Scheduling Job Placement Interviews at a University

Linus Schrage

Lindo Systems
http://www.lindo.com
and
University of Chicago

Abstract. We describe experience with a job interview scheduling system that has been in use since 2001 at the University of Chicago Graduate School of Business(UCGSB). During interview season, a typical week has 300 job-seekers requesting 1000 interviews from 100 interviewing companies, to be scheduled into 1200 timeslots. We match job-seekers with companies with time slots so that we maximize the number of desired interviews scheduled, no job-seeker has two interviews scheduled at the same time, no interviewer has two interviews scheduled for the same time, and no interview is scheduled for a job-seeker during an interval which he has designated as unavailable. One complication is that each company specifies(in advance) its interview length, ranging from 30 to 120 minutes. For example, a 30 minute interview slot with Company X may conflict with two 45 minute slots with Company Y that overlap with it. The interview scheduling system is part of a larger job interview bidding system. We examine some of the complications of directly combining the interview assignment part with the auction part.

1 Introduction

The job placement office is a very important component of any major business school. Perhaps the most visible activity of a placement office is to match job-seekers with on-campus interviewers from organizations that are contemplating hiring some of these job-seekers. Especially in difficult economic times, it is important to do an efficient job of matching job-seekers with appropriate interviewers. During interview season, a typical interviewing company will send 2 or 3 interviewers to campus for 1 or 2 days of interviewing. In advance, each company will specify whether it wants to have 30 minute, 45 minute, or 60 minute interviews. There may also be short rest gaps scheduled between interviews.

At UCGSB, this interview matching is a weekly process involving two steps. Well in advance of the week in question, once it is known which companies will be interviewing during the week in question, and on which days, and how many interview slots each company will have available, then step 1 is to hold an auction in which job-seekers bid on their favorite companies. The scarce time slots are then assigned to bidders in essentially a highest-bid-wins fashion. The second step in this weekly process is to assign successful bids and their associated job-seekers and companies to specific time slots. This is our main concern in the sections 2 and 3.

2 The Problem of Assigning Interviews to Time Slots

The output of step 1 is: a) a list of timeslots for each company specifying when each company can hold interviews, b) a list of (job-seeker, company) pairs of interviews won by jobseekers, and c) a list of "black out" intervals for each jobseeker specifying when each jobseeker cannot hold interviews, e.g., because of attendance in a course. The step 2 assignment problem is then defined algebraically as follows.

x_{skt} = 1 if job-seeker s is assigned to an interview with company k in its t^{th} time slot, else 0. This variable is defined only for job-seeker/company/time slot combinations for which a) the job-seeker has won an interview with the company in step 1, b) the company is holding interviews in that time slot, and c) the job-seeker is willing to have an interview in that time slot, i.e, the time slot does not overlap one of his black out intervals;

n_k = number of interviewers that company k has available in each of its time slots.

Maximize value of interviews scheduled, i.e.,

Maximize $\Sigma_s\Sigma_k\Sigma_t\, x_{skt}$;

subject to:

For each company k and timeslot t:

Number of interviews scheduled ≤ interviewers available, i.e.,

$\Sigma_s\, x_{skt} \le n_k$;

For each job-seeker s and company k:

Number interviews scheduled ≤ 1, i.e.,

$\Sigma_t\, x_{skt} \le 1$;

For each job-seeker s, any two overlapping time slots t and v of companies k and p:

Number interviews scheduled ≤ 1, i.e.,

$x_{skt} + x_{spv} \le 1$;

For all s,k, and t,

x_{skt} = 0 or 1;

3. Solving the Model

If the time slot overlap constraints were not present, then the above model would be a transportation linear program and it would have naturally integer solutions when solved as a continuous linear program. The time slot conflict constraints, however, tend to result in fractional solutions if the model is solved as a linear program. The simplest illustration of this is if one job-seeker is trying to get an interview with each of three companies, and the only time slots available are overlapping, e.g., Company A : 10:30 to 11:15, Company B: 10:45 to 11:15, and Company C: 11:00 to 11:30. Loosely speaking algebraically, we have the three constraints:

$$
\begin{aligned}
A + B \quad\quad\; &\le 1; \\
A \quad\quad + C &\le 1; \\
B + C &\le 1;
\end{aligned}
$$

If solved as an LP, we will get A = B = C = 0.5, for a total of 1.5 interviews. It is clear, however, that only one interview can be assigned out of the three.

The above three interviews that each pairwise conflict with each other, constitute a "clique" in integer programming terminology. Most commercial integer programming solvers, LINDO in our case, detect such cliques and replace the above three constraints by the one constraint:

$$A + B + C \le 1;$$

If the resulting model is then solved, most of the variables will be 0 or 1, however, integer programming methods must be used to take care of the fractional variables. Below we tabulate our computational experience with two recent, slightly larger than typical, weeks.

Week A:	Week B	
265	448	bidders(jobseekers desiring interview slots),
129	135	companies interviewing during the week,
1056	1751	objects available(interview slots),
936	1660	objects requested(interviews desired by job-seekers)
12,321	42,403	constraints in the original integer program,,
10,051	41,384	variables in the original integer program,,
2,853	6,538	rows after reduction, including clique,
7,063	30,966	variables after reduction,
820.5	1580.166	interviews granted in continuous LP relaxation.
817	1579	interviews granted at integer optimum.
28	83	seconds to solve to optimality.

As the two example cases illustrate, the model has not been difficult to solve in practice. Several seconds is probably typical.

4. Experience and Extensions

Prior to the use of the above model, a heuristic was used for doing the assignment. We do not have any data on side-by-side comparisons of the heuristic and the optimal approach, however, users seem to be quite happy with the results. We do have some data from a related application to get some feel for the amount of improvement obtained by replacing a heuristic by an optimum procedure based on integer programming(IP). Graves, Sankaran, and Schrage(1993) describe a heuristic for doing course registration at UCGSB. The results for data from the Spring Term of 2000, comparing the heuristic with an optimal solution, are shown below:

Solution Comparison of IP with Old Heuristic Method:

350 object types(course sections), 2,091 bidders,
84,176 bids(variables), 2441 constraints.

	IP	Heuristic	% Improvement
Total value awarded	25,507,457	23,555,500	8.3
Successful bidders	2062	1949	5.8
Solution time(secs.)	48	negligible	

The "value awarded" is simply the sum of the winning bid values in the auction. "Successful bidders" is the number of students who were awarded a course schedule of some sort from among those that the job-seeker requested. Thus, an improvement of 5% is not an unreasonable expectation.

4.1 Extensions to Placement Assignment

There are several extensions we are considering: a) Use a more refined objective or measure of goodness, b) allow an interview to overlap a job-seeker specified blackout interval at a penalty, c) integrate the interview bidding/auction process with the follow-on interview assignment process, d) allow bidders to specify a budget constraint when submitting two or more bids.

With regard to (a), if we simply maximize the number of successful assignments, there may be alternate optima. Some of these alternate optima might be preferred to others. One plausible refinement is to apply the bid amount from step 1 of the process to each x_{skt} and maximize the value of assignments as measured by these bid amounts. A similar approach could be used for (b). If a job-seeker prefers to not have his interview overlap with one of his blackout intervals, but is willing to tolerate an overlap to get the interview, then we could generate a variable x_{skt} that overlaps a blackout interval, but we would reduce the value applied to it in the objective by a bidder specified penalty. Somewhat related to (a) and (b), a job-seeker might tolerate but not prefer to have "back to back" interviews. To represent this, a penalty variable might be introduced for any two such intervals to discourage such assignments. Other considerations might be that we would prefer a solution where each of two job-seekers get one interview each to a solution where one of the job-seekers gets two interviews but the other gets none. Again, this is easily represented, at the expense of adding additional variables and constraints to the formulation.

With regard to (c), we can notice that about 10% of the interviews that are sold in the step 1 auction, in fact do not get scheduled in step 2 because of time conflicts. The obvious idea would be to integrate steps 1 and 2. This is essentially idea (a) revisited. If we used the assignment model of step 2 immediately in step 1, with the job-seekers bid values in the objective, we would clearly improve the total value of the interviews awarded. The big challenge, however, would be to determine an appropriate clearing price for each company, because in this case it could be that a lower bid by one job-seeker would be successful while a higher bid of another job-seeker fails because of scheduling conflicts encountered by the second job-seeker.

With regard to (d), in the current system each job-seeker is given 1000 points at the beginning of the season. Each week a job-seeker has to pay points to the extent that she wins interviews that are successfully scheduled. A crucial feature of the system is that Vickrey prices are used, that is, a clearing price is calculated for each company, and each successful bid pays the clearing price. Currently, the system does not allow a bidder to submit a set of bids totaling more than his “point” wealth. For example, suppose that the clearing prices for companies A and B turn out to be 700 and 100 and a certain job-seeker is willing to pay up to 800 for either, although he could not pay more than 1000 in total. Currently he could not submit the two bids 800 for A and 800 for B, even though in retrospect he could have bought both. We have been able to formulate a model that allows clearing price based budget constraints, and display the results for four different weeks below.

				Value delivered by budget type		
Week	Bidders	Bids	Companies	No Budget	Price based	Bid based
1	193	725	40	61596	61393	45717
2	283	1036	39	86228	85317	72242
3	293	1078	57	91696	90794	77575
4	197	382	11	28014	27919	27790

Clearly, the last two columns suggest that there may be a substantial improvement in value delivered by allowing the more flexible budget constraint.

References

1. De Werra, D.: “An Introduction to Timetabling”, *European Journal of Operations Research*, Vol. 19, (1985) 151-162

2. Graves, R., Sankaran, J., Schrage, L.: “An Auction Method for Course Registration", *Interfaces*, Vol. 23, no. 5, (1993) 81-92.