An MCMC based Course to Teaching Assistant Allocation

Saurabh Kumar*, Shana Moothedath†, Prasanna Chaporkar‡, Madhu N. Belur§

Department of Electrical Engineering
Indian Institute of Technology Bombay, India
Email: [*saurabhkm, †shana, ‡chaporkar, §belur]@iitb.ac.in

***Abstract*—Allotting Teaching Assistants (TAs) to courses is a common task at university centers which typically demands a good amount of human effort. We propose a method to allocate using computer algorithm. The presence of conflicting constraints, posed by requirements which determine tradeoff among them tend to make this problem difficult to solve. This is essentially a matching problem and in this paper has been modeled as a Markov Chain of various intermediate allotments. Later we perform simple Monte-Carlo simulations over a naive bucket filling allotment. This leads us to a globally optimal allotment with a promise of faster convergence.**

***Keywords:* Matching Problem, Allotment Problem, Markov Chain Monte-Carlo Simulations, Hungarian Algorithm, Maximum weight matching.**

I. Introduction

Allocation/matching theory, a name referring to several loosely related research areas concerning matching, allocation, and distribution of indivisible resources, such as jobs, school seats, houses and so on, lies at the intersection of game theory, social choice theory, and mechanism design. An allotment problem is where a set of participants are asked for preferences for a certain task that might have to be allotted to them considering each of their preferences and also other task specific requirements. Such problems can be modeled and solved using many 'matching' solvers like bucket filling, Hungarian method based matching algorithms and so on. The allocations are done based on the preferences specified by the participant parties, and the optimum allocation is the one that maximizes the global satisfaction of all the members.

Matchings are mainly of two types, one-sided matchings and two-sided matchings. One sided matching are those in which one side participants do not have a preference list. Examples of these are allocations of dormitory rooms, courses, summer houses and so on where the allotment is solely based on the preference list of the student's side. More precisely, such an allotment is always optimal for the chooser's side.

The second category of matching problems called as two-sided matchings are the ones where both the sets of participants give their preference orders for the other side. Examples for these kind of problems are stable marriage problem, room mate allotment problem and so on. In a stable marriage problem the participants are men and women. The allotment can be either men optimal or women optimal, based on the criteria specified. Two-sided matching model and the concept of "stable matchings" was introduced by Gale and Shapley [1] in the year 1962. They introduced an algorithm known as the deferred acceptance algorithm, using which they they proved that a stable matching always exists. More details of two-sided matching can be found in [2], [3]. However, the wide applications of this result was revealed through the work of Roth in the year 1984 [4].

This paper describes a two sided allotment problem, where one side consists of students and the other side consists of courses. To describe in detail, consider a scenario in which a set of courses are going to run at an institute and each of these courses require a set of teaching associates. Each of the course is asked for the number of TAs they would be requiring and students are asked for their choice preferences, as to which course they would like to be TA for. An 'optimum' allotment of students to courses is to be found.

Various algorithms have been developed in the past for solving such allotment problems. Automatic allotment of package tours is solved using Answer Set Programming [5]. A stochastic allocation problem is dealt with in [6] and solved in which a given quantity of resource is to be allocated to each activity. Taillard et. al. [7] try solving a quadratic assignment problem by using taboo search. Various genetic algorithm based methods also be found in the literature along with swarm optimization techniques which try to deal with this problem [8],[9]. In our problem, the Munkres algorithm can also be used to solve such problems as presented in [10].

A way to judge how good an allotment is by checking for its utility with respect to certain requirements. The utility can be measured in terms of:

- Student preference
- Course getting capable students
- Consistency of TA capability across courses.

The measure we use for course satisfaction is the grades that students allotted to a course had secured. The grades secured by the assisting students in that particular course is expected to have a direct impact on the performance of the whole course. Thus the average grade of the assisting students in that course is taken as a measure of the course satisfaction. A measure of student happiness is to see that each of the student ends up getting his highest possible choice. Clearly, a course that denied a particular student should not get allotted to another student of lower grade. A measure of consistency of TA quality of allotment is the variance of GPA of students across courses for an allotment. This ensures that the students are evenly distributed across courses. Here 'even' means that, no course ends up getting all TAs of high CGPA (Cumulative Grade Point Average) letting some other courses getting only low CGPA TAs. This is because, this allotment will result in serious problems in the overall performance of the courses.

An important point to note here is that not all the sub-objectives listed above are local. They are actually a global measure and is unlikely to be achievable by a local algorithm. Considering this as a bipartite graph matching problem, these constraints end up conflicting each other and prevents us from allotting weight for all the edges. In this paper we describe and compare three methods which can be used to solve this problem:

- Bucket Filling
- Hungarian Method
- Markov Chain Monte Carlo Simulations.

II. Few Solution Techniques

We detail few existing methods for solving allocation problems.

A. Bucket Filling

A naive way to solve this problem would be to arrange the courses in a certain order based on their popularity/importance or number of takers from past records. Also the students are arranged in the order of their GPA or any other criteria. After we have these two ordered lists we keep allotting the students to courses as they come by. For example first student on student list is allotted the first TA position in the course list, the second student allotted the second TA position and so on. To summarize,

- Order the list of students and the list of courses in a arbitrary manner.
- Irrespective of the preference ordering, allot each student to courses in a serial order.

Note that this activity requires significant amount of manual work to actually order the courses and students in the desired manner. And most importantly the work is not completely programmable. Also there is no guarantee that the allotment is optimal and best possible. Performing a simple test for its utility will usually give a bad result on an allotment obtained from this method. This method is likely to be biased against a certain set of courses which happen to come up in the list before actual bucket filling.

B. Maximum Weight Matching

Another way would be to model the problem by constructing a student vs courses array and performing a maximum weight matching over it. Each students choices are given a certain weight and the lowest weight is given to course he/she did not opt for. Then this maximum weight matching problem can be solved by using the classical Munkres algorithm. This method was proposed by J. Munkres in the paper [11], in which he proposes to solve the transportation problem using the Hungarian method.

This method is guaranteed to give a unique allotment and to converge. A downside to this approach though is that it is computationally intensive. A typical requirement in this solver is that the constraints should be 'local' in the sense that there are checks/values to be computed local to each vertex: like student preference number, course's requirement of relatively good students as TAs. However, often for organizational or administrative reasons, some constraints are not local, but what we term as 'global', for example, both the average and variance of the academic performance of students allotted to a course must be quite consistent across courses. Such a constraint cannot be computed locally at a vertex. This is because, the optimum matching corresponding to one requirement need not be the optimum with respect to the other requirement. This is because of the competing nature of the constraints and hence the Hungarian method for maximum matching cannot be used.

C. Markov Chain Monte Carlo Simulations

Markov Chain Monte Carlo (MCMC) is a widely accepted method for solving a wide variety of sampling problems in statistics, physics, compute science and econometrics. Sampling problems have many computational applications such as [12],

- Approximate counting: Given a problem instance, the objective is to obtain the size of the possible solutions.
- Statistical physics: The state space consist of all possible configurations of statistical mechanical system. The stationary probability has probability of each configuration related to its energy. The objective here is to sample configurations according to stationary distribution for examining the probabilities of a configuration.

The Markov Chain Monte Carlo method involves constructing a dynamic stochastic process (a finite Markov chain) whose states corresponds to the set of interest such as feasible solutions to a combinatorial optimization problem. The process can progress through the state space Ω by making local perturbations of the structures. Moreover, if the process is ergodic, that is, there exists a stationary distribution, then the chain can evolve in such a way that the final state of the chain asymptotically converges to a stationary distribution π over Ω, independent of the initial condition. In such a case by simulating the process for a sufficiently large number of steps we can reach the desired distribution π starting from any initial state. This sampling technique starts with some arbitrary initial state and guarantees the convergence to the desired distribution over some time.

MCMC techniques are widely applied for solving integration and optimization problems in large dimensional spaces. The idea of Monte Carlo simulation is to draw an i.i.d. set of samples $\{x^{(i)}\}_{i=1}^{N}$ from a target density $P(x)$ defined on a high-dimensional space $\mathcal{X}$, like the set of possible configurations of a system, the space on which the posterior is defined, or the combinatorial set of feasible solutions. These N samples can be now used to approximate the target density.

III. Problem Statement

We propose to model the problem at hand as a Markov chain with various intermediate allotment being the states of this Markov chain. The resulting allotment guarantees convergence to a unique stationary distribution. This procedure also has an advantage of converging much faster than the Hungarian method based approach with a lower memory requirement.

The algorithm consists of preparing the data in a form of student-vs-course matrix similar to the Hungarian method based approach. Simple row flips based Monte-Carlo simulations are then performed on this student-vs-course matrix to obtain subsequent allotments. These transition between the states of this Markov chain is governed by the change in the utility over allotments.

Defining the Utility of an Allotment

The utility of an allotment is modeled as a linear combination of 3 sub-objectives which being the Student happiness, Course satisfaction and a measure of the TA quality for an allotment. These three sub-objectives are briefly described below:

- Student Satisfaction (SS)
  This gives a measure of satisfaction of all students in a particular allotment. The choices of students to all the courses are captured as 1, 2 , and 3 for the first, second and third choices respectively. Thus happiness factor is defined as the sum of their choice numbers and lower the happiness factor, the allotment is more student favoring. Ideally we would want this is be as low as possible for an optimal allotment.

- Course Satisfaction (CS)
  This gives a measure of satisfaction of all the courses in a particular allotment. The courses would like to be allotted to a student who secured good grades in the past offering of the course. This can be modeled as the sum of the grades students secured in the course they have been allotted to in a particular allotment. If a student has not opted for the course, his grade of that course are taken to be 0.

- Capability of TAs across courses (CTA)
  A measure of how unbiased an allotment is towards the courses is necessary to make sure an even distribution of skilled TAs across all the courses. This measure can be modeled using the variance of the average GPA across the courses. This quantity kind of also regularizes the other two sub-objectives to maintain a balance between them. We would eventually want this quantity to be high so that each course has a balanced GPA of the TAs allotted to it.

These sub-objectives are now linearly combined to give an overall allotment value of an allotment. The linear combination enables a flexibility in the kind of allotment generated and hence provides us with a knob to tune which of the above three are to be given more/less importance while allotting. Thus at state X, the global utility function is then written as follows:

$$U(X) = K_1 \times SS_X + K_2 \times SA_X + K_3 \times CTA_X$$

where K_1, K_2 and K_3 are parameters which denote relative importance of each of the sub-objectives and can be chosen by the user as per their requirement. As per the definitions of sub-objectives and quantities they denote, SS needs to be minimized whereas CS and CTA need to be maximized to attain our optimum allotment. Hence, we choose K1 to be a negative constant along with both K2 and K3 being positive.

Let X_n be an allotment which is an $M \times N$ matrix, where M and N denote the number of students and to number of courses respectively. The parameter n here denote the iteration count. Let $U(X_n)$ denote the utility of the allotment X_n. The MCMC iterations can then be summarized as follows:

Algorithm 1 MCMC iterations

for $n \leftarrow 1$ to maxIterations **do**
    $X_{n+1} \leftarrow$ swapRows(X_n)
    if $U(X_{n+1}) \geqslant U(X_n)$ **then**
        $X_i \leftarrow X_{i+1}$
    else
        $\beta \leftarrow 10\log(n)$
        Choose $p \in (0,1)$ such that,
        $p \leftarrow \exp(\beta(U(X_{n+1}) - U(X_n)))$
        $X_i \leftarrow X_{i+1}$ with prob p.

The swapRows operations is performed by first generating two random numbers between 1 to M and then swapping the rows corresponding to them in an allotment X_n.

IV. Results

For simulation, we used real time data from a total of 59 courses and 308 students of Department of Electrical Engineering, Indian Institute of Technology Bombay. An Ubuntu based 3.2 Gz computer was used to implement this in Python. The simulation results are shown in Figures 1, 2 and 3.

Figure 1 shows the convergence plot for MCMC simulations over iterations. The initial allotment index is negative as the student satisfaction sub-objective is very high and we choose K_1 to be 1. The plot shows that by about 40,000 iterations the chain converges satisfactorily.

Figure 2 compares allotment by the bucket filling initialization with MCMC based allotment. In the bucket filling allotment, a lot of students end up getting their last choice which is denoted to be 10. After the MCMC simulations majority of students get their first choice. This can be controlled with parameter K_1 in the overall allotment value function.

Similarly in the Figure 3, with a simple bucket filling initialization, the average GPA across course peaks at 9 and is skewed. After the MCMC simulations the distribution of average GPA become better and peaks around 8.25. This can

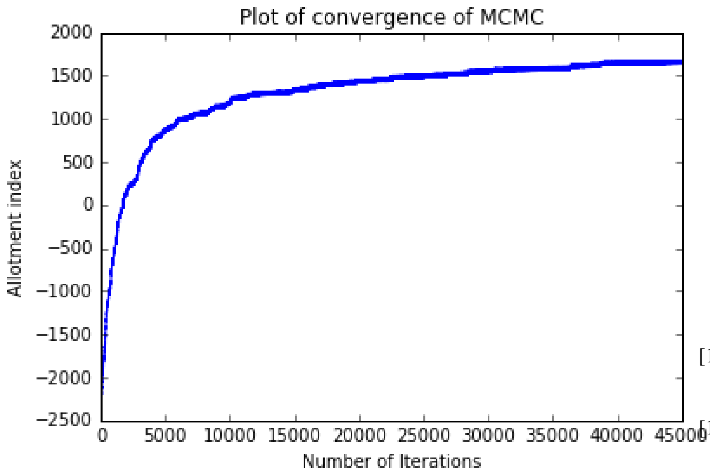

Figure 1: Plot of convergence for MCMC.

be controlled with parameter K_3 in the overall allotment value function.

V. Conclusions

Our work focused on the class of matching problems where the conventional polynomial time algorithms cannot be implemented, like the course to TA allotment as a representative example. The performance index of a matching is a combination of local and global constraints. A brute-force method to find the best matching is often not practical due to the exorbitant time required to find the best allocation for even a typical size of say 100 students.

To this end, we proposed a randomized algorithm for solving this using the Markov Chain Monte Carlo (MCMC) method. We constructed a Markov chain associated with this bipartite graph where every state is a perfect matching and the state space consists of all possible perfect matchings. We conducted simulations using the Markov chain and obtained affirmative results showing the convergence and the attainment of a near optimum within minutes. This method has been put to use now for allocation in our academic unit which comprises of about 300 TAs.

Acknowledgments

We would like to thank Mahak Arora and Kariyappa Singadi from PC Lab, Department of Electrical Engineering, IIT Bombay for providing us all the required data for this work.

References

[1] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962.

[2] L. S. Shapley and M. Shubik, "The assignment game I: The core," *International Journal of Game theory*, vol. 1, no. 1, pp. 111–130, 1971.

[3] A. S. Kelso Jr and V. P. Crawford, "Job matching, coalition formation, and gross substitutes," *Econometrica: Journal of the Econometric Society*, pp. 1483–1504, 1982.

[4] A. E. Roth, "The evolution of the labor market for medical interns and residents: a case study in game theory," *The Journal of Political Economy*, pp. 991–1016, 1984.

[5] C. Dodaro, N. Leone, B. Nardi, and F. Ricca, "Allotment problem in travel industry: A solution based on ASP," *Web Reasoning and Rule Systems*, pp. 77–92, 2015.

[6] H. Mendelson, J. S. Pliskin, and U. Yechiali, "A stochastic allocation problem," *Operations Research*, vol. 28, no. 3, part-II, pp. 687–693, 1980.

[7] É. Taillard, "Robust taboo search for the quadratic assignment problem," *Parallel Computing*, vol. 17, no. 4, pp. 443–455, 1991.

[8] P. C. Chu and J. E. Beasley, "A genetic algorithm for the generalised assignment problem," *Computers & Operations Research*, vol. 24, no. 1, pp. 17–23, 1997.

[9] A. Salman, I. Ahmad, and S. Al-Madani, "Particle swarm optimization for task assignment problem," *Microprocessors and Microsystems*, vol. 26, no. 8, pp. 363–371, 2002.

[10] F. Bourgeois and J.-C. Lassalle, "An extension of the Munkres algorithm for the assignment problem to rectangular matrices," *Communications of the ACM*, vol. 14, no. 12, pp. 802–804, 1971.

[11] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957. [Online]. Available: http://www.jstor.org/stable/2098689

[12] V. Guruswami, "Rapidly mixing Markov chains: A comparison of techniques," *Available: http://cs.ashington.edu/homes/venkat/pubs/papers.html*, 2000.

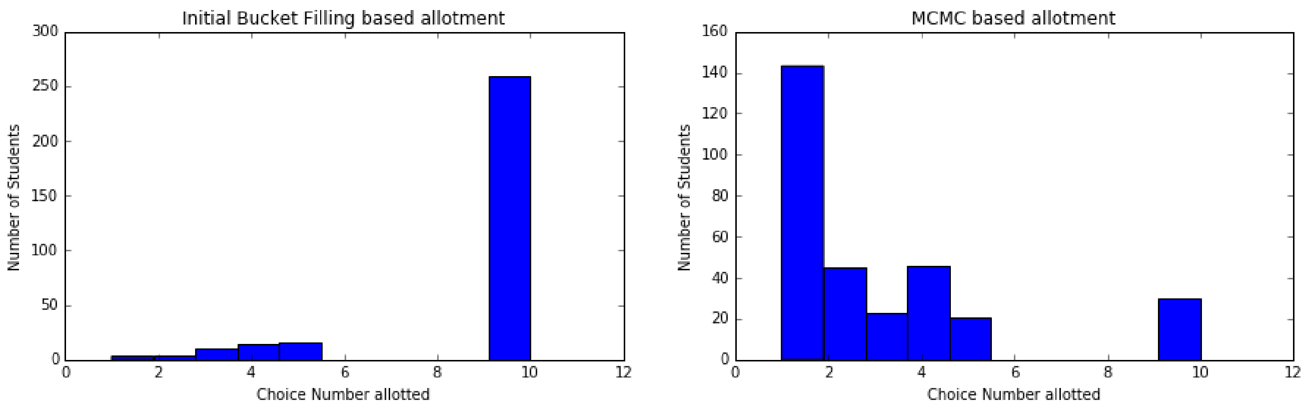

Figure 2: Plot of Choice Number vs number of students.

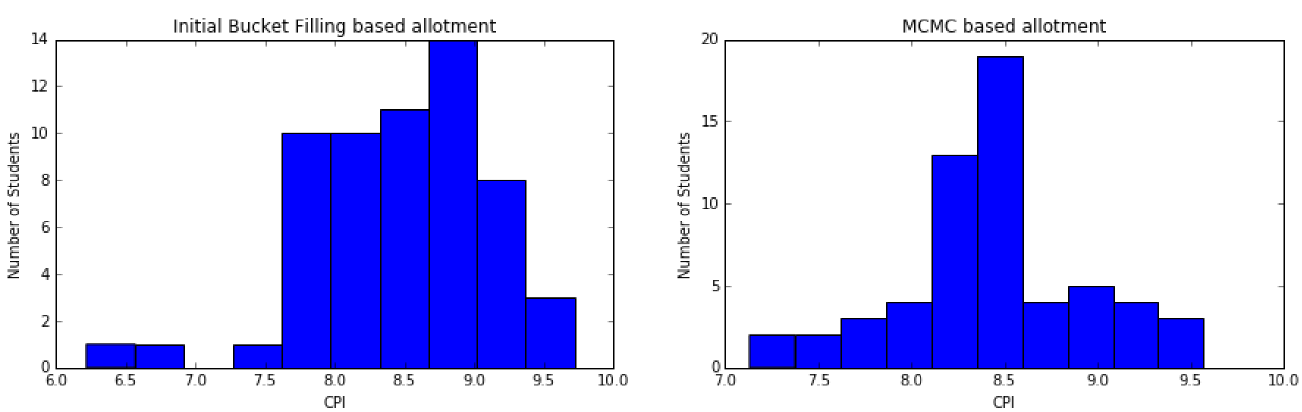

Figure 3: Plot of GPA vs number of students.