Course Allocation System

Lokesh Venkatachalam
Undergraduate Researcher
Center for Security, Theory, Algorithms and Research (CSTAR)
International Institute of Information Technology - Hyderabad
Hyderabad, India
lokesh.v@research.iiit.ac.in

***Abstract*—Course allocation the problem of allocating seats in university courses satisfying the upper bound of number of students, the credits of the student in a given semester in manner. such that maximum numbers of students get the courses they have preferenced.**
This paper proposes a novel software System to course allocation where students can make informed decisions about the course preferences, administration and faculty can allocate the courses such that maximum number of students are happy.

I. Introduction

Course allocation has two major components, students giving the preferences and faculty,administration allocating the courses.
Students usually gather information from seniors about course content, gradings,professor etc But the information about previous allocation is not well known among students, which makes it tougher for them to make informed decisions.
So in the proposed software system there will information about

- Proposed course content, grading policy.
- Earlier course content, grading policy if exists
- Earlier course allocation, students preference data

Many courses in university have a upper bound on number of students who can take it. Some courses might also have lower bound on number of students for the course to proceed. Student have also have to satisfy the minimum number of credits for a semester. In certain universities students are given option to give more than preference for a given course type, but in other universities students are allowed to give only one preference for a given course type in a given round of allocation In most cases universities use simple first come first serve (FCFS) is used to allocate courses, which puts students without good internet, computer system at time of portal opening,etc on a disadvantage. It becomes like a lottery at the end. It also neither stable nor statergy proof.

So in this software system we are using the course allocation by stable matching algorithm **efficiency adjusted deferred acceptance mechanism (EADAM)**. This algorithm is a modification of the **Gale-Shapley student optimal stable mechanism (SOSM)** where the efficiency loss due to randomised tie-breaking is improved but at the expense of statergy-proofness[1]. This algorithm gives more no of students their higher order of their preferences.

II. Literature Review

Allocation algorithms can be judges based on three major parameters.

- **stability** - describes whether there exists a mapping where both students and teacher is betteer off
- **strategy-proofness** - A algorithm is said to be statergy proof if the students have no incentive to not give their true preference
- **efficiency** - describes whether there can be another matching which is better to more students or faculty than one proposed by algorithnm at cost of few people.

There is no existing algorithm which is STABLE, STATERGY PROOF and very efficient. All algorithms have some drawback in atleast any one of the three parameters.

- **First Come First Serve(FCFS)** is the most preffered algorithm, but is neither STABLE nor STATERGY PROOF.
- **Gale-Shapley student optimal stable mechanism (SOSM)** is STABLE and STATERGY PROOF, but more efficient algorithms exist[1].
- **Efficiency Adjusted Deferred Acceptance mechanism(EADAM)** is STABLE and more efficient the SOSM but not statergy proof[1].

III. System Architecture

In this section,I propose a software system for course allocation which will give basic information about each course to the students and an automatic course allocation based on students preferences reducing the manual work done by faculty and administration.

A. Users

There are mainly there types of users.

- **Administrators**
  Users with highest level of access. admin users job is add new students, faculty accounts to software system, update the course details, update the credit requirements for students before each semester, updating the time frame for course preferencing, running the course allocation algorithm.

- **Faculty**
  Users with access to list the courses, give details about the courses, and have access to previous data on student allocation.
- **Student**
  Students are the basic users who have the access to give one to three preferences to each course slot and also have access to view course content and previous allocation data

B. Data

1) Course Type
   - Name ex: Open elective, CS open elective , ECE open elective, Science , Math , Humanities, etc.
   - credit: No of credits
2) Course
   - Course Type
   - Course Code
   - Course Name
   - Course Description - Outline,Area,Prerequisites needed,recommended book, material etc
   - Previous final course grading scheme
   - Capacity
3) Course Allocation Data contains:
   - Course Name
   - Year
   - Semester
   - Course Capacity of the year
   - Faculty of the year
   - No of students admitted as compulsory in round 1
   - No of first preference in round 2
   - No of second preference in round 2
   - No of third preference in round 2
   - No of first preference students allotted in round 2
   - No of second preference students allotted in round 2
   - No of third preference students allotted in round 2
   - No of first preference in round 3
   - No of second preference in round 3
   - No of third preference in round 3
   - No of first preference students allotted in round 3
   - No of second preference students allotted in round 3
   - No of third preference students allotted in round 3
   - Faculty requests
   - No of faculty requests accepted
   - total no of students who took the course at end.
   - No of F grades.
   - No of A grades.
4) User
   - User Email-ID
   - User Type

C. workflow

- Credit requirements of students updated based on previous semester grades and students current semester requirements.
- Parallely faculty start to list the courses approved by dean/department head in course allocation software system with relevant details.

- After courses are listed, the lower bound (if exists) and upper bound on number students allowed to take the course are updated along with the previous years data of the course allocation for the courses listed.
- Dates of choosing course preference are announced by administration after courses are listed.
- First round of course allocation occurs where students are given the compulsory course they have to choose based on research/honours area.
- The latest bound on course intake is updated after round one.
- Round two starts, where students give three preferences for each course slot.
- Round two allocation occurs.
- Students who were allotted there third preference are given a chance to choose or reject the course allocation.
- Faculty taking a course can request the dean to allocate his course to few students who have not been allotted until end of round 2.
- Round three starts, where students fill three preferences for each course slot which have not be allocated in round two.
- Round three allocation occurs.
- Round four is manuall allocation to students who still have not been fulfill credit requirements after round three.
- With round 4 course allocation ends.

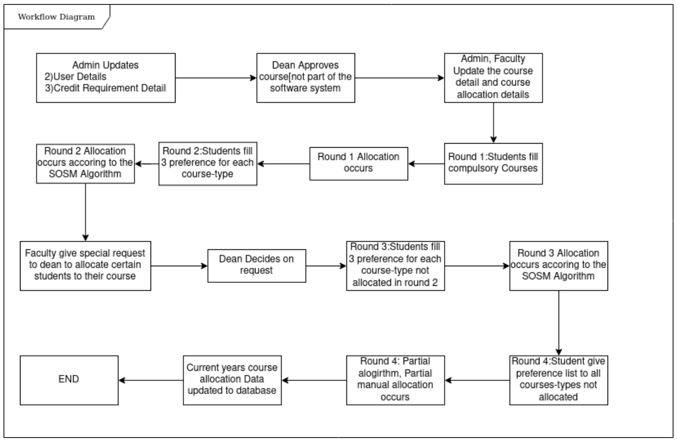
Workflow Diagram
Admin Updates
2)User Details
3)Credit Requirement Detail
Dean Approves course[not part of the software system
Admin, Faculty Update the course detail and course allocation details
Round 1:Students fill compulsory Courses
Round 1 Allocation occurs
Round 2:Students fill 3 preference for each course-type
Round 2 Allocation occurs accoring to the SOSM Algorithm
Faculty give special request to dean to allocate certain students to their course
Dean Decides on request
Round 3:Students fill 3 preference for each course-type not allocated in round 2
Round 3 Allocation occurs accoring to the SOSM Algorithm
Round 4:Student give preference list to all courses-types not allocated
Round 4: Partial alogirthm, Partial manual allocation occurs
Current years course allocation Data updated to database
END

Fig. 1. Class UMl diagram Link

D. Use Cases

1) Authentication
   - Admin creating new accounts with credentials.
   - Student and Faculty logging in via CAS,
2) Data updating
   - Admin updating course capacity
   - Admin updating the previous years allocating data.

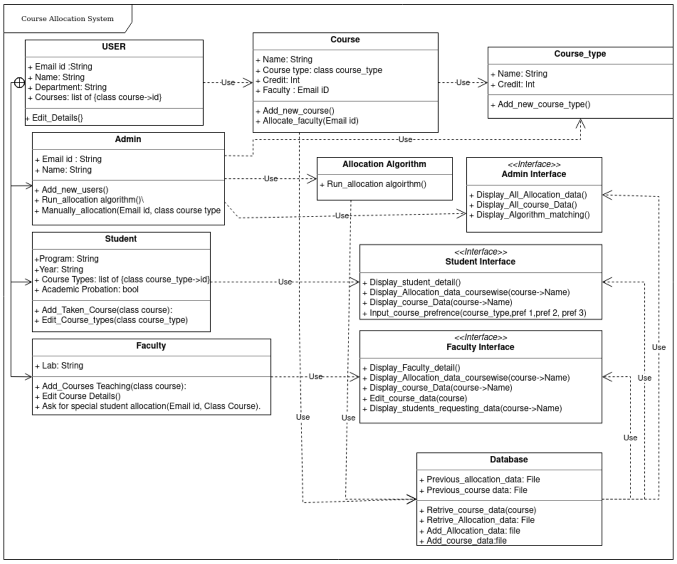
Course Allocation System
USER
+ Email id :String
+ Name: String
+ Department: String
+ Courses: list of {class course->id}
+ Edit_Details{}
Course
+ Name: String
+ Course type: class course_type
+ Credit: Int
+ Faculty : Email iD
+ Add_new_course()
+ Allocate_faculty(Email id)
Course_type
+ Name: String
+ Credit: Int
+ Add_new_course_type()
Admin
+ Email id : String
+ Name: String
+ Add_new_users()
+ Run_allocation algorithm()\
+ Manually_allocation(Email id, class course type
Allocation Algorithm
+ Run_allocation algoirthm()
<<Interface>>
Admin Interface
+ Display_All_Allocation_data()
+ Display_All_course_Data()
+ Display_Algorithm_matching()
Student
+Program: String
+Year: String
+ Course Types: list of {class course_type->id}
+ Academic Probation: bool
+ Add_Taken_Course(class course):
+ Edit_Course_types(class course_type)
<<Interface>>
Student Interface
+ Display_student_detail()
+ Display_Allocation_data_coursewise(course->Name)
+ Display_course_Data(course->Name)
+ Input_course_prefrence(course_type,pref 1,pref 2, pref 3)
Faculty
+ Lab: String
+ Add_Courses Teaching(class course):
+ Edit Course Details()
+ Ask for special student allocation(Email id, Class Course).
<<Interface>>
Faculty Interface
+ Display_Faculty_detail()
+ Display_Allocation_data_coursewise(course->Name)
+ Display_course_Data(course->Name)
+ Edit_course_data(course)
+ Display_students_requesting_data(course->Name)
Database
+ Previous_allocation_data: File
+ Previous_course data: File
+ Retrive_course_data(course)
+ Retrive_Allocation_data: File
+ Add_Allocation_data: file
+ Add_course_data:file
Use

Fig. 2. Class UMl diagram Link

- Admin updating the credit requirements of the students.
- Admin updating the time frame for course preferencing.

3) Course listing
    - Admin/Professor listing the courses approved by dean/department head.
    - Admin/Professor updating content of course information.
4) Course preferencing
    - Students choose their course preference.
    - Student ability to view the course information, previous allocation data.
5) Course Allocating
    - Admin runs the course allocation algorithm and allocates the course
    - Admin/Professor can request the dean to allocate the courses to few students who have not been allotted until end of round 2.
6) Rounds
    - The ability to run more than two rounds of allocation

IV. Conclusion and Future Work

The proposed software system makes it easier for administration to allocate courses and provides a more fairer chance in getting the wanted course.

We can options to give weightage to the preferenced courses, add option to see live course requesting data for students to see the demand.

Using the data collected from the allocation over period of years universities can work towards satisfying the demand each course gets.

The data can also be used for betterment of allocation algorithm.

References

[1] Franz Diebold, Haris Aziz, Martin Bichler, Florian Matthes, and Alexander Schneider. Course Allocation via Stable Matching. Business Information Systems Engineering, 6(2):97110, Apr 2014.