ETM
ENGINEERING & TECHNOLOGY MANAGEMENT

Optimization of student utilization in PSU library

Course Title	**:**	**Operations Research**
Course Number	**:**	**EMGT 540**
Instructor	**:**	**Dr. Timothy Anderson**
School	**:**	**Engineering and Technology Management, Portland State University**
Term	**:**	**Fall**
Year	**:**	**2011**
Prepared by	**:**	**Opinderjit Bhella Priti Maheshwari Neil Runde Aarthi Neethirajan Duygu Ucar**

Executive Summary

The labor staffing and scheduling process in many organizations is vital for successful operations. For companies where the work is done on a shift basis, the scheduling process can become cumbersome, inefficient and challenging. Intelligently dealing with this complexity can be a competitive advantage.

In this paper, we address the problems faced in the scheduling process of the circulation desk at the Portland State University library. The circulation desk has a varying volume of customer requests depending on what day it is and the time. A fulltime manager manually attempts to appropriately staff students based on the transaction volume, student availability and hours requested by the student. This paper proposes a method to optimize the scheduling of student staff whenever the library is open according these criteria.

Our study aims to address the following overarching goals:

1. Understand and describe the current methodology used by library officials to schedule the student staff
2. Apply the principles and efficiencies of Operation Research techniques
3. Recommend an improved approach towards the existing approach of scheduling
4. Contribute to the existing method to optimize the utilization of student staff

Table of Contents

Literature Research

Operations Research has no shortage of research papers discussing interesting staffing formulations. However, each solution has unique characteristics and not universally applied, so finding relevant puzzle pieces is necessary. For example, Gordon and Erkut developed a spreadsheet based decision support tool that generated shift times, scheduled volunteers according to various constraints and preferences. They produced master and individual schedules [1]. Their model was to improve volunteer scheduling for the Edmonton folk festival. Ashley wrote a white paper on optimization of staff scheduling at the library circulation desk of University of Kansas [2]. He modeled his scheduling problem as a binary integer linear formulation. Ashley's formulations were similar to our circulation desk problem; however, he did not generate dynamic shifts based on demand. Our circulation desk required many 2, 4, or 6 hour shift permutations. Gendron formulated an integer programming model for scheduling employees for a large warehouse distributor [3].

Our paper expanded on the formulations as presented in these papers by creating dynamic shifts and adding preference goals.

Portland State University

The Portland State University Library (Miller Library) is the heart of the Portland State student community. It provides resources for research, teaching, and learning support in a rapidly changing information environment. Along with its significant collection of information resources, the library delivers a superlative reference and instruction program dedicated to improving students' academic success. It also offers outstanding special collections and archives featuring unique materials of regional and scholarly interest. Finally, it provides an extensive array of user-centered information services.

The circulation desk is a primary service point in the Portland State Library. The desk is staffed when the Miller Library is open and provides a variety of services described below:

1. Checking out library materials
2. Placing a recall
3. Obtaining Portland State identification cards
4. Reporting library issues
5. Returning library materials
6. Placing a search for materials or resources
7. Library special-use rooms
8. Lost & found

The aim of this paper is to use Operations Research and similar methodologies to schedule staff at the circulation desk to meet the demand of services stated above.

Data Gathering

Preliminary data and problem definition was collected by several meetings with the library sponsors. The authors met with Michael Bowman and Bronwyn Dorhofe (who reports to Michael and is the user services coordinator). Through these meetings, we were able to obtain several documents related to:

1. Students working schedules at the circulation desk in August' 11, September'11
2. Daily exit turnstile count from October 1st - 31st, 2011
3. Shifts and hours worked by employee from August 1st, 2011 and November 1st, 2011
4. Student Training Handbook
5. Students expectations
6. Student circulation job description
7. Library circulation student peer supervisor job description
8. Library Student Employee Handbook
9. Circulation statics from August 1st, 2011-October 31st, 2011

These documents helped the authors establish many of the modeling constraints.

Current Approach

Currently, Bronwyn (the user services coordinator and our sponsor) manages the weekly schedule on an Excel spreadsheet. She determines the amount of students she needs to staff and manually allocates the shifts while keeping in mind each person's preference for blocked time, requested hours and the maximum hours per person per week. Bronwyn then uses the '*WhenToWork'* online scheduling software and manually publishes the schedule on a weekly basis where the students can view it using an Internet browser. An example of a published schedule is shown in *Appendix A*.

Problem Statement

The Miller Library is open 192 hours per week with different peak times through out the day on different days. Data was collected from the door rotation and book checkout to determine these peak times. For example, the busiest time at the circulation desk is from 11:00 am–4:00 pm and 5:00 pm–5:30 pm. The library currently staffs 10 students to work at the circulation desk and priorities the schedule around the student's preferences and the library's demand.

This creates a unique problem because the schedule is dynamic and student preferences are fluid. In addition to the class schedules, the maximum amount of work hours allowed is a different constraint for international students than U.S. citizens. For example, the international students can work a maximum 20 hours per week whereas the U.S. citizens can work a maximum of 40 hours per week. During our interview, it was considered acceptable to overstaff the students. This is because, when the circulation desk is not busy, some of these students can attend to other duties, such as checking books or re-shelving. Finally, a minimum of 1 and up to 3 students can be staffed at the circulation desk.

After speaking with Bronwyn, she stated the following objects:

1. Primary Objective: maximize target cover based on library traffic patterns (or demand)
2. Secondary Goal: maximize the student's desired hours to work per week

From the documentation and interviews, our starting assumptions are as follows:

1. Students can only work 1 shift a day
2. A shift can be 2 hour, 4 hour or 6 hour
3. Students can work no more than the maximum hours as set by the coordinator
4. Students should be able to block out time periods (periods they can't work due to class etc.)

We modeled object 1 as the primary object and object 2 as a secondary goal. We decided to create our user interface in Excel and model the formulations in 'GPLK' and 'Python'.

Model 1 – Create weekly schedules based on library demand

The first model determines the daily shift schedules based on library demand for a time period. The library coordinator opted for a 30 minute period; however, our model supports any time period increments that are less than 2 hours.

Decision Variables

x_j = the number of students assigned to shift j. Shifts have a start time period and can be a duration of 2, 4 or 6 hours.

Parameters

D_i = the number of students needed in time period i. (e.g. D_1 represents period from 10:00 to 10:30 with a need of 3 students)

C_{ij} = 1 if time period i is covered by shift j, 0 otherwise

Sets

S_4 = a subset of 4 hours shifts

S_6 = a subset of 6 hours shifts

Model

Minimize the surplus of students

$$\text{minimize} \quad \sum_j x_j$$

s.t.

1. This constraint determines all the feasible 2, 4 or 6 hour shifts that can fill the library's demand for period i.

$$\sum_{j} C_{ij}\ x_j\ \geq\ D_i\ \ \forall i$$

2. This constraint restricts the model from assigning too many students to 4 and 6 hour shifts. We started with these initial restrictions; however, these RHS values need to be fine tuned to meet the library's and student's preferences.

$$\sum_{j\ \in s_4} x_j\ \leq 4$$

$$\sum_{j\ \in s_6} x_j\ \leq 2$$

$$x_j\ \geq\ 0\ and\ integer\quad \forall j$$

This model was executed as 7 separate linear programs with the demand period data for each day. This resulted in 75 integer variables for each day, with a combined total of 525 variables.

Model 2 – Assign students to the shift schedules from model 1

The second model assigns students to the subset of shifts and staff demand as determined from the first model.

Decision Variables

x_{ijk} = 1 if student i is assigned on day j to shift k, 0 otherwise

y_{ijk} = 1 if student i is working on day j to shift k, 0 otherwise

q = mini-max for secondary goal, i.e. student hours requested

Parameters

D_{jk} = number of students needed to work on day j in shift k; this table is determined from the first model

A_{ijk} = 1 if student i is available to work day j in shift k; this table is created from the data available from the spreadsheet and the first model

P_i = hours requested by student i

T_k = the duration hours for shift k

W = the priority weight for the mini-max decision variable q

Model

Maximize the coverage and student requested hours

$$maximize \sum_{\substack{i \in Person \\ j \in Day \\ k \in Shift}} x_{ijk} - q$$

$s.t.$

1. This constraint determines if the library coverage requirements are met.

$$\sum_{i \in Person} x_{ijk} \geq D_{jk} \quad \forall jk$$

2. Linking constraint, 1 if student i is working, 0 o.w.

$$x_{ijk} - My_{ijk} \leq 0 \quad \forall jk$$

3. This constraint only assigns a shift if student is available to work.

$$y_{ijk} = A_{ijk} \quad \forall jk$$

4. This constraint assign no more than one shift a day for each person.

$$\sum_{k \in Shift} x_{ijk} \leq 1 \quad \forall ij$$

5. This constraint is a secondary goal to maximize the student's weekly requested hours.

$$\frac{\left(\sum_{jk} T_k \, x_{ijk}\right) - P_i}{P_i} W \leq q \quad \forall i$$

$$x_{ijk} \geq 0 \text{ and integer} \quad \forall i,j,k$$
$$y_{ijk} = \{1,0\} \text{ and binary} \quad \forall i,j,k$$
$$q \geq 0$$

This model is executed once per week. This resulted in 8064 binary variables.

Front-end UI

Our development environment consisted of a front-end UI and several back-end services. The application framework and architecture is available in *Appendix B*. The front-end UI was developed in Excel to match the coordinator's current work flow. A schedule can be generated by following the 4 steps below.

Steps to generate a schedule

Step 1: As shown in Figure 6.1, the library coordinator will need to add the student's name in "stdnt name" column, maximum number of hours the student can work in a week in "wkly mx hrs" and the hours a student would like to work in "req'd hrs column." The "actual" column will automatically be updated to contain the total hours the student has worked for the week. In Figure 6.2, the library's global policy will need to be entered. For example, the maximum number of hours that a student can work and the maximum and minimum shift sizes.

(1) emp prefs			
stdnt name	wkly mx hrs	req'd hrs	actual
Tommy	20	20	40
Sally	20	20	40
James	20	20	96
Opinder	20	20	68
Neil	20	20	97
Robert the 3rd	20	20	84
Basil	20	20	84
Christy	20	20	40
Hannah	20	20	104
Michelle	20	15	56
Joan	20	0	20
Lisa	20	0	20

Fig: 6.1

global policy	
max req hrs per wk	20
max shift size	6
min shift size	2

Fig: 6.2

Step 2: In step 2, the library coordinator will need to enter the demand for the number of students required for a particular time. This heat map is color coded and automatically changes with demand value. For example, green means no demand and red means high demand. See Figure 6.3.

(2) library demand

schedules	8:00	8:30	9:00	9:30	10:00	10:30	11:00	11:30	12:00	12:30	13:00	13:30	14:00	14:30	15:00
sunday	0	0	0	0	0	0	0	0	0	3	3	3	3	3	3
monday	1	1	1	2	2	3	3	3	3	3	3	3	3	3	3
tuesday	1	1	1	1	2	3	3	3	3	3	3	3	3	3	3
wednesday	1	1	1	1	2	3	3	3	3	3	3	3	3	3	3
thursday	1	1	1	1	2	3	3	3	3	3	3	3	3	3	3
friday	1	1	1	1	2	3	3	3	3	3	3	3	3	3	3
saturday	0	0	0	0	0	3	3	3	3	3	3	3	3	3	3

schedules	15:30	16:00	16:30	17:00	17:30	18:00	18:30	19:00	19:30	20:00	20:30	21:00	21:30	22:00	22:30
sunday	3	3	3	3	3	3	3	3	3	3	3	3	2	1	1
monday	3	3	3	3	2	2	2	2	2	2	2	2	2	1	1
tuesday	3	3	3	3	2	2	2	2	2	2	2	2	2	1	1
wednesday	3	3	3	3	2	2	2	2	2	2	2	2	2	1	1
thursday	3	3	3	3	2	2	2	2	2	2	2	2	2	1	1
friday	3	3	3	3	2	2	2	2	2	2	2	2	2	1	1
saturday	3	3	3	3	3	3	3	3	3	3	3	3	2	1	1

Fig: 6.3

Step 3: In step 3, the library coordinator will need to add the unavailability of each student as shown in the Figure 6.4. A value of "0" indicated that the student is not able to work that time period.

Sally

schedules	8:00	8:30	9:00	9:30	10:00	10:30	11:00	11:30	12:00	12:30	13:00	13:30	14:00	14:30	15:00
sunday	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
monday	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
tuesday	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
wednesday	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
thursday	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
friday	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
saturday	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

schedules	15:30	16:00	16:30	17:00	17:30	18:00	18:30	19:00	19:30	20:00	20:30	21:00	21:30	22:00	22:30
sunday	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
monday	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
tuesday	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
wednesday	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
thursday	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
friday	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
saturday	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fig: 6.4

Step 4: Finally, click "generate schedule" (as seen in Figure 6.5).

After the click event, VB macros collect the appropriate data from Excel's data cells, exports it to the back-end services and waits for the response. This back-end services create the appropriate data structures from the exported data and calls the **Model 1 LP** for each day of the week to generate the daily shifts schedules based on period demand. Then, it calls the **Model 2 LP** to assigned students to the generated shifts from Model 1. Finally, a weekly schedule is generated and control is returned back to Excel with notifications of any errors, i.e. if no feasible solution was found. See *Appendix B*.

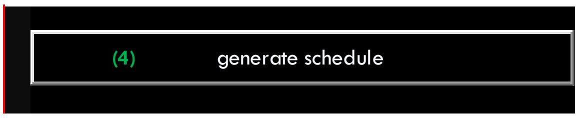


Fig: 6.5

LP optimization output and metrics

Figure 6.6 is the output of the optimization process after the back-end services are complete. It shows a series of shifts for each person for each day. For example, James is assigned to work from 11:30 to 6:30 on Tuesday.

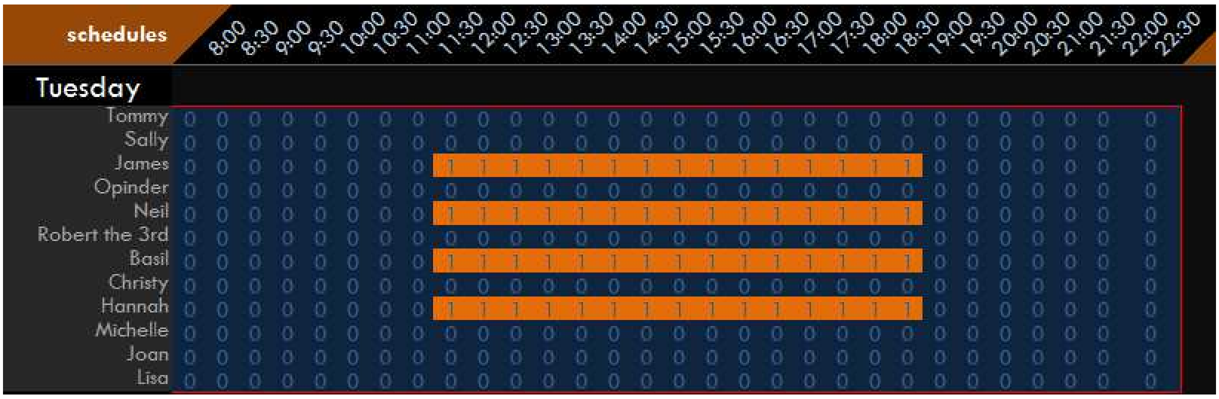


Fig: 6.6

Figure 6.7 displays a metric where the gray shaded region shows the needed coverage at a given demand period and the orange line the actual coverage.

Fig: 6.7

Conclusion

The models and application implementation presented in this paper will assist our sponsors to balance library demand and student preferences while reducing the uncertainty of the manual method. Our application has several advantages over the current method:

1. Quickly optimizes on weekly demand for any time period, e.g. finals, summer etc.
2. New students can quickly be added to the system
3. Changes in student class schedules and block periods can easily be updated
4. Changes in library hours for operation can easily be modified
5. A user friendly UI in a familiar environment (Excel)
6. Some metrics support to help analyze the demand vs. coverage for each day of the week

However, in order to clearly see the advantages, the authors encourage our sponsors to use both methods (manual and our proposed application) to determine, in the long run, which method produced the optimal results quickly and easily. Once the advantages are determined, this application can be used by other library departments or other departments in the University, e.g. student staff at PSU transportation services or the student rec center, etc.

Future work

Since this application was written as a class project in a quarter based term, time was the limiting resource. This required us to reduce scope and remove many features. The following are suggestions for enhancements to our application:

1. Currently, the Excel front-end application is limited to scheduling 12 students in 3 types of shifts: 2 hour, 4 hour and 6 hour. Since the LP model is generalized, the application front-end can be written in an alternative UI to support more students and shift types.
2. Data was exported in Excel's row/column format. It was tedious to convert this format to a general workable data structure. A data translation layer can be created to support universal data formats, such as XML.
3. New secondary goals can be integrated. For example, minimizing labor cost or maximizing student preference for day/time A over day/time B.
4. New reporting and usage history related metrics
5. Some of the improvements might introduce working with or storing sensitive data, which creates privacy issues. In this case, encryption routines need to be implemented to protect the data.

Appendix A

The 'WentToWork' scheduling software.

WhenToWork

Portland State University

September - 2011

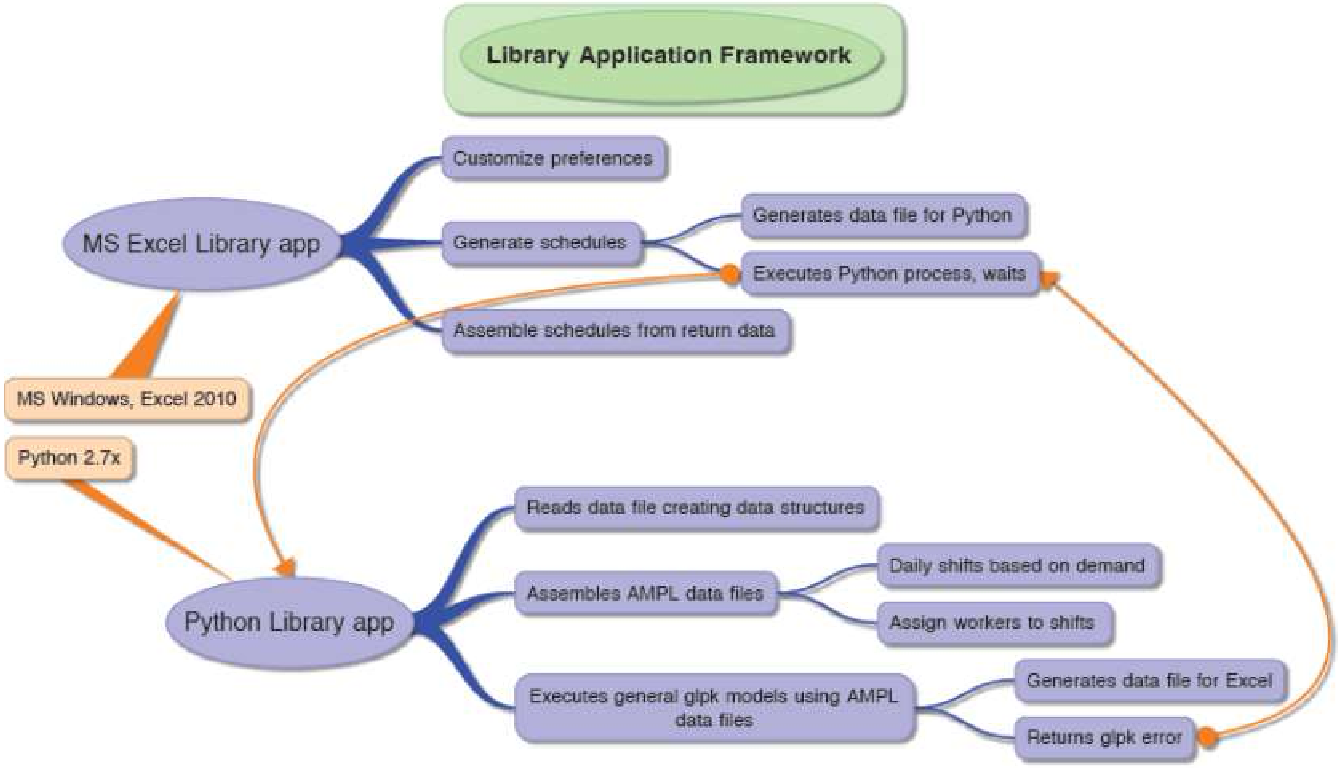
Circulation

Printed On Nov 2, 2011

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
				1	**2**	**3**
				Circulation 8am - 12pm [redacted] (deleted) 10am - 3:30pm [redacted] 12pm - 4pm [redacted] 3pm - 9pm [redacted] 4pm - 8pm [redacted]	Circulation 8am - 12pm [redacted] (deleted) 10am - 3pm [redacted] 3pm - 6pm [redacted]	
4	**5**	**6**	**7**	**8**	**9**	**10**
		Circulation 8am - 3:30pm [redacted] 12pm - 4pm [redacted] 3pm - 9pm [redacted] 4pm - 6:30pm (Unassigned)	Circulation 8am - 12pm [redacted] (deleted) 10am - 4:30pm [redacted] (deleted) 12pm - 6pm [redacted] 3pm - 9pm [redacted]	Circulation 8am - 12pm [redacted] (deleted) 10am - 3:30pm [redacted] 12pm - 4pm [redacted] 3pm - 9pm [redacted] 4pm - 8pm [redacted]	Circulation 8am - 12pm [redacted] 10am - 3pm [redacted] 3pm - 6pm [redacted]	
11	**12**	**13**	**14**	**15**	**16**	**17**
	Circulation 8am - 12pm [redacted] 10am - 4:30pm [redacted] (deleted) 12pm - 6pm [redacted]	Circulation 8am - 3:30pm [redacted] 12pm - 4pm [redacted] 3pm - 6am [redacted]	Circulation 8am - 12pm [redacted] 10am - 4:30pm [redacted] (deleted) 12pm - 6pm [redacted] 3pm - 6pm [redacted]	Circulation 8am - 12pm [redacted] 10am - 3:30pm [redacted] 12pm - 4pm [redacted] 3pm - 6pm [redacted]	Circulation 8am - 4:30pm [redacted] 3pm - 6pm [redacted]	
18	**19**	**20**	**21**	**22**	**23**	**24**
	Circulation 8am - 4:30pm [redacted] (deleted) 10am - 1:30pm [redacted] 12pm - 6pm [redacted]	Circulation 8am - 3:30pm [redacted] 12pm - 4pm [redacted] 1pm - 4pm [redacted] 3pm - 6pm [redacted]	Circulation 8am - 12pm [redacted] 10am - 4:30pm [redacted] (deleted) 12pm - 6pm [redacted] 1pm - 4pm [redacted] 3pm - 6pm [redacted]	Circulation 8am - 12pm [redacted] 10am - 3:30pm [redacted] 12pm - 4pm [redacted] 1pm - 4pm [redacted] 3pm - 6pm [redacted]	Circulation 8am - 3pm [redacted] 10am - 2pm [redacted] (deleted) 1pm - 4pm [redacted] 3pm - 6pm [redacted]	
25	**26**	**27**	**28**	**29**	**30**	
	Circulation 7:30am - 12pm [redacted]	Circulation 7:30am - 11:30am [redacted]	Circulation 7:30am - 12pm [redacted]	Circulation 7:30am - 11:30am [redacted]	Circulation 7:30am - 12pm [redacted]	

Appendix B

The Library Application Framework.



This is the application framework that was implemented for this project. The framework consists of a front-end service using Excel as the UI and a back-end services using Python and GLPK to process the LP models. This separation allows the front-end/back-end services to exist on any PC server as long as the interface between them does not change.

References

1. Lynn Gordon and Erhan Erkut, "Improving Volunteer Scheduling for the Edmonton Folk Festival," Interfaces, Vol. 34, No. 5 (Sep. - Oct., 2004), pp. 367-376.
2. David W. Ashley, "A spreadsheet optimization system for library staff scheduling," Computers and Operations research, Vol. 22, No. 6 (July, 1995), pp. 615-624.
3. Bernard Gendron, "Scheduling Employees in Quebec's Liquor Stores with Integer Programming," Interfaces, Vol. 35, No. 5 (Sep. - Oct., 2005), pp. 402-410.