**The problem that we are trying to solve:**

This project aims to schedule the courses of ECE department in Ohio State University automatically given the following constraints and information related to the offered courses:

**Basic constraints:**

1. The same instructor can’t teach 2 different courses simultaneously.
2. The same room can’t be scheduled for 2 different courses simultaneously.
3. The room scheduled for a specific course must have no less seats than the corresponding course capacity.

**Advanced constraints:**

1. For the courses with course number in the range of 3000~5999, the 3rd digit of the course number represents its study area (e.g. 0-Comm/DSP, 1-Emag, 2-Circuits, 3-SSEP, 4-Power, 5-Control, 6-Computer). The courses with the same study area can’t be scheduled in the same time slots.
2. For the courses with course number in the range of 5000~8999, the 3rd digit of the course number represents its study area (e.g. 0-Comm/DSP, 1-Emag, 2-Circuits, 3-SSEP, 4-Power, 5-Control, 6-Computer). The courses with the same study area can’t be scheduled in the same time slots.
3. Some instructors may have preference of time to teach their courses (e.g. Professor Lee may only have time to teach the course on 10AM-MWF).
4. No more than 2 large volume courses can be scheduled at the same time.
5. No more than 1 extra-large volume course can be scheduled at the same time.

**Information related to the offered courses:**

1. Course: the course is classified by the OSU course ID system, e.g. 5421. Usually the course ID is represented as 4-digit number, the first digit represents course difficulty, the third digit represent course field.
2. Class: a class is a real scheduling object we are dealing with, a course can have multiple classes, taught by different or same instructor.
3. Time Slot: all the available time a course can choose
4. Instructor: a class need an instructor to teach, sometimes we don't know who will be teaching the course, some time we know the instructor will be teaching multiple course, some time the instructor requires some preferred teaching time, we need to consider all the situation
5. Classroom: a classroom is a physical container of the class, we need to know the classroom capacity, classroom type (e.g. Lecture or Lab, does it include computers, etc.)

**The basic idea of this project:**

A course schedule should include the arrangement for all courses offered in one semester, and the arrangement of each class contains the related instructor to teach the course, the room for teaching the course, and the time to teach the course.

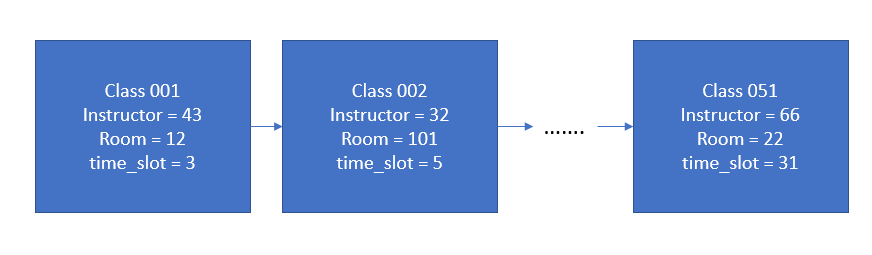


Figure 1 a single chromosome

Each course schedule is treated as a single chromosome in the genetic algorithm, and all the constraints (requirements) of the course schedule is coded to the fitness function.

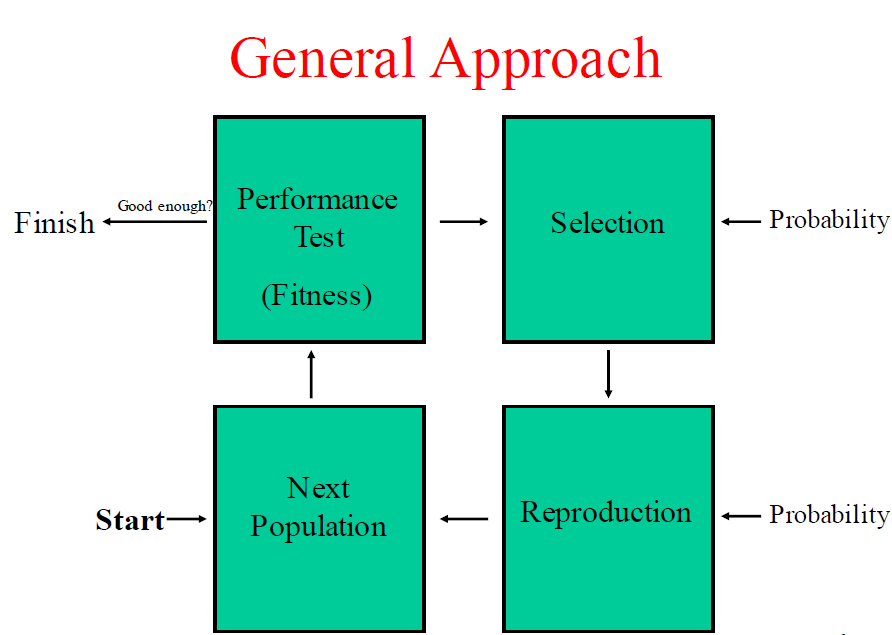


Figure 2

In the initial state, the program would randomly generate a group of chromosomes, each of the chromosome is a possible course schedule, then the fitness score of each chromosome is calculated through the fitness function in the performance test process. The fitness score is a numeric metric used to measure how well the current chromosome (solution of course schedule) satisfies the constraints, the higher the better (in our case, the fitness score ranges in [0,2], 2 is a “perfect” solution regarding the constraints).

The process after performance test is selection, it is a probability-based process. In this process, a roulette wheel selection method is used to choose parent chromosomes in the current generation to mate and generate new chromosomes (i.e. new possible solution of the course schedule) in the following reproduction process.

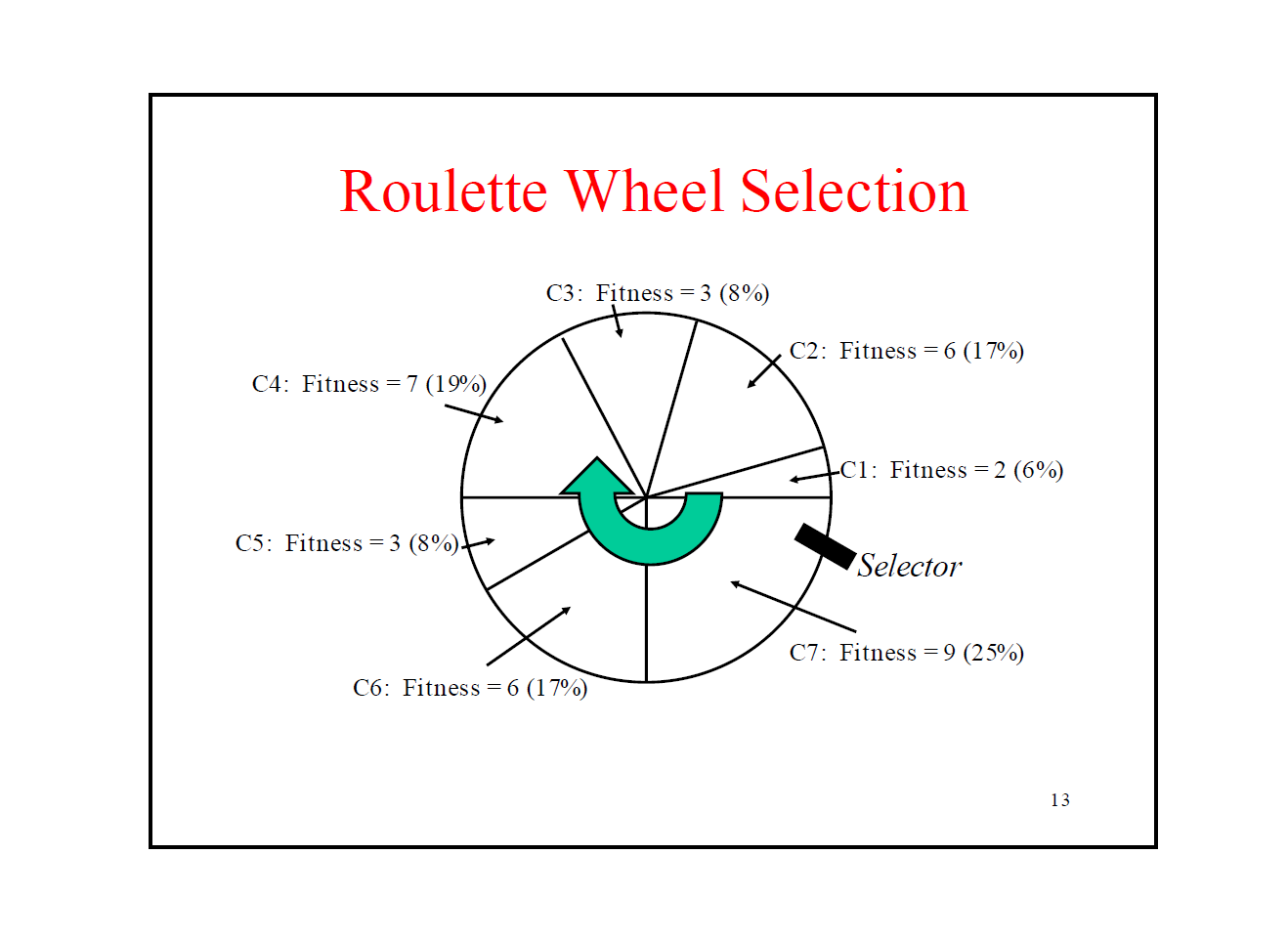


Figure 3 Roulette Wheel Selection

Fig 3 showed the Roulette wheel selection process, which is a probability-based selection. The chromosome with higher fitness score has higher odds to be selected as the parent chromosome to generate new chromosomes for the next generation.

After selection of the parent chromosomes, a mating process is applied to the parent chromosomes to generate new chromosomes.

In our approach, an adaptive mating strategy is used to generate new chromosomes. In the mating process, our program will randomly select some number of points in the chromosome for exchanging purpose. Each point in the chromosome represents a certain class with its arrangement. The number of exchanging points is based on the chromosome’s fitness score in the previous performance test (i.e. the higher fitness score of the parent chromosome, the less number of exchanging points is used, this is similar to adaptive gradient descent, the closer you get to the local optima, the smaller the iterative step becomes).

For example, if we are applying a 1-point mating process to the parent chromosomes, then the program would randomly choose a class (the cross over point), and the mating process will exchange either time\_slot or room of this class between the parent chromosomes to generate 2 new chromosomes as their children chromosomes for the next generation. The following figure is an example of 2-point mating process.

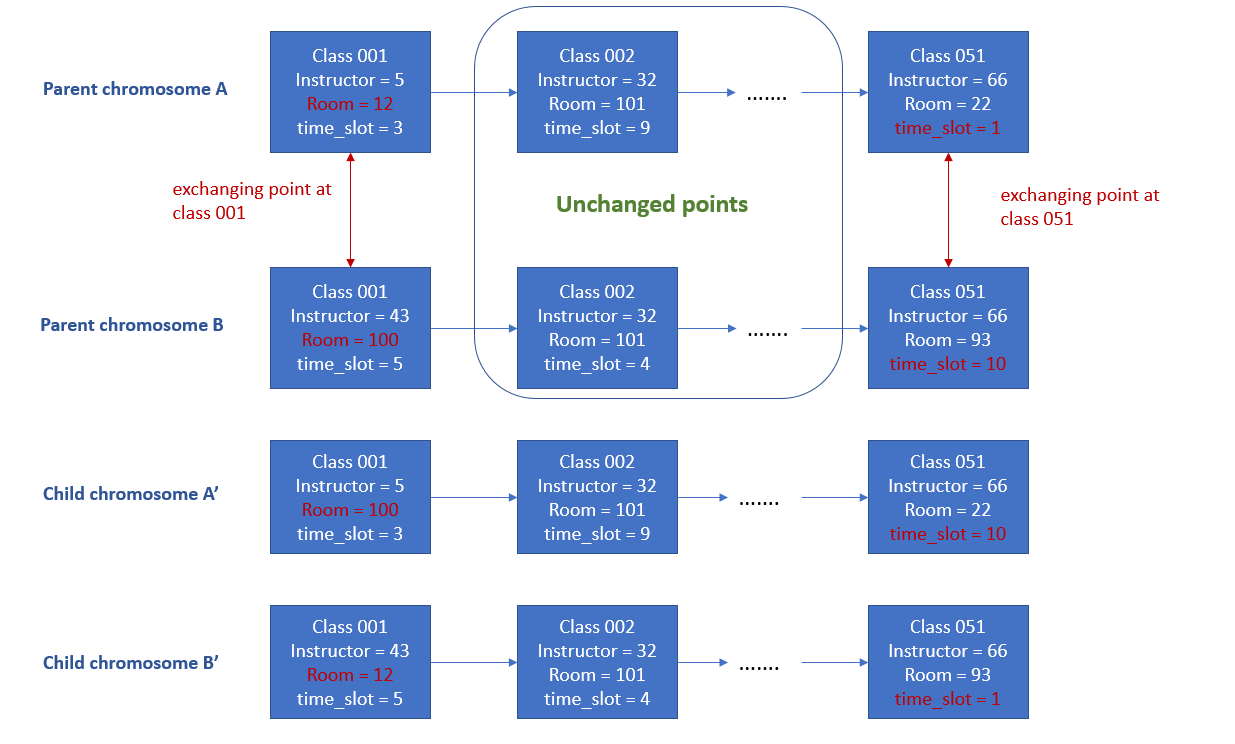


Figure 4. 2-point mating process

Fig 4 showed a 2-point mating process, the 1st exchanging point is class 001, and room arrangement is randomly chosen to be swapped between parent chromosomes, and the 2nd exchanging point is class 051, the time arrangement of this class is randomly chosen to be swapped. In this example, the length of chromosome is 51, it includes course schedule of 51 classes, and the 2 exchanging points is randomly chosen as the first and the last points.

After the reproduction process, the new generation of chromosomes are passed to performance test again and the whole process becomes iterative as shown in figure 2. Once the performance test found a “perfect” chromosome or the iteration has reached a user-defined maximum iteration times, the whole process will stop and output the best chromosome found so far.

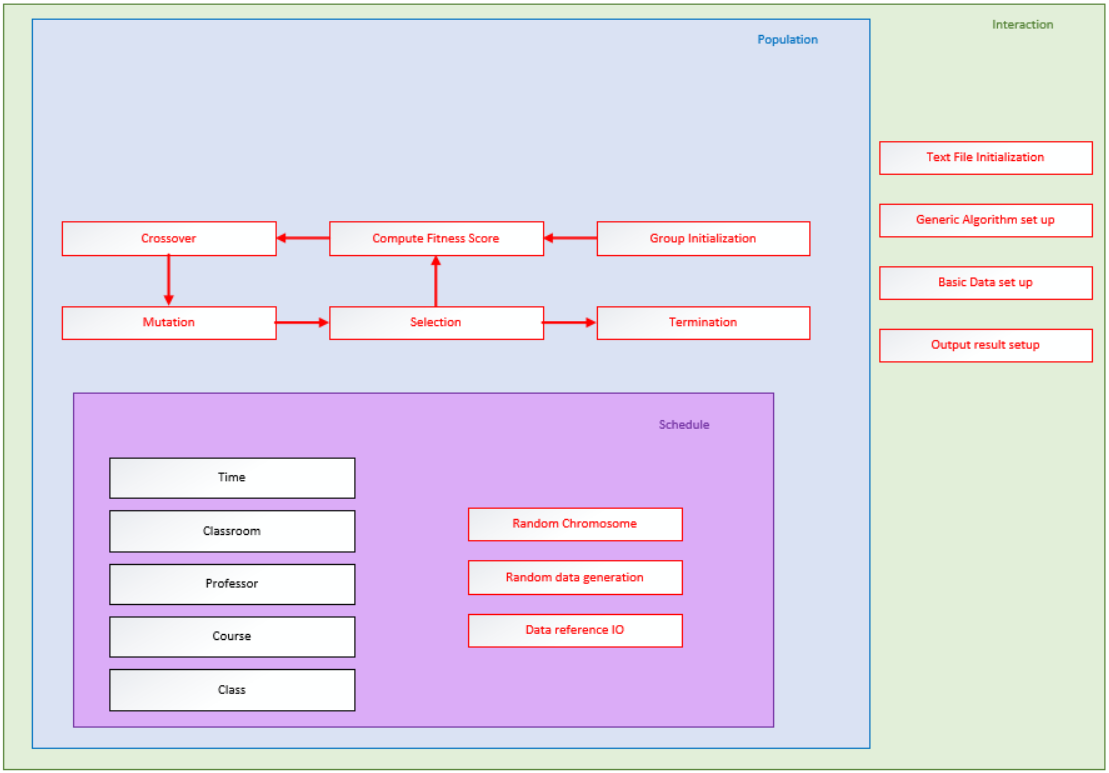
**The work flow of our program:**



**The current progress:**

We’ve implemented basic constraints (1), (2), and all advanced requirements.

**The basic framework of our C++ program:**



**The GitHub website:**

<https://github.com/geneticAlg/CourseSchedule_GA>