TA Scheduling

Senior Design 1998

Drew Hall
Michael Chiang
Emily Krannich
Sean Vanlandingham
Table of Contents

Management Summary ........... 2

Analysis of the Situation ...... 3

Technical Description of GAMS Model .... 4

Technical Description of Visual Basic Program .... 8

Analysis of Results .......... 9

Conclusions and Critique .... 10
Management Summary

Problem

Each semester, the CSE department at SMU faces the problem of matching TAs with scheduled lab times. Currently, Mary Alys Lillard and Beth Minton spend over 80 man-hours per semester manually matching TAs with over 60 lab sections. The CSE department schedules labs, oftentimes occurring simultaneously, every weekday between the hours of 8:00 AM and 7:00 PM. The process of assigning TAs to lab sections is bounded by several constraints including schedule agreement, skill level, prior experience, maximum and minimum number of labs to teach, classification, and language ability.

Approach

A network optimization model best describes the TA scheduling problem. By using GAMS optimization software we will both create a TA network and solve for an optimal solution. Through a Visual Basic user input program TA schedules, lab times, and other criteria will be translated into GAMS code. After solving the network problem, the GAMS solution output will be interpreted into a user-friendly format by the Visual Basic software.

Deliverables

Upon completion, our team will provide the following to the CSE department:

- A Visual Basic user input program (installable on any Windows 95 computer)
- An instruction manual
- Adequate training for Mary Alys Lillard and Beth Minton
Analysis of the Situation

Network Model

The TA scheduling problem can best be solved using a network optimization model with mixed integer programming (MIP). We found a network model appropriate because it not only solved the assignment problem, but it also accounted for the additional problem constraints.

First, the model solves the assignment problem by creating arcs between TAs and lab sections, accounting for skill level and availability. Second, through the use of upper and lower bounds, the model forces all labs to be filled and limits the number of labs a TA can teach. Additionally, costs are assigned on arcs to account for language ability and experience.

The following model is a simplified example of the actual problem:

![Network Model Diagram]

Note: bounds and costs exist for all arcs, but are only shown for TA1 & 1305-01.

In this model, the yellow nodes represent individual TAs, and the green nodes represent the individual lab sections. The arcs indicate skill and availability. For instance, TA1 is skilled to teach 1305 and is available for sections 01 and 02. Therefore, arcs connect the TA1 node to both the 1305-01 and 1305-02 nodes. The bounds on the arc connecting SOURCE to TA1 indicate that TA1 can teach at most two labs, but has no minimum lab requirement. The bounds on the arc connecting TA1 to 1305-01 indicate that TA1 is skilled and available to teach 1305-01. The cost of +25 on this arc takes into account TA1’s lack of experience and moderate language proficiency (specific constraints explained further in Technical Description of GAMS Model). The bounds on the arc connecting 1305-01 to SINK indicate that the lab must be filled by a TA.
Technical Description of GAMS Model

The following is the GAMS code for the above network model. Comments highlighted in red precede each section, describing individual components of the program. This simplified model uses realistic hypothetical values; however, the full version uses actual data provided by Mary Alys Lillard.

* TA Scheduling Team
* Senior Design 1998

$OFFLISTER OFFSYMLIST OFFSYMREF OFFUEL LIST
OPTION limrowo, limcol=0, solprint=OFF, sysout=OFF;

* Declaration of all TA and lab section nodes along with their attributes.

SETS  N      List of network nodes
     /SOURCE Super source
     TA1
     TA2
     TA3
     1305-01
     1305-02
     1341-01
     1341-02
     2341-01
     2341-02
     1305
     1341
     2341
     SINK Sink node
     /
     A      Arc attributes
     /
     COST   Unit cost of flow
     UPPER  Upper bound on flow
     LOWER  Lower bound on flow
     /

ALIAS (N,N2);

* Scalar value included in flow balance equation and objective function will allow for a partial solution of infeasible problems

SCALAR BIGPEN Large penalty for violating a constraint /10000/;

* Declaration of constraints for all possible arcs between TAs and lab sections. (i.e. if a TA is available and skilled for the lab section, then an arc will be assigned between them)

* The UPPER bound allows the lab to be filled by a particular TA only once.
* Blank values for LOWER are assumed to be zero.
* COSTS for each TA are assigned as follows:
* +10 for no experience in teaching course
* +15 for moderate English ability (for 1305 only)
* +30 for nominal English ability (for 1305 only)

TABLE ARC(N,N2,A) Characteristics for individual network arcs with nonzero UPPER

<table>
<thead>
<tr>
<th>Arc</th>
<th>COST</th>
<th>UPPER</th>
<th>LOWER</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA1.1305-01</td>
<td>25</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>TA1.1305-02</td>
<td>25</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>TA2.1341-01</td>
<td>10</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>TA2.2341-01</td>
<td>10</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>TA2.2341-02</td>
<td>30</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>TA3.1305-01</td>
<td>30</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>TA3.1305-02</td>
<td>30</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>TA3.1341-01</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TA3.1341-02</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TA3.2341-02</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* UPPER and LOWER bounds for supply set the maximum and minimum number of labs
* a TA is allowed teach.
* UPPER bound of 2 indicates an undergraduate student.
* UPPER bound of 4 indicates a graduate student.

TABLE SUPPLY(N,A) Supply available at node N (max & min used is UPPER & LOWER)

<table>
<thead>
<tr>
<th>Node</th>
<th>UPPER</th>
<th>LOWER</th>
<th>COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA1</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TA2</td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>TA3</td>
<td>4</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

* UPPER and LOWER bounds for demand are set at 1 to force each lab
* section to be filled.

TABLE DEMAND(N,A) Amount of demand required at node N (max & min absorbed is UPPER & LOWER)

<table>
<thead>
<tr>
<th>Node</th>
<th>UPPER</th>
<th>LOWER</th>
<th>COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1305-01</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1305-02</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1341-01</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1341-02</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2341-01</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2341-02</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

POSITIVE VARIABLES
FLOW(N,N2) Flow on arcs
SOURCE(N) Supply created at source node N
SINK(N) Demand absorbed at source node N
POSDEV(N) Node infeasibility in positive direction
NEGDEV(N) Node infeasibility in negative direction

FREE VARIABLE
TOTCOST  Total cost value to minimize;

BINARY VARIABLE
X(N,N2)  Binary variable for course constraint

EQUATIONS
FLOWBAL(N)  Flow balance equation for each node
OBJ  Objective function to minimize
TIMETA1  Time constraint for TA1
TIMETA3  Time constraint for TA3
TA31305  Course constraint for TA3
TA31341  Course constraint for TA3
TA32341  Course constraint for TA3
TA3COURSE  Course constraint for TA3

* NON-NETWORK CONSTRAINTS
* Additional constraints not represented by standard network model

* 1305-01 & 1305-02 are taught at the same time.
* Equations limit TA1 and TA3 to teaching one or the other, but not both.

TIMETA1.. (FLOW("TA1","1305-01") + FLOW("TA1","1305-02")) =L= 1;

TIMETA3.. (FLOW("TA3","1305-01") + FLOW("TA3","1305-02")) =L= 1;

* The following four equations create a constraint limiting TA3 to
* teaching two or fewer courses. (i.e. labs from only 1305 & 1341)

* The following three equations each use a binary variable to limit flow on
* all arcs for a given course. For example, if the binary variable equals one,
* then flow can exist on any arc for that particular course. However, if the binary
* variable equals zero, then flow cannot exist on any arcs for that particular course.
* Since TAs are limited to teaching a maximum of four courses, the maximum value
* for any right hand side is four. Therefore, the number four is chosen as a multiplier
* for the binary variables.

TA31305.. FLOW("TA3","1305-01")+FLOW("TA3","1305-02") =L= 4*X("TA3","1305")

TA31341.. FLOW("TA3","1341-01")+FLOW("TA3","1341-02") =L= 4*X("TA3","1341")

TA32341.. FLOW("TA3","2341-02") =L= 4*X("TA3","2341")

* The fourth equation sets the sum of the three binary variables to
* less than or equal to two, thereby limiting the TA to teaching only
* two out of three courses.

TA3COURSE.. X("TA3","1305")+X("TA3","1341")+X("TA3","2341") =L= 2;

* The flow balance equation sets the flow coming out of the SOURCE node equal to the flow
* going into the SINK node.
FLOWBAL(N).. SUM(N2 \$ ARC(N2,N,"UPPER"),FLOW(N2,N)) + (SOURCE(N) \$ SUPPLY(N,"UPPER")) + POSDEV(N) =E= SUM(N2 \$ ARC(N,N2,"UPPER"),FLOW(N,N2)) + (SINK(N) \$ DEMAND(N,"UPPER")) + NEGDEV(N);

* Objective equation which minimizes total cost.

OBJ.. SUM((N,N2) \$ ARC(N,N2,"UPPER"), ARC(N,N2,"COST")*FLOW(N,N2)) + SUM(N, SOURCE(N)*SUPPLY(N,"COST") + SINK(N)*DEMAND(N,"COST") + SUM(N,BIGPEN*(POSDEV(N)+NEGDEV(N))) =E= TOTCOST;FLOW.UP(N,N2) = ARC(N,N2,"UPPER")..

FLOW.LO(N,N2) = ARC(N,N2,"LOWER");
SOURCE.UP(N) = SUPPLY(N,"UPPER");
SOURCE.LO(N) = SUPPLY(N,"LOWER");
SINK.UP(N) = DEMAND(N,"UPPER");
SINK.LO(N) = DEMAND(N,"LOWER");

* Solves network using mixed integer programming (MIP).

MODEL NW /ALL!;
SOLVE NW USING MIP MINIMIZING TOTCOST;
DISPLAY TOTCOST.L, FLOW.L, X.L, POSDEV.L, NEGDEV.L

* Prints output file.

file schedule /solution.dat!;
put schedule;
loop((N,N2) \$ FLOW.I(N,N2), put N.l, @12, N2.l, @24 /);
Technical Description of Visual Basic Program

Why Visual Basic?

GAMS optimization software is highly effective at solving specific models; however, it requires a significant amount of time and a skilled GAMS programmer. Since the CSE department needs to repeat the process of selecting TAs each semester, a more general and user friendly model is needed. A Visual Basic user input program eliminates the need for a GAMS programmer by translating data into GAMS code.

Description of Process

The user begins by inputting TA schedules, lab times, and other criteria, which are stored in a Microsoft Access database. The user can then utilize the program to create a GAMS text file from the given database. This text file can be transferred to the UNIX platform (titan.cis.smu.edu) using an FTP program. Once in UNIX, GAMS is used to solve the network problem. The output from the GAMS solution can then be transferred back to the user's PC using FTP. Finally, the program will translate the output file into a readable format specified by the CSE department.

Note: A more complete description of the Visual Basic program can be found in the user manual.
Analysis of Results

The original output from GAMS is both lengthy and difficult to understand. A short function (put file) added to the end of the program bypasses this problem by creating a new data file in a simple format.

The function outputs the following solution to the simplified network model:

<table>
<thead>
<tr>
<th>TA</th>
<th>Lab</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA1</td>
<td>1305-02</td>
</tr>
<tr>
<td>TA2</td>
<td>1341-01</td>
</tr>
<tr>
<td>TA2</td>
<td>2341-01</td>
</tr>
<tr>
<td>TA2</td>
<td>2341-02</td>
</tr>
<tr>
<td>TA3</td>
<td>1305-01</td>
</tr>
<tr>
<td>TA3</td>
<td>1341-02</td>
</tr>
</tbody>
</table>

The solution adheres to all constraints, producing a list assigning TAs to every lab. Since the model minimizes cost, we can assume that this is the optimal solution.
Conclusions and Critique

Conclusions

Through the successful implementation of the Visual Basic program and GAMS model, the CSE department will eliminate the need for excessive man-hours of planning at the beginning of each semester. The process of selecting TAs, which previously required 80 hours of work, now requires only 2-4 hours to complete. In addition to saving time for the faculty, students will also benefit from the program. Since GAMS minimizes cost (lack of experience and language barriers), students will receive instruction from the best possible TA.

Critique

The Visual Basic program easily automates the creation of a text file for GAMS, but it is limited to satisfying only the specified constraints. Adding additional constraints (e.g. only two labs per day) would require a proficient GAMS and Visual Basic programmer. Individual exceptions to the constraints will always arise in a given situation, but a model cannot possibly be built to account for these special cases. The user can alleviate this problem by manually manipulating the GAMS text file.

A final critique involves the issue of technical support for the product. Because our group is comprised of graduating seniors, the CSE department will be solely dependent on the user manual for support.