Application of Scheduling Algorithms in College Admission Problem using Bipartite Graph

Dr. H. B. Walikar[1], Ravikumar H. Roogi[2], Shreedevi V. Shindhe[3], Ishwar. B[4]

#1 Prof. Dept of Computer Science, Karnatak University, Dharwad

#4 Prof. Dept of Computer Science, Karnatak University, Dharwad,

#2, #3 Research Scholars. Dept of Computer Science, Karnatak University, Dharwad

Abstract:

The admission in college is perhaps the simplest model can be solved by using scheduling algorithm applications and graph theory approach using bipartite graph. We study the admission problem purely on merit bases. Each applicant applies for the admission in college of their 3 choices departments in which they wanted to take admission. College prepares a merit list and conduct counseling by giving a token number to the applicants. Our aim is every applicant should get admission of their 1st or 2nd or 3rd choices.

Keywords: Scheduling algorithm, bipartite graph, waiting time.

1. Introduction:

Suppose there are n students who are applying for colleges and there are k colleges that these students can apply for. Each student has a strict preference ordering over all colleges, and each college also has a strict preference ordering over all students. By strict preference, it is impossible for a college to accept all the students who apply for it, due to limited resources. In fact, a college only accepts a specific number of students (quotas) in each academic year. So every student cannot possibly get into their top choices. On the other hand, a student also accept offer of admission from only one college. Thus, it is not guaranteed that all students whom a college had made an offers. This problem is called college admissions problem. The college admissions problem has been widely studied by economists and game theorists. It is well-known to be closely related to the stable marriage problem. Instead of trying to find matching between students and colleges, the stable marriage problem tries to find matching between males and females in the community, such that each couple cannot be better off by pairing with other people in the community [1].

The college admission problem is first studied by Gale and Shapely (1962) in a seminar paper in which they proposed the now well-known deferred-acceptance algorithm. Many variants and extensions of the original model with useful applications have been studied (Knuth (1976), Roth (1984), Gustfield and Irving (1989), Roth and Sotomayor (1990), Roth (2002), Abdulkadiroglu and Sonmez (2003), Abdulkadiroglu (2005a, b) Roth et.al. (2007), Roth(2008)). It is only recently that the college admission problem with an entrance criterion has been studied (Perach, Polak and Rothblum (2007), Perach and Rothblum (2010). They design an algorithm that respects eligibility criteria and produces a weakly stable outcome. They also study incentive compatibility properties [2].

Quick Sort:

The quick sort algorithm works by partitioning the array to be sorted, then recursively sorting each partition. In below algorithm, one of the array elements is selected as a pivot value. Values smaller than the pivot value are placed to the left of the pivot, while larger values are placed to the right [3]. Here in this paper we use vice versa instead of increasing order we go by decreasing order.

int function *Partition* **(**Array *A***,** int *Lb***,** int *Ub***)**;

begin

select a *pivot* from *A[Lb]…A[Ub];*

reorder *A[Lb]…A[Ub]* such that:

all values to the left of the *pivot* are ≤ *pivot*

all values to the right of the *pivot* are ≥ *pivot*

return *pivot position;*

end;

procedure *QuickSort* **(**Array *A***,** int *Lb***,** int *Ub***)**;

begin

if *Lb* < *Ub* then

M = Partition (A, Lb, Ub);

QuickSort (A, Lb, M – 1);

QuickSort (A, M + 1, Ub);

end;

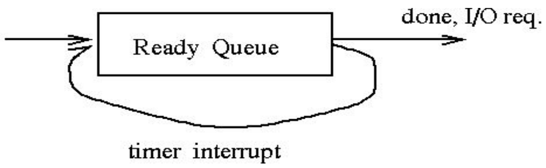
Basic Scheduling Algorithm:

FCFS - First-Come, First-Served [4]

- Non-preemptive
- Ready queue is a First In First Out queue
- Jobs arriving are placed at the end of queue
- Dispatcher selects first job in queue and this job runs to completion of CPU burst.

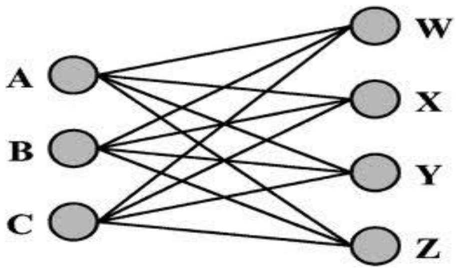
RR - Round Robin [4]

- Preemptive version of First Come First Serve
- Treat ready queue as circular
- arriving jobs are placed at end
- dispatcher selects first job in queue and runs until completion of CPU burst, or until time quantum expires
- if quantum expires, job is again placed at end.



Bipartite graph:

In the mathematical field of graph theory, a **bipartite graph** (or **bigraph**) is a graph whose vertices can be divided into two disjoint sets U and V such that every edge connects a vertex in U to one in V; that is, U and V are independent sets. Equivalently, a bipartite graph is a graph that does not contain any odd-length cycles [5].



In this paper, we explain how to solve college admission problem using Scheduling Algorithm and its graphical representation using bipartite graph. Suppose there are n students applying for admission in college for k departments. Each student given 3 choices of departments which they can apply for admission. We use Quick Sort method to make a merit list of students and assign token numbers for students according to their ranks, then we conduct a counseling and allocate seats to students according to token number by using First-come, First-served Scheduling algorithm and also showed by using Round-Robin Scheduling algorithm and represented this problem in Bipartite graph in section 2 and we conclude paper in section 3.

2. Proposed work:

Given a set no. of students $S = \{s_1, s_2, s_3, \dots \dots ., s_n\}$ and a set no. of Department $D = \{d_1, d_2, d_3 \dots ., d_k\}$, let $S \times D$ denote the set of all possible pairs of the form(s, d), where $s \in S$ and $d \in D$. Each students applies for the departments consedering the applications that student can apply to three departments at a time by their choices. The goal is to find that every student should get admission, if not their first choice they can get through their second or third choices which ever they get a chance. The advantage of this is to try to get all students an admission and here students need not to apply new application for other departments because here we allowed them in one application can apply for three departments of their choices. Here seats are not given according to student first choices but based on students scores. According to students score a merit list is created and ranks are given and token number are assigned according to students ranks. A counseling is conducted and based on token number seats are allocated to every students. If any student is absent when his token number is called in counseling then they get a chance in second round counseling.

In College, we have many sections like Arts, Social Science, Science, Commerce, Law, Education, Management courses and many one year diploma courses. Each sections having many departments, here in this problem we take one section i.e. Science. In Science we have many departments like Computer Science, Mathematics, Statistics, Chemistry, Biotech, Microbiology, Genetics, Physics, Electronics, Zoology, Botany, and Geography. Here students apply in science section for admission.

Given a set no. of students $S = \{s_1, s_2, , s_3, , \dots \dots ., s_n\}$ and a set no. of Department $D = \{d_1, d_2, d_3 \dots ., d_k\}$, each student has a choice of selecting three departments and each department has limited seats that are no. of students it can accept in academic year. However, each student can accept only one course for admission. The goal is to find that every student should get admission based on their three choices. Based on the score, on purely merit system seat is allocated to the students.

Consider the following set of students applied for admission according to their scores as shown in Table 1. Student Applications for departments Computer Science- CS, Mathematics- M, Statistics- S, Physics- P, Chemistry- C, Botany- B

Table. 1: Student Applications

Application number	1	2	3	4	5
Student Names	s_1	s_2	s_3	s_4	s_5
Department	CS, M, C	P, M, C	C, S, M	CS, S, B	P,C, B
Score	430/500	425/500	426/500	427/500	420/500

The first algorithm to sort the list is quick sort. With this scheme we sort list of Students considering their total marks and assign a rank and give token number. Here instead of increasing order, we go in decreasing order. Table .2 Sorting list

Table. 2 (a)

430	425	426	427	420

Pivot

Table. 2 (b) Lb Ub

430	425	426	427	420

Pivot

Table. 2 (c)

430	427	426	425	420

In table. 2, the pivot selected is 426. Indices are run starting both ends of the array. One Indices starts on the left and selects an element that is smaller than the pivot, while other indice starts on the right and selects an element that is larger than pivot. In this case number 430 and 420 is not selected because it is already larger and smaller then the pivot. So it selects 425 and420, these elements are then exchanged as shown in table.2 (c). This process repeats until all elements to the left of the pivot are $\geq$ the pivot, and all items to the right of the pivot are $\leq$ the pivot. Quick Sort recursively sorts the two sub- arrays, resulting in the array as shown in Table.2. As the process, it may be necessary to move the pivot so that correct ordering is maintained. In this manner, Quick Sort succeeds in sorting the array. Here the list is sorted and the merit list is ready, we assign a token number from starting serially.

After assigning token number to all students, we use the simple first- come, first-served (FCFS) scheduling algorithm to allocate seats to the students by conducting counseling. With this scheme, in the counseling request 1st token number student and allocate seat of their 1st choice. The implementation of the First-come, First-serve is easily managed with first-in, first-out queue. When a student enters into the queue for seat, if their 1st choice seat available it is allocated or 2nd or 3rd choice at the head of the queue. Once the seat allocation is done student is removed from the queue.

Consider the following set of students that arrive at time 0 along with time allocation to each student in minutes as shown in table.3.The main constraints for allocation of time to each student are verification of marks cards, income certificate, etc. Note that whatever constraints mentioned here is just author's opinion and it is not mandatory. If the student arrived in the order according to the token number and are served in First-come, First-serve order, we get the result shown in the Figure.1 and if any student absent when their term comes, they have to wait for 2nd round counseling.

Table.3: Counseling Schedule

Application Number	1	4	3	2	5
Student Names	s_1	s_4	s_3	s_2	s_5
Counseling time taken	5	7	8	10	6

The waiting time for the 1st student S_1 is 0 minutes, 5 minutes forS_4, 12 minutes forS_3, 20 minutes for S_2 and 26 minutes forS_5. The average waiting time is 6 minutes. This algorithm is non-preemptive.

s_1	s_4	s_3	s_2	s_5
0	5	12	20	30

Figure.1: Gantt chart of First-come, First-serve algorithm.

The Round Robin algorithm is designed especially for time sharing system. Round Robin is similar to First-come, First-serve but preemptive is added to switch between students. A small unit of time called time quantum [6] is defined. In process of verification checking, there is all semester marks card verification, income verification, caste verification etc. In above algorithm we have to do these all steps within time of allocation of students. Wasting of time is possible while using above discussed algorithm in each step of verification. In this method we can avoid wasting of time between each step [7]. In this case, we give break to student between each step namely time quantum so that simultaneously we can work with more than one student at a time. This leads to saving student time, counseling time and possibility of verifying more students. Here we can minimize average waiting time of students as shown in figure.2, in the following chart the time quantum is 4 units.

s_1	s_4	s_1	s_3	s_1
0	5	10	15	20

Figure.2: Gantt chart for Round Robin algorithm

Here Shortest Job First algorithm and Priority Scheduling algorithm is not applicable for this problem. Since the seats are allocated on purely merit bases to students, the priority is not given to any students like women quota, caste quota, sports quota, etc for allocation. Till student comes to counseling, how much time taken by each student in counseling is not known so cannot tell which student takes less time and which student takes more time so shortest job first algorithm is not applicable.

In summary, the above discussed algorithms having advantages as well as disadvantages but the best suitable algorithm is Round Robin algorithm.

Graph theory plays an important role in this problem. In this work, the admission problem is approached using the simpler class of bipartite graph. Given a set no. of students $S = \{s_1, s_2, s_3, \dots \dots, s_n\}$ and set no. of Dept $D = \{d_1, d_2, d_3 \dots, d_k\}$, for '$S$' students based on their score the available number of seats in 'D' departments to be allocated. This is done as follows, a bipartite graph (or bigraph whose vertices can be divided into two disjoint sets U and V such that every edge connects a vertex in U to one in V; that is, U and V are independent sets.) G where the vertices are the numbers of students say $\{s_1, s_2, s_3, \dots \dots, s_n\}$ and numbers of Departments say $\{d_1, d_2, d_3 \dots, d_k\}$ such that vertices are connected by edges, where edge represents choices of students to departments. For example consider the Table. 1

In table. 1 there are 5 students namely $S = \{s_1, s_2, s_3, s_4, s_5\}$ and 6 departments namely D={Computer Science- CS, Mathematics- M, Physics- P, Statistics- S, Chemistry- C, Botany- B }. The bipartite graph is constructed as follows in figure. 3

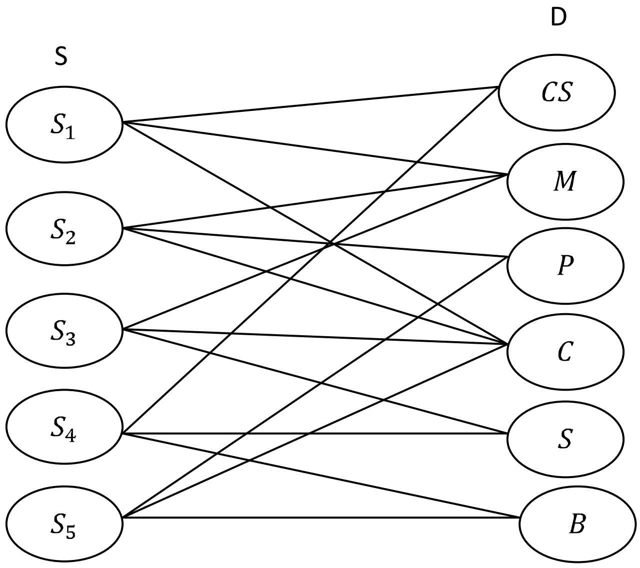


Figure.3: Bipartite graph with 5 students and 6 departments

All the students need not give 3 choices compulsory. They can give 1 or 2 or 3. From the graph we see that the total no. of vertices in set (S, D) gives the total no. of applications. The degree of each vertex in the set (S, D) specifies the student's subject choice. The degree of each vertex in the set (D, S) gives the application received for that particular course.

Algorithm to find the students choices

```
for s ← 1 to n
  for d ← 1 to k
    if (SD [s][d]= = 1)
       deg [s] ++ ;
```

	d_1	d_2	d_3	d_4	d_5	d_6
s_1	1	1	0	1	0	0
s_2	0	1	1	1	0	0
s_3	0	1	0	1	1	0
s_4	1	0	0	0	1	1
s_5	0	0	1	1	0	1

Figure. 5 student choice matrix

The following example will illustrate the result of this section. Let the choices be given by:

$S_1: CS, M, C$ $\qquad$ $CS: S_1, S_4$

$S_2: P, M, C$

$S_3: C, S, M$

$S_4: CS, S, B$

$S_5: P, C, B$

$M: S_1, S_2, S_3$

$P: S_2, S_5$

$C: S_1, S_2, S_3, S_5$

$S: S_3, S_4$

$B: S_4, S_5$

After merit list the final outcome given by:

$S_1: CS$

$S_4: S$ (assume 1st choice are full so 2nd choice)

$S_3: C$

$S_2: P$

S_5: B (assume 1st choice and 2nd choice are full so 3rd choice)

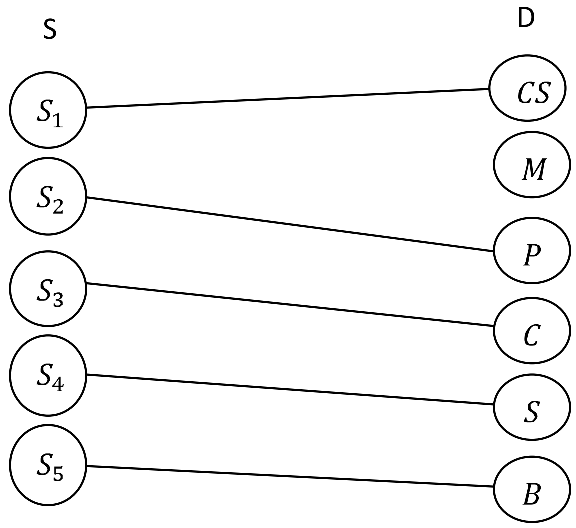


Figure. 4: Solution to allocation of seats

After the seat allocation we get the graph as shown in figure. 4. Here if the degree of any student vertex is zero then it means that particular person did not take admission in any department. The degree of department vertex shows the no. of students admitted for that department. If degree of any department is zero, which means no one took admission for that particular department. By adding the degrees of each vertex in department sets we get the total no. of students admitted in the university.

Algorithm to find the number of students in departments

```
for s ← 1 to n
  { for d ← 1 to k
      if adm [s][d]=1
        { deg [d] ++   //total no. of students admitted in dept
          deg ad [s] ++
          break
```

	d_1	d_2	d_3	d_4	d_5	d_6
s_1	1	0	0	0	0	0
s_2	0	0	1	0	0	0
s_3	0	0	0	1	0	0
s_4	0	0	0	0	1	0
s_5	0	0	0	0	0	1

Figure 6. Students admitted in department matrix

```
        }
        else
           deg ad [s]=0    // students has not taken admission in any department
   }
```

Here, finally seats are allocated on purely merit bases according to student choices over departments.

3. Conclusion:

Several Scheduling algorithm applications have been described in this paper and also graphical representation using bipartite graph described in simple way. We hope this paper raises the awareness and adoption of Scheduling algorithm application amongst university admission problem.

4. References

[1] Petcharat Viriyakattiyaporn, "College Admission Mechanism: Student-optimality vs. College-optimality".
[2] Azar Abizada and Siwei Chen, "The college admission problem with entrance criterion".
[3] Thomas Niemann, "Sorting and Searching Algorithms: A Cookbook"
[4] Insup Lee, Department of Information Science, University of Pennsylvania, "Software Systems OS overview- CPU Scheduling", CIS 505, Spring 2007.
[5] Jimmy Salvatore, University of Chicago, "Bipartite Graphs and Problem Solving".
[6] Fone. D, Hollinghrust S, Temple M, Round A, Lester N, Weightman A: "Systematic review of the use and value of computer simulation modeling in population health and health care delivery." Journal of public health medicine, 25 (4). Pp. 325-335.
[7] Kirthi Ramamrithan and John A. stankovic: "Scheduling Algorithms and Operating Systems Support for Real-Time Systems". Proceedings of the IEEE: 1994.
[8] Gale, D. and L. Shapely (1962), "College admissions and the stability of marrige", American Mathematical Monthly, Vol.69, 9-15.
[9] Knuth D. E. (1976) Marriages stables. Les Presses de L'Universite Montreal, Montreal.
[10] Gustfield D, and Irving RW (1989), "The stable marriage problem: structure and algorithms." The MIT Press, Cambridge.
[11] Abdulkadioroglu A., and Sonmez T. (2003), "School choice: a mechanism design approach", American Economic Review, Vol. 93, 729-747.
[12] McVitie D. G. and Wilson L. B. (1970), "Stable marriage assignment for unequal sets"' BIT, Vol. 10, 295-209.