

Improving Volunteer Scheduling for the Edmonton Folk Festival

Lynn Gordon

Faculty of Education, University of Alberta, Edmonton, Alberta, Canada T6G 2G5, lynn.gordon@ualberta.ca

Erhan Erkut

School of Business, University of Alberta, Edmonton, Alberta, Canada T6G 2R6, erhan.erkut@ualberta.ca

The annual Edmonton Folk Music Festival is run almost entirely by its 1,800 volunteers. While people are usually enticed into volunteering for the folk fest by perks, such as free access to the entertainment, gourmet meals, and T-shirts, their willingness to return year after year depends on an intangible degree of satisfaction. In spring 2003, a crew coordinator sought to automate the scheduling process for his crew of about 35 volunteers to save time and to accommodate volunteers' preferences when possible. We developed a spreadsheet-based decision-support tool that generated shift times, scheduled volunteers according to various constraints and preferences, and produced master and individual schedules.

Key words: programming: linear; organizational studies: personnel.

History: This paper was refereed.

The Edmonton Folk Festival is a four-day outdoor event that has been held annually since 1980 (<http://www.edmontonfolkfest.org/>). In 2002, 15,000 people attended the afternoon workshops and the evening main-stage performances featuring local artists and internationally renowned musicians. With only a handful of paid staff, the event's success and longevity are attributable to its volunteers. In 2002, 1,800 volunteers worked on 35 crews (for example, gate, kitchen, and security) contributing over 50,000 volunteer hours. In return for their hours of service, the festival gives volunteers free admission, T-shirts, and meals. Edmonton, known as "the City of Festivals," hosts over a dozen festivals and major sporting events during the summer. In a city with a population of 670,000, competition for volunteers is fierce. Attracting new recruits and maintaining its veteran volunteers from year to year is essential to the operation of the folk fest.

Crew coordinators schedule the volunteers, attend organizational meetings, and supervise their crews. We focus only on gate-crew scheduling of 30 to 40 volunteers. In previous years, the two volunteer coordinators of the gate crew used a trial-and-error, paper-and-pencil procedure to develop a set of shifts for gate operation over the four-day event and assigned volunteers to these shifts according to constraints, such as maximum hours per volunteer. Shifts had specified start and end times and were of varying lengths. Coordinators used overlapping shifts to deal with surges in attendance during peak hours.

The final schedule given to volunteers listed the set of shifts over the four days and the volunteers assigned to each shift. Drawing up the schedule was time consuming and frustrating. Minor changes often led to major revisions and numerous drafts. Even when the coordinators finished a final draft of their schedules, they were always uncertain whether they had accounted for all the constraints. Their only means of checking was to painstakingly go through the schedules manually to ensure that they had scheduled all the volunteers for the right number of hours and that no volunteers' shifts conflicted. The two gate-crew coordinators estimated that the hours required for scheduling equaled the hours required for all their other responsibilities.

To reduce the time they spent on this annual chore, one of the coordinators sought our assistance in automating the scheduling process. The festival had no funds for buying scheduling software; even if it had, the typical software programs would be unlikely to address the many constraints and volunteer preferences. In April 2003, the coordinator asked the University of Alberta School of Business' Centre for Excellence in Operations (<http://www.bus.ualberta.ca/ceo/>) for help. We agreed to provide resources for the project and to create a user-friendly spreadsheet-based decision-support tool that would generate shifts, allow the coordinators to input volunteer information and preferences, schedule volunteers by maximizing preferences, and produce schedules and reporting forms.

The Ifs, Ors, Ands, and Buts of Representing Volunteer Preferences

The general problem of scheduling employees or crews includes such tasks as forecasting demand, generating staffing requirements, scheduling shifts and tours, and rostering (Buffa et al. 1976; Ernst et al. 2004; Thompson 1998a, b; 1999). Of the considerable literature in this area, much is situation specific. The problem of scheduling volunteers for a four-day event has characteristics that separate it from most other employee-scheduling problems, such as the limited availability of volunteers (for both shifts and tasks), the acyclic nature of the problem, and the high priority placed on satisfying volunteer preferences. While addressing preferences may be of secondary importance in scheduling paid workers, it is of utmost importance in our problem. Volunteers who are dissatisfied with the schedule are much more likely to refuse to serve than paid workers. Events that cannot accommodate volunteers' preferences may lose volunteers to competing events that can.

Some researchers have dealt with problems that include limited employee availability: Glover et al. (1984), Glover and McMillan (1986), Loucks and Jacobs (1991), Love and Hoey (1990), Thompson (1990, 1996), and Litchfield et al. (2004). These authors all propose heuristics to solve their problems. In contrast, thanks to the small size of our problem, we were able to solve the problem of assigning volunteers to shifts optimally using a widely available commercial solver.

In our problem, the coordinators manually determine a set of shifts for the supervisors and another for the volunteers. They also estimate the minimum number of volunteers needed for each shift and at each of the two gate locations (top and bottom of the hill). The coordinators base their estimates of the number of volunteers needed on past experience of typical slow and peak periods and on current expectations of attendance for well-known performers. They develop the shifts with the expectation that each volunteer will work exactly 20 hours, the requirement for receiving the perks for the event. The typical rotation includes a four-hour shift Thursday, a four-hour shift Friday, two three-hour shifts Saturday, and two three-hour shifts Sunday.

With a set of shifts and the requirements needed at the two gates in hand, the coordinators had traditionally estimated the number of volunteers needed and using the list of names supplied by the festival's manager of volunteers, manually filled in the slots according to a set of implicit rules. One of the challenges throughout the project was to make the implicit rules explicit and to represent them appropriately in the model. This back-and-forth process extended what we

initially expected would be a two-week project into two months.

Our preliminary integer-programming model provided a schedule that satisfied the constraints that were easy to express: the schedule met the demand requirements for every time period, each person worked exactly 20 hours, and no one worked back-to-back shifts or late-night, early-morning combinations. In addition, we included options allowing coordinators to designate guaranteed shifts to volunteers and to input volunteer unavailability.

Upon completion of this first phase, we generated mock schedules. In reviewing the schedules, the coordinators expressed a number of implicit scheduling rules. Our discussions revealed that we needed to make a number of revisions:

—“Yes, everyone works 20 hours, but we need a number of people for prefest activities and for gate construction. If a person works prefest or construction hours then they'll work only three three-hour shifts (nine hours) on the weekend rather than four (12 hours).”

—“Wherever possible we ensure that we have a minimum number of experienced people on each shift.”

—“Could we distribute the people with first-aid training throughout the shifts, particularly at the top gate, which is away from the first-aid tent?”

—“Although, we don't want to schedule anyone on back-to-back shifts, we need to override the constraint when necessary.”

—“We generally try to create a balance between working the top and bottom gates. The top gate has a view of the main-stage performances, but the bottom gate does not. Similarly, we also try to balance day and night shifts. In particular, no one should work more than two night shifts.”

Using a variety of methods, we built these requests into the scheduler. We used constraints to ensure that 20 percent of the volunteers on each shift were experienced, that 15 percent had first-aid training, and that each volunteer—supervisor or crew member—had a maximum of two night shifts. We based these constraints on the coordinators' expectations and assumptions about volunteer preferences. To assist the coordinators with data entry, we created a worksheet that allowed them to input volunteer availability, to guarantee shifts, and to override constraints on an individual basis. Finally, we used goal programming to minimize the imbalance between the number of shifts scheduled at the top and bottom gates for each volunteer (appendix, Model 3).

We refined many of these constraints further during the project to better reflect the coordinators' implicit rules in scheduling or to account for other possibilities they realized the model could handle. For example,

we started out by allowing a maximum of two final shifts per volunteer starting at 7:30 P.M. or later. Then, we tightened this constraint to a maximum of two shifts starting at 6:00 P.M. or later. However, this constraint was too restrictive, so we changed to a penalty system. We assigned the evening shifts a penalty value between one and three and we assigned all others a value of zero. The Friday and Saturday night shifts, the most undesirable of the shifts, both had three penalty points. Each person was restricted to a maximum of five penalty points. We refined the balance between top and bottom gates further into day and evening shifts, because the volunteers preferred the top gate at night because of the view, but they preferred the bottom gate during the day because it was closer to the activities.

Once we incorporated these additional features, we determined the minimum number of volunteers needed based on the total number of volunteer hours required. On our advice, the coordinators requested 32 volunteers for the gate crew. One-third of the volunteers on the list they received were new to the gate crew, and three of the six supervisors from the previous year were returning. As was the routine, one of the coordinators called the volunteers to welcome them to the crew. In conversation and on the volunteer information sheets, many of the volunteers made further requests:

—“I have a wedding to go to on Saturday, so I’ll work two shifts on Thursday and Friday to make up for it.”

—“I’ll work the evening shift, but only if I can work with Steven.”

—“I prefer working the opening shifts.”

—“I refuse to work if I’m scheduled at the same time as Karen.”

—“I don’t think I’m able to climb the hill to work at the top gate. Please schedule me only at the bottom gate.”

These unsolicited requests are typical of those the volunteers made to the gate-crew coordinators. The coordinators’ official policy was that crew members had to be available for all shifts; they gave them no opportunities to state personal preferences. However, several vocal individuals consistently ignored the policy and by phone, e-mail, or on volunteer sheets stated that they were not available for some shifts (for example, during the day on Thursday and Friday), indicated shift preferences (for example, prefer morning shifts), or asked to be paired with a partner or spouse for personal reasons and transportation. Although they were aware they were sending mixed messages, the coordinators tried to accommodate such requests. Inevitably they got complaints that made them concerned that the volunteers would not return the following year.

Whether to solicit and accommodate preferences was a source of tension in our meetings. On the one hand, because they knew that the schedule was tight and that they had difficulty accommodating even a few requests, the coordinators feared setting volunteers’ expectations too high. They could not accommodate every request. They worried that asking volunteers for preferences and then not filling them would anger the volunteers. On the other hand, if they maintained their current policy, they feared volunteers would not feel valued and would feel that the coordinators were setting a double standard when they did accommodate some people.

We agreed that they would probably not be able to fill all requests, but we could develop a schedule that would accommodate as many preferences as possible. In fact, by incorporating volunteers’ preferences we would simplify the problem by reducing the number of feasible solutions. With no requests, the number of possible shift combinations for each person (in isolation) was over 1,200. However, a request to work on Friday night reduced the number of feasible shift combinations for that person to four because of consecutive shifts and evening shifts constraints. Also, by treating individuals who wanted to work together as a team with a labor supply of two, we would effectively reduce the number of people to be scheduled and simplify the problem further.

The final decision regarding preferences was that it would be too time consuming to solicit preferences from all volunteers. Instead, we would view this year as a test to see to what extent we could accommodate the preferences submitted.

The scheduling model tried to satisfy the constraints and preferences the coordinators set (for example, balance of gates and maximum night shifts) and volunteer preferences. We used a point system to handle volunteer preferences. In this initial year, we assigned points to the indicated preferences at the coordinators’ discretion. In subsequent years, we would need a method for distributing points. Volunteers could specify their first, second, and third choices and allot a predetermined value to each. Or we could give volunteers points (either equally or based on experience), and they could independently distribute their points strategically (that is, assigning more points for shifts that would likely be in high demand). The preference system we used is flexible enough to allow volunteers to assign negative points for undesired shifts (rather than use the unavailability feature). For example, a volunteer could assign 50 points to the Friday evening shift, 30 points to the Sunday afternoon shift, and –20 points to the Sunday morning shift. The objective of the volunteer-to-shift assignment model (appendix, Model 2) is to maximize the number of volunteer preference points.

Revising the Set of Shifts

During the process of designing the scheduling model, the coordinators changed the manually determined shifts for supervisors and volunteers six times. Each time, we had to adjust the scheduler. Determining a suitable set of shifts was difficult for several reasons. First, in the previous year the volunteers worked four five-hour shifts (one per day); however, some volunteers complained that five hours was too long, and the coordinators were trying to develop a different and untested set of shifts. Second, the festival provided meals to volunteers during set hours. Volunteers working during those hours were entitled to delivered meals, which created logistics problems. Also, volunteers left the gates to eat their meals. The coordinators struggled to develop shifts to minimize or, preferably, to eliminate the number of delivered meals. Finally, the coordinators were not certain how many volunteer hours were available on Saturday and Sunday, because many of the volunteers who worked prefestival hours reduced either their Saturday or Sunday hours from six to three. Although the hours from open to close on Saturday were longer and the demand was greater than on Sunday, when developing schedules manually the coordinators addressed each day separately and assumed that the same number of volunteer hours were available on both days. As a result of these difficulties, the shifts the coordinators generated, particularly for Saturday and Sunday, were overstaffed during slow periods (late morning) and understaffed at peak times (early evening) (Figure 1).

After the coordinators abandoned a number of unsatisfactory sets of shifts because of conflicts, concerns about the evening understaffing, poor use

of volunteer time in overstaffing, and uncertainty as to the number of volunteer hours available on Saturday and Sunday due to prefestival activities, we worked with the coordinators to develop shifts that would meet their needs. We welcomed this challenge because it gave us more flexibility and an opportunity to make a greater impact than we would by producing a tool for slotting individuals into a set schedule. We developed an optimization tool (appendix, Model 1) that determined two-, three-, and four-hour shifts for Saturday and Sunday based on the volunteer hours available while accounting for prefestival hours. We added constraints to ensure that the same numbers of people staffed the two-hour shifts and the four-hour shifts so that we met the requirement of six hours per day. As a result, we were able to develop a smoother shift set that met all demand requirements, did not overstaff in slow periods, avoided the need to deliver meals, and was consistent with the volunteers' general request to limit shifts to four hours (Figure 2). A total of six volunteers on Saturday and 10 on Sunday worked the two- and four-hour shifts, and the remaining volunteers worked the usual three-hour shifts.

The coordinators were pleased with the result and approved the proposed set of shifts. Barring any unforeseen problems or changes in the future, they envisioned using the same set of shifts in the years to come as the shifts met all of their needs.

Comprehensible Schedules

Once we entered the information for individual volunteers, incorporated all preferences, and selected a

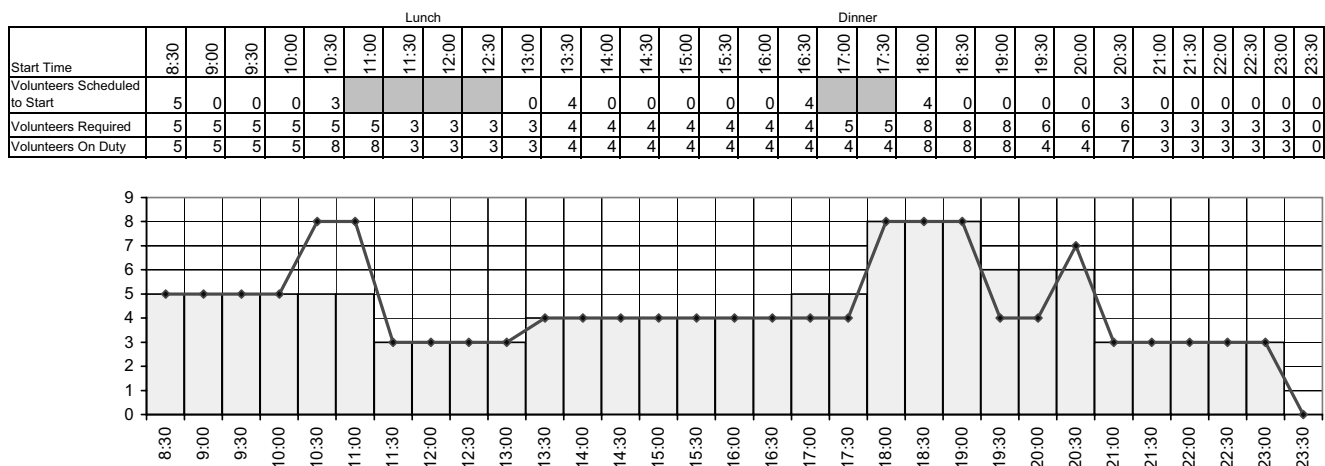


Figure 1: The coordinators determined the number of volunteers required (bar graph). The model produced the number of volunteers to schedule for three-hour shifts and their start times for the Saturday shifts at the top gate. We indicate the constraint on start times during meals in gray. We restricted these times to eliminate meal deliveries. The line graph indicates the total number of volunteers on duty. Although we examined demand and supply in 15-minute intervals, we show them in 30-minute intervals.

	Lunch												Dinner																		
Start Time	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	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	23:00	23:30
2 hours	0	0	0	0	0	0	3			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3 hours	5	0	0	0	0					0	4	0	0	0	0	0	5			3	0	0	0	0	0	0	0	0	0	0	0
4 hours	0	0	0							0	0	0	0	0	0					0	0	0	3	0	0	0	0	0	0	0	0
Volunteers Required	5	5	5	5	5	5	3	3	3	3	4	4	4	4	4	4	4	5	5	8	8	8	6	6	6	3	3	3	3	3	0
Volunteers On Duty	5	5	5	5	5	5	3	3	3	3	4	4	4	4	4	4	5	5	5	8	8	8	6	6	6	3	3	3	3	3	0

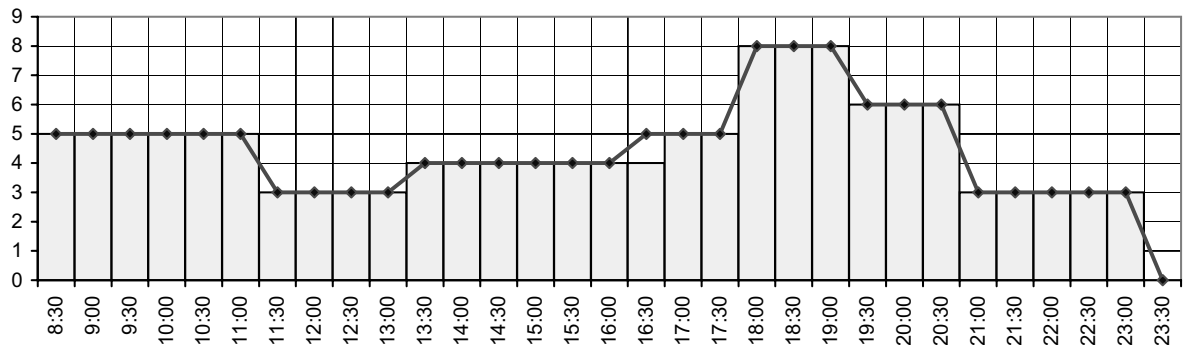


Figure 2: The coordinators determined the number of volunteers required (bar graph). The model produced the number of volunteers to schedule for two-, three-, and four-hour shifts and their start times for the Saturday shifts at the top gate. We indicate the constraint on start times during meals in gray. The line graph indicates the total number of volunteers on duty. The result is a supply curve, allowing for varying shift lengths, that matches the demand with less understaffing or overstaffing than previously existed.

shift pattern, the next task was to create a final schedule that the coordinators could distribute to the volunteers. In the previous years, the coordinators had created the schedule manually using word-processing software. They created each shift with a corresponding list of volunteers for each day of the event. The process was time consuming. In 2002, they attempted to reduce the time they spent on this onerous task by assigning codes to the volunteers, but this proved frustrating because the volunteers had to search for their codes and could not determine who else had been assigned to their shifts. While this schedule ensured that the coordinators had assigned the appropriate number of volunteers to each shift, they could not easily tell whether an individual's set of shifts satisfied the constraints on hours worked and back-to-back shifts, or met personal and coordinator preferences concerning shifts desired, night shifts, or gate balancing.

We produced three schedules that met their needs for error checking and readability. The first was a master schedule. Using native Excel functions, we produced a master schedule immediately after solving the integer-programming problems. The schedule provided a table of all volunteers and all shifts. Using the master schedule, the coordinators could view who was and who wasn't assigned to the different shifts and whether anyone had been scheduled for back-to-back shifts. They could also look across at each

person's assigned set of shifts quickly to assess night shifts and gate balance.

We developed a second version of the schedule using VBA that was similar to schedules produced in previous years. This version included all shifts and listed the volunteers working each shift. It assured the coordinators that they had assigned an appropriate number of people to each shift. We also produced individual schedules to eliminate errors by volunteers who failed to locate their names under all their shifts. This third schedule provided each volunteer with a list of his or her shifts for the event. Although producing the three schedules was one of the simplest components of the project (because it involved no modeling or optimization, but merely displaying the results in different formats), the coordinators and volunteers valued them highly. We could create and print a schedule in seconds rather than hours, which made last minute changes possible.

By mid-June we had produced a final draft of the schedule and distributed it to the volunteers. In the regular volunteer schedules, we were able to assign all partners to the same shift, except in one instance when the partners' personal schedules conflicted. In this case, we assigned them to work all shifts together except the one shift in conflict. Each volunteer had a balance between top and bottom gates in both the day and evening events. No one had back-to-back or late-night, early-morning combinations unless he or she requested it. Volunteers with first-aid training were

spread throughout the shifts. One violation occurred in the number of experienced volunteers scheduled because a number of new volunteers stated they had to work Friday evening. Most important, we incorporated all articulated volunteer preferences. Similarly, we satisfied all constraints and preferences related to the supervisors. We were also able to incorporate split duties, whereby some volunteers worked as supervisors for some shifts and as regular volunteers for others.

The coordinators were very pleased with the results. At this point, they fully realized the value of asking volunteers for their preferences, and they were excited to be able to offer them the opportunity to voice their preferences in the years to come.

Making It User Friendly

Throughout the project, we assured the coordinators that we would be available the following year to help them with problems or unique situations arising from volunteer requests. However, they wanted a product that they could use independently the next year. Working with one coordinator, we developed a decision-support tool in Excel that would run each

part of the process automatically. He played an essential role in checking errors, and as a result, the tool became usable by someone who was comfortable with Excel but who had no experience with Solver or VBA.

The final product was a set of 11 worksheets and a 10-page instruction manual. The start page (Figure 3) provided brief instructions and command buttons that transferred information, initiated Premium Solver (Frontline Systems 2003), and wrote VBA-generated schedules. From the start page, the coordinators worked through the scheduling process.

The coordinators used two worksheets to enter data on volunteers and to generate forms (for example, a telephone list and a final evaluation report). They entered data on collapsible worksheets to compile all the information related to each supervisor and each volunteer, including teams and individuals, first-aid training, required hours, availability for each shift, guaranteed shifts, personal preferences, and overrides on standard constraints (for example, back-to-back shifts).

The coordinators entered shift times and staffing requirements on one worksheet, and they were automatically transferred to other worksheets. To solve shifts for supervisors and regular volunteers, we had

**2003 Folk Festival
Gate Crew Scheduling**

Shifts

Data Entry

Supervisors

Volunteers

Gates-partners

Gates-singles

Schedules

- 1 **Crew** worksheet: Enter all personal information
- 2 **Shifts** worksheet: Enter all Shift times and requirements for top and bottom gates
- 3 **Data** worksheet:
 - Put partners together on the same line
 - Separate supervisors, experienced and new volunteers
 - Enter First-Aid Training: 0=No, 1=Yes
 - Enter required hours
 - Availability: 0 = Not Available, 1 = Available (default)
 - Guaranteed: 0 = Not Guaranteed, 1 = Guaranteed
 - Preferences: Enter points for preferred shifts
 - Override any constraints if necessary
 - When complete press Data Entry
- 4 Solve scheduling for Supervisors and Volunteers
 - Press Supervisors
 - Press Volunteers
 - Press Gates-partners. This will schedule people working in partners.
 - Press Gates-singles. This will schedule remaining individuals.
- 5 **Schedules: Msched, Sched, IndSched** worksheets
 - Msched:** Master Schedule is an overview of each person and shift.
 - Sched:** Compressed Schedule. Lists each person on each shift.
 - IndSched:** Provides a list of all shifts for each volunteer.

Figure 3: The start page leads the coordinators through a five-step scheduling process. In steps one to three, the coordinators enter all volunteer information, shift times, and supply requirements. In step four, Solver is invoked and assigns supervisors and volunteers to shifts and to gates. In step five, Excel and VBA produce the three schedules.

to use a four-stage process to keep the number of decision variables below the maximum allowed for the coordinators' old version of Premium Solver:

—We used one worksheet for the supervisors' schedule.

—In preparing the regular volunteers' schedule, Solver assigned volunteers to shifts according to their preferences and such general constraints as required hours and minimum staffing levels.

—In assigning partners together, the model split the volunteers for each shift into two groups for the top and bottom gates. To reduce the number of decision variables, Solver assigned top and bottom gates to volunteers working in partners only in the first stage. One of the general constraints was to set a maximum number of partners. That is, if we needed six people on a shift, we could assign a maximum of two pairs to that shift, because we could not split up a third pair and assign the two individuals to separate gates.

—After assigning partners to jobs, the model calculated the number of volunteers we still needed at each gate and filled these openings with the remaining individuals.

The coordinators could initiate all four scheduling worksheets from the start page, or they could go to each worksheet and select Solver from the tools. Although it would have been possible to run the worksheets consecutively, a black-box automation method could take an uncomfortably long time. By executing the worksheets separately, we could allow the coordinators to see progress and play with the schedule manually.

When we tested the tool with one of the coordinators, a number of questions and problems arose. He had some trouble understanding the worksheets themselves, and he also pointed out potential errors in various situations. He had problems recognizing when demand for volunteer hours exceeded supply overall or supply for particular shifts, or when he had violated constraints concerning first-aid and experienced personnel. To address the issues the coordinator raised and to accommodate manual changes, we modified the tool using Excel's automatic formatting feature to flag nondefault values, violated constraints, and unrealized volunteer preferences.

At the end of June 2003, the coordinators felt comfortable using the scheduler independently. They were able to generate test schedules that proved to be accurate under a host of conditions. The time they took to complete the process depended on a variety of factors; however, by limiting the number of feasible solutions by entering a reasonable number of preferences, available volunteers, and guaranteed shifts, they could complete the semiautomated process in less than an hour.

The festival took place on August 7th through the 10th, 2003. The coordinators used the tool we designed successfully to develop the shifts and the schedules. The project was successful because we combined optimization tools and what-if features on a spreadsheet, and we cooperated closely with the end-users. The tool could be useful for many years to come, although we suspect that, having tasted the power of optimization, the coordinators may want further refinements for next year.

Appendix

Model 1: Determining Daily Shift Schedules

Decision Variables

x_j = number of volunteers assigned to shift j .

(Shifts are determined by a start time and duration. For example, Shift 1 can be a two-hour shift starting at 8:30, and Shift 2 can be a three-hour shift starting at 10:00. Only shifts that do not require a meal delivery are allowed.)

y_i = number of surplus volunteers during 15-minute period i .

(y_1 corresponds to 8:30–8:45; y_2 corresponds to 8:45–9:00, and so on.)

Parameters

C_i = number of volunteers needed during 15-minute period i .

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

Index Sets

S_{i2} = index set of two-hour shifts on Day i (1 = Saturday, 2 = Sunday).

S_{i4} = index set of four-hour shifts on Day i (1 = Saturday, 2 = Sunday).

Model

$$\begin{aligned} \text{Min } & \sum_i y_i \\ \text{s.t. } & \sum_j a_{ij}x_j - y_i = C_i \quad \forall i, \\ & \sum_{j \in S_{i2}} x_j = \sum_{j \in S_{i4}} x_j, \quad i = 1, 2, \\ & x_j, y_i \geq 0, \text{ integer } \quad \forall i, j. \end{aligned}$$

Notes

(1) The organizers opted for 15-minute intervals for possible shift start times to allow for more flexibility in scheduling based on when the gates were to open, the anticipated demand, and constraints related to meal times. In the final schedule, shift start times of XX:15 and XX:45 existed on Friday and Sunday but not on Thursday or Saturday.

(2) It is possible to eliminate the y -variables from the formulation using substitution, but we opted for the above formulation for transparency reasons.

(3) The model could be used to solve the problem for all four days combined. However, the coordinators were content to select the Thursday and Friday shifts manually. Hence, we solved the model only for Saturday and Sunday. For Saturday and Sunday, two-hour, three-hour, and four-hours shifts are allowed. The second constraint set ensures that the number of volunteers assigned to two-hour shifts is the same as the number of volunteers assigned to four-hour shifts on each day.

(4) Festival organizers set the hours of daily gate operation. The time period per day determines the number of variables and constraints required for the problem. The Saturday-and-Sunday problem requires 204 variables (that is, 204 15-minute intervals) and 122 constraints.

Model 2: Assigning Volunteers to Shifts

Decision Variables

$x_{ij} = 1$ if volunteer (pair or individual) i is assigned to shift j , and 0 otherwise.

(The shift sets used in Models 1 and 2 are not the same. Model 2's shift set is a subset of Model 1's shift set. Model 2 uses only the shifts selected by Model 1.)

Parameters

n_i = population of volunteer team i (1 for individuals, 2 for pairs).

t_j = length of shift j .

p_j = penalty value for shift j (measure of undesirability).

u_{ij} = preference expressed by volunteer i for shift j .

D_j = number of volunteers needed during shift j .

Index Sets

T = index set for Thursday shifts.

F = index set for Friday shifts.

Sa = index set for Saturday shifts.

Su = index set for Sunday shifts.

C = set of all pairs of consecutive (back-to-back) or overlapping shifts.

M = index set for morning shifts.

A = index set for afternoon shifts.

E = index set for evening shifts.

P = index set for prefestival shifts.

O = index set for experienced volunteers.

R = index set for volunteers with first-aid training.

Model

Assign necessary number of volunteers to each shift:

$$\sum_i n_i x_{ij} = D_j \quad \forall j.$$

Every volunteer has to work 20 hours in total:

$$\sum_j t_j x_{ij} = 20 \quad \forall i.$$

One shift on Thursday:

$$\sum_{j \in T} x_{ij} = 1 \quad \forall i.$$

One shift on Friday:

$$\sum_{j \in F} x_{ij} = 1 \quad \forall i.$$

No more than six hours on Saturday:

$$\sum_{j \in Sa} t_j x_{ij} \leq 6 \quad \forall i.$$

No more than six hours on Sunday:

$$\sum_{j \in Su} t_j x_{ij} \leq 6 \quad \forall i.$$

No conflicting shifts:

$$x_{ij} + x_{ik} \leq 1 \quad (j, k) \in C \quad \forall i.$$

No more than one morning shift:

$$\sum_{j \in M} x_{ij} \leq 1 \quad \forall i.$$

No more than one afternoon shift:

$$\sum_{j \in A} x_{ij} \leq 1 \quad \forall i.$$

No more than one prefestival (setup) shift:

$$\sum_{j \in P} x_{ij} \leq 1 \quad \forall i.$$

No more than five penalty points:

$$\sum_j p_j x_{ij} \leq 5 \quad \forall i.$$

At least 20 percent of volunteers on a shift must be experienced:

$$\sum_{i \in O} n_i x_{ij} \geq 0.2 \sum_i n_i x_{ij} \quad \forall j.$$

At least 15 percent of volunteers on a shift must have first-aid training:

$$\sum_{i \in R} n_i x_{ij} \geq 0.15 \sum_i n_i x_{ij} \quad \forall j.$$

Maximize volunteer preferences:

$$\max \sum_i \sum_j u_{ij} x_{ij}.$$

Binary decision variables:

$$x_{ij} \geq 0, \quad \text{binary.}$$

Model 2 contains 506 decision variables and 465 constraints. We applied a similar (and much smaller) model to scheduling shift supervisors. It is possible for the coordinators to set some of the decision variables equal to one (modeling of preassignment) and some equal to zero (modeling of unavailability). These can be thought of as extreme ways of modeling preferences.

Model 3: Assigning Volunteers to Gates

Decision Variables

$x_{ijk} = 1$ if volunteer (pair or individual) i is assigned to shift j at gate k , and 0 otherwise.
 ($k = 1$ corresponds to the top gate and $k = 2$ corresponds to the bottom gate.)
 a_i, b_i, c_i, d_i = deviational variables (slack and surplus variables).

Parameters

G_{jk} = number of volunteers needed at gate k during shift j .
 (If D_j in Model 2 is even, then $G_{jk} = D_j/2$. If D_j is odd, then $G_{j1} = G_{j2} + 1$ and $G_{j1} + G_{j2} = D_j$.)
 $v_{ij} = 1$ if volunteer (pair or individual) i is assigned to shift j , and 0 otherwise.
 (This is the optimal solution to Model 2.)

Model

The objective is to minimize the sum of the deviations:

$$\min \sum_i (a_i + b_i + c_i + d_i).$$

Consistency between shift assignments and gate assignments:

$$x_{ij1} + x_{ij2} = v_{ij} \quad \forall i, j.$$

Assign necessary number of volunteers to each gate during each shift:

$$\sum_i n_i x_{ijk} = G_{jk} \quad \forall j, k.$$

Gate balancing for the day (shift start time prior to 18:00):

$$\sum_{j \in MUA} x_{ij1} - \sum_{j \in MUA} x_{ij2} + a_i - b_i = 0 \quad \forall i.$$

Gate balancing for the evenings (shift start time at 18:00 or later):

$$\begin{aligned} \sum_{j \in E} x_{ij1} - \sum_{j \in E} x_{ij2} + c_i - d_i &= 0 \quad \forall i, \\ a_i, b_i, c_i, d_i &\geq 0 \quad \forall i, \\ x_{ijk} &\geq 0, \quad \text{binary.} \end{aligned}$$

Model 3 requires 968 decision variables and 424 constraints. It is possible to combine Models 2 and 3, which would result in a larger model. In that case, one can use a weighted combination of the two objectives (volunteer preferences and gate balance) to generate multiple solutions. We have been able to solve the combined problem optimally.

If Model 3 is to be solved separately from Model 2, it is possible to reduce the size of Model 3 by eliminating all variables that correspond to volunteer-shift assignments not selected by Model 2 (as well as the first constraint set). The following variable definition reduces the problem size even further:

$x_{ij} = 1$ if volunteer (pair or individual) i is assigned to shift j at Gate 1, and 0 if volunteer (pair or individual) i is assigned to shift j at Gate 2.

References

- Buffa, E. S., M. J. Cosgrove, B. J. Luce. 1976. An integrated work shift scheduling system. *Decision Sci.* 7(4) 620–630.
- Ernst, A. T., H. Jiang, M. Krishnamoorthy, D. Sier. 2004. Staff scheduling and rostering: A review of applications, methods and models. *Eur. J. Oper. Res.* 153(1) 3–27.
- Frontline Systems, Inc. 2003. <http://www.solver.com>.
- Glover, F., C. McMillan. 1986. The general employee scheduling problem: An integration of MS and AI. *Comput. Oper. Res.* 13(5) 563–573.
- Glover, F., C. McMillan, R. Glover. 1984. A heuristic programming approach to the employee scheduling problem and some thoughts on “managerial robots.” *J. Oper. Management* 4(2) 113–128.
- Litchfield, J. A., A. Ingolfsson, J. Cheng. 2003. Rostering for a second cup. *INFOR* 41(3) 287–300.
- Loucks, J. S., F. R. Jacobs. 1991. Tour scheduling and task assignment of a heterogeneous work force: A heuristic approach. *Decision Sci.* 22(4) 719–738.
- Love, R. R., Jr., J. M. Hoey. 1990. Management science improves fast food operations. *Interfaces* 20(2) 21–29.
- Thompson, G. M. 1990. Shift scheduling in services when employees have limited availability: An L.P. approach. *J. Oper. Management* 9(3) 352–370.
- Thompson, G. M. 1996. A simulated-annealing heuristic for shift scheduling using non-continuously available employees. *Comput. Oper. Res.* 23(3) 275–288.
- Thompson, G. M. 1998a. Labor scheduling, part 1: Forecasting demand. *Cornell Hotel Restaurant Admin. Quart.* 39(5) 22–31.
- Thompson, G. M. 1998b. Labor scheduling, part 2: Knowing how many on-duty employees to schedule. *Cornell Hotel Restaurant Admin. Quart.* 39(6) 26–37.
- Thompson, G. M. 1999. Labor scheduling, part 3: Developing a workforce schedule. *Cornell Hotel Restaurant Admin. Quart.* 40(1) 86–96.

Vicki Fannon, Manager of Volunteers, Edmonton Folk Music Festival Society, PO Box 4130, Edmonton, Alberta, Canada T6E 4T2, writes: “I am pleased to tell you about the wonderful scheduling program that

was developed for us by Lynn Gordon and Erhan Erkut. It has already saved us a great deal of time, while making schedules for our gate crew more balanced, fair, and efficient. We hope to be able to extend its use to other crews next year, in order to take full advantage of its functionality.

"First, it might be useful to tell you a little bit about us. The Edmonton Folk Music Festival began in 1980 and it rapidly grew in both its product and its reputation. Now, in its 24th year, the EFMF is one of the premier festivals in North America. We offer more than 60 acts over four days of programming, and we sell 10,000 weekend passes. With day ticket holders, volunteers, and other guests, we typically have audiences of more than 15,000. A key to making it work is our volunteer base. We depend on more than 1,800 volunteers in crews ranging from security, to gate, to site construction, and to the festival kitchen.

"One of the most difficult tasks for crew coordinators is developing schedules. Each volunteer must serve for 20 hours during the four days of the festival. The gate crew, for example, must arrange for the 40 volunteers to staff three locations for a total of 47.5 hours over the four days. It is also necessary to schedule around the festival kitchen's meal times. In addition, it is desirable to balance the shifts so that no volunteer gets a disproportionate share of early morning hours or conflicts with the mainstage show. We also like to balance the crews with rookies and veterans, and to place at least one first-aid trained volunteer on each shift. As you can see, this is a very difficult task. In addition, volunteers have preferences or conflicts with work and family that they would like for us to take into account, if possible. We have always done the scheduling manually, with the result that it has required scores of manhours and has never achieved entirely satisfactory results. The new software makes it possible to generate a much better schedule in a few hours, rather than the days and sometimes weeks that have been required before.

"The software that Gordon and Erkut developed runs on an MS Excel platform. Since so many of our coordinators are familiar with Excel, this application is easy to learn. What's more, the intuitive interface makes the process a snap. There is also a great deal of flexibility so that we can enter constraints (such as avoiding consecutive shifts), availability of volunteers, guaranteed shifts when a volunteer must be scheduled, volunteers' preferences, and volunteers' characteristics (e.g., a rookie, trained in first aid, wants to work with a partner). This has not only made the process of scheduling much easier, but it has also made it possible to produce much more fair and efficient schedules. This year, only one person needed to trade one shift, compared to a much larger number of such problems in previous years. We have received a number of compliments from volunteers concerning their schedules, and no complaints. Another significant advantage of the system is the ability to make adjustments quickly and easily when volunteers resign and/or new volunteers join the crew at the last minute. Manually fitting people into an established schedule can be most frustrating, but this system makes it is very straightforward.

"As the development progressed, we began to understand the benefits of this approach and we began to ask for more and more functionality. The system grew well beyond its initial parameters and we can see opportunities for further improvement next year. As one example, we were having difficulty with scheduling volunteers into the shifts we had set. Gordon and Erkut suggested that we reconsider the shifts themselves, and offered a new schedule that solved the problem. We are grateful for the developers' patience in teaching us the advantages of the modelling approach in general, and in developing this application for us.

"We are very pleased with this application. We definitely plan to continue using it with the gate crew and we hope to extend it next year."