình 12: Hàm chọn lọc

Giữ lại phần trăm có độ thích nghi phù hợp nhất (Có thể giữ lại ½, lai ghép để tạo thêm số lượng).

Sắp xếp quần thể lại theo điểm đánh giá (rate) từ cao xuống thấp.

2.6 Lai ghép

```
# Hàm lai ghép
def hybridGen(self):
    n = len(self.populations)
    for times in range(n):
        father = rd.randint(0, len(self.populations) - 1)
        mother = rd.randint(0, len(self.populations) - 1)
        con = self.populations[father].copy()

        while (mother == father): # Đảm bảo không bị trùng
            mother = rd.randint(0, len(self.populations) - 1)

        for _id in con.keys():
            if (rd.randint(0,1) == 1):
                con[_id] = self.populations[mother][_id]

        self.populations.append(con)
```

Hình 13: Hàm lai ghép

Mục đích tạo ra thêm n cá thể mới bổ sung vào quần thể:

- Random cá thể (gen) cha, mẹ (đảm bảo hai gen cha, mẹ không giống nhau).
- Đặt cá thể con = copy cá thể cha.
- Với từng _id quyết định xem con sẽ nhận gen của cha hoặc mẹ.
- Thêm vào quần thể.
- Lặp n lần.

2.7 Đột biến

Tạo thêm n cá thể với gen được đột biến.

Random một cá thể mới new.

```
# Hàm đột biến
def mutationGen(self):
    lecturerId = list(self.lecturers.keys())

    # Random một cá thể bất kỳ
    index = rd.randint(0, len(self.populations) - 1)
    new = self.populations[index].copy()

    # Đối với đột biến kiểu thêm vào hoặc mất đi
    scheduleRoom = [[[] for i in range(7)] for j in range(4)]

    for val in self.populations[index].values():
        if (type(val) != int):
            for idMH, room, weekDays, shift in val:
                scheduleRoom[shift][weekDays].append(room)
```

Hình 14: Random một cá thể

Random vị trí gen bị đột biến.

```
_id = rd.choice(lecturerId)

while (self.lecturers[_id] is None or self.lecturers[_id] == []):
    _id = rd.choice(lecturerId)
```

Hình 15: Random vị trí gen bị đột biến

Random loại đột biến:

typeMutation = rd.randint(0, 2)

- Thêm vào (0):

- Random thứ và ca dạy.
- Đảm bảo không bị trùng ca dạy đã có.
- Random mã môn học, phòng.

- Đảm bảo không bị trùng phòng trong một ca của một thứ.
- Thêm list mới vào new.

```

if (typeMutation == 0):
    # đột biến kiểu thêm vào
    haveShifts = [[x[2], x[3]] for x in new[_id]]
    shift = self.myRandom(0)
    weekDays = self.myRandom(1)

    while ([weekDays, shift] in haveShifts):
        shift = self.myRandom(0)
        weekDays = self.myRandom(1)

    subjectCode = rd.choice(self.lecturers[_id])
    room = rd.choice(self.classRoom)

    while (room in scheduleRoom[shift][weekDays]):
        room = rd.choice(self.classRoom)

    new[_id].append([subjectCode, room, weekDays, shift])

```

Hình 16: Đột biến kiểu thêm vào

- Xóa bớt một nhiễm sắc thể (1): Random nhiễm sắc thể và xóa.

```

if (typeMutation == 1):
    # đột biến kiểu xóa bớt
    while (len(new[_id]) <= 1):
        _id = rd.choice(lecturerId)
        removeId = rd.randint(0, len(new[_id]) - 1)
        new[_id].pop(removeId)

```

Hình 17: Đột biến kiểu xóa bớt

- Thay đổi (2):
 - Random thứ và ca dạy.
 - Đảm bảo không bị trùng ca dạy đã có.
 - Random mã môn học, phòng.

- Đảm bảo không bị trùng phòng trong một ca của một thứ.
- Đổi list mới vào một list trong new.

```

elif (typeMutation == 2):
    # đột biến kiểu thay đổi
    haveShifts = [[x[2], x[3]] for x in new[_id]]
    shift = self.myRandom(0)
    weekDays = self.myRandom(1)

    while ([weekDays, shift] in haveShifts):
        shift = self.myRandom(0)
        weekDays = self.myRandom(1)

    subjectCode = rd.choice(self.lecturers[_id])
    room = rd.choice(self.classRoom)

    while (room in scheduleRoom[shift][weekDays]):
        room = rd.choice(self.classRoom)

    new[_id][rd.randint(0, len(new[_id]) - 1)] = [subjectCode, room, weekDays, shift]

```

Hình 18: Đột biến kiểu thay đổi

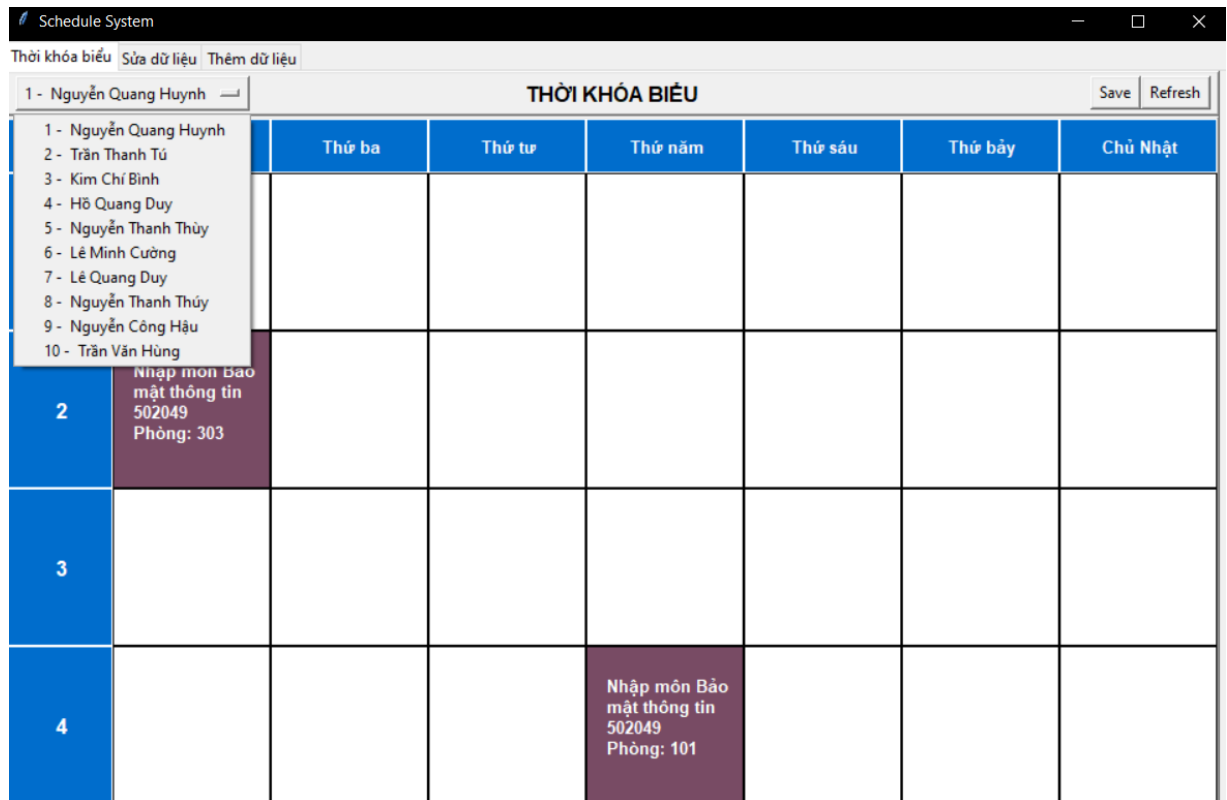
Thêm new vào quần thể:

self.populations.append(new)

CHƯƠNG 3 – KẾT QUẢ

Giao diện chính THỜI KHÓA BIỂU gồm:

- Option menu: xem thời khóa biểu của từng giáo viên.
- Button Refresh: để làm mới lại lịch khi không ưng ý.
- Button Save: để save lại thời khóa biểu của giảng viên ra file schedule.csv.



Hình 19: Giao diện chính THỜI KHÓA BIỂU

Giao diện SỬA DỮ LIỆU gồm:

- Form GIÁO VIÊN:

- Mã giáo viên: chứa IDGV đã bị disable để không sửa được.
- Tên giáo viên: chứa TENG V, có thể sửa.
- Mã môn học: chứa các MaMH mà GV đó dạy, có thể sửa.

- Form MÔN HỌC:

- Mã môn học: chứa các MaMH đã bị disable để không sửa được.
- Tên môn học: chứa TENMH, có thể sửa.
- Số tiết: chứa số tiết MH, có thể sửa.
- Số lớp: chứa số lớp MH mở, có thể sửa.

The screenshot displays the 'SỬA DỮ LIỆU' (Edit Data) window of the 'Schedule System'. The window has a title bar with standard OS controls. Below the title bar is a navigation bar with three tabs: 'Thời khóa biểu', 'Sửa dữ liệu' (active), and 'Thêm dữ liệu'. The main content area is divided into two sections. The first section, titled 'GIÁO VIÊN', has a dropdown menu showing '1 - Nguyễn Quang Huynh' and 'Save'/'Delete' buttons. Below this are three input fields: 'Mã giáo viên' (Teacher ID) with value '1', 'Tên giáo viên' (Teacher Name) with value 'Nguyễn Quang Huynh', and 'Mã môn học' (Subject Code) with value '502049;502057'. The second section, titled 'MÔN HỌC', has a dropdown menu showing '501049 - Nhập môn Phân tích kinh doanh' and 'Save'/'Delete' buttons. Below this are four input fields: 'Mã môn học' (Subject Code) with value '501049', 'Tên môn học' (Subject Name) with value 'Nhập môn Phân tích kinh doanh', 'Số tiết' (Number of lessons) with value '45', and 'Số lớp' (Number of classes) with value '2'.

Hình 20: Giao diện SỬA DỮ LIỆU

Giao diện THÊM DỮ LIỆU gồm:

- Form GIÁO VIÊN:

- Mã giáo viên: chứa IDGV tự động tăng và có check IDGV tồn tại.
- Tên giáo viên: chứa TENG V.
- Mã môn học: chứa MaMH GV đó dạy.

- Form MÔN HỌC:

- Mã môn học: chứa các MaMH và có check MaMH tồn tại.
- Tên môn học: chứa TENMH.
- Số tiết: chứa số tiết MH và có check sotiet ≥ 3 .
- Số lớp: chứa số lớp MH cần mở và có check solop ≥ 1 .

Schedule System

Thời khóa biểu Sửa dữ liệu Thêm dữ liệu

THÊM DỮ LIỆU

GIÁO VIÊN

Mã giáo viên: 11

Tên giáo viên:

Mã môn học:

Save

MÔN HỌC

Mã môn học:

Tên môn học:

Số tiết: 0

Số lớp: 0

Save

Hình 21: Giao diện THÊM DỮ LIỆU

TÀI LIỆU THAM KHẢO

Tiếng Việt

1. https://vi.wikipedia.org/wiki/Ti%E1%BA%BFn_h%C3%B3a
2. https://vi.wikipedia.org/wiki/Di_truy%E1%BB%81n
3. <https://viblo.asia/p/thuat-toan-di-truyen-ung-dung-giai-mot-so-bai-toan-kinh-dien-phan-1-RQqKLxJzK7z>
4. <https://viblo.asia/p/thuat-toan-di-truyen-ung-dung-giai-mot-so-bai-toan-kinh-dien-phan-2-07LKXwYp5V4>