Labor Scheduling, Part 3

Developing a Workforce Schedule

by Gary M. Thompson

"Who will help me bake this bread?," asked the little red hen. "Not I," said the dog. "Not I," said the cat. "Not I," said the mouse. "Then I will," said the little red hen.

Workforce scheduling is a balancing act that requires managers to take into account a number of factors. Among those factors are the many issues involving employees, including their availability, their abilities, their desire to work, and the cost of their labor.

The balancing act also involves pressure on the manager to keep costs down, and labor often represents a large portion of controllable costs. At the same time, managers are expected to maintain appropriate customer-service levels. This article, the third in a series on workforce scheduling, examines scheduling issues and proposes methods that balance the firm's and the employees' goals.

As I have explained previously on these pages, workforce scheduling comprises four tasks. The first task is to forecast customer demand for the service.[1] The goal of that task is to predict the characteristics of the service transaction that change over time, such as customer-arrival rates. The second task is to translate the forecasts of customer

[1] As discussed in the first article in this series: Gary M. Thompson, "Labor Scheduling, Part 1: Forecasting Demand," *Cornell Hotel and Restaurant Administration Quarterly*, Vol. 39, No. 5 (October 1998), pp. 22–31.

Gary M. Thompson, *Ph.D., an associate professor of operations management at the Cornell University School of Hotel Administration, has written a four-part series on labor management, of which this is Part 3; Parts 1 and 2 appeared in the October and December 1998 issues of* Cornell Quarterly.

demand into employee requirements.[2] The outcome of that task is a calculation of the number and skill levels of employees needed to serve customers adequately throughout a planning cycle, which is usually a week or more of work shifts. The third task is to develop a work schedule that supplies sufficient staffing while also accounting for employee requirements (such as availability). The workforce schedule specifies which employees are working, where they are working, and when they are doing it. Like the other two tasks connected with developing a schedule, creating the schedule is a planning activity that occurs before the service is delivered. A fourth task, to be discussed in a forthcoming article, is controlling and administering the schedule as the work unfolds.

The current article describes the issues involved in developing a workforce schedule. I explain the considerations for creating the schedule that account for employees' availability and preferences while also meeting the organization's specific service and economic objectives. I have developed these workforce-scheduling ideas and concepts in my teaching, research, and consulting activities during the past 13 years.

Overview of Workforce Scheduling

Most workforce-scheduling routines use one of two classic frameworks. As a prelude to discussing the scheduling procedure I recommend, I'll discuss the nature and limitations of those frameworks.

Scheduling criteria. Workforce-scheduling routines usually seek either to minimize the cost of the schedule (subject to constraints that ensure some level of staffing) or to maximize the schedule's overall benefit to the organization. Managers also try to draw up a schedule that will make the employees reasonably satisfied with the number and timing of their work hours. Unfortunately, developing a schedule that meets the goal of minimum cost or maximum benefit can conflict with employee preferences. I will return to this matter later in this article.

Classic frameworks. The first classic framework for developing a workforce schedule was presented in 1954 by George Dantzig.[3] Dantzig's framework for workforce scheduling, which I'll call the "D-framework," carries the objective of minimizing schedule cost subject to the proviso that the number of employees scheduled must equal or exceed the number of employees required for each period. Stated another way, the D-framework attempts to minimize the cost of providing the service while meeting or exceeding the employee requirements in every planning period. While Dantzig's framework prohibits understaffing, it permits overstaffing. However, its objective of minimizing the schedule cost should eliminate overstaffing, if at all possible.

Although scheduling routines that apply the D-framework are commonly in use, the approach has two shortcomings. First, the approach does not account for actual employee availability. It looks at minimum cost for a given service standard without considering whether employees can or will work a given shift. Should employees not be available as specified by a D-framework schedule, the manager cannot meet or exceed staffing requirements in all periods and the result would be no feasible schedule. Thus, the D-framework approach can break down when one attempts to implement it, because some employees are available for work only at individually specified times.

A second shortcoming of the D-framework is its prohibition of understaffing. When managers attempt to schedule employees using the D-framework approach, I've observed that they often circumvent the understaffing prohibition to avoid pronounced overstaffing. They do this by reducing the employee requirements in some or all of the periods that have no overstaffing, thus developing a schedule that better balances understaffing and overstaffing.

To remedy some of the shortcomings of the D-framework, Elbridge Keith presented a framework for scheduling that takes into account both understaffing and overstaffing.[4] Proposed in 1979, the so-called K-framework allows understaffing as it works toward an objective of minimizing schedule cost. To do so, the K-framework calculation assigns a pseudo cost—a false penalty—to under- and overstaffing. The formula uses this equation: number of employees scheduled to work in a period, plus understaffing in the period, minus overstaffing in the period equals the number of employees needed in the period (which is the number that is known, based on earlier analysis).[5]

The K-framework develops a workforce schedule by attempting to minimize the real labor-related

[2] As explained in the second article in this series: Gary M. Thompson, "Labor Scheduling, Part 2: Knowing How Many On-duty Employees to Schedule," *Cornell Hotel and Restaurant Administration Quarterly,* Vol. 39, No. 6 (December 1998), pp. 26–37.

[3] George B. Dantzig, "A Comment on Edie's 'Traffic Delays at Toll Booths,'" *Operations Research*, Vol. 2, No. 3 (1954), pp. 339–341.

[4] E. G. Keith, "Operator Scheduling," *AIIE Transactions*, Vol. 11, No. 1 (1979), pp. 37–41.

[5] Mathematically: $S_p + U_p - O_p = r_p$, where S is the number of employees scheduled to work, U is understaffing, O is overstaffing, r is the required number of employees, and p is a given period. Thus, mathematically one could be both overstaffed and understaffed by the same amount (e.g., 10 + 2 - 2 = 10). The false penalties on U and O, however, will result in either understaffing or overstaffing, but not both.

Terms Used in this Paper

The accompanying article employs a number of terms that may not be familiar to the reader. The terms are used as follows.

A *scheduling horizon* is a period of typically one week to several months for which schedules are developed at one time.

Planning periods, which are also called *planning intervals*, are subsets of the scheduling horizon. These are the detailed intervals used for staff planning. A planning-interval duration of 15 to 30 minutes is commonly used in service industries.

Overstaffing, or *surplus staffing*, is a situation in which more employees are scheduled in a planning period than are ideally needed.

Understaffing, or *short staffing*, is a situation in which fewer employees are scheduled in a planning period than ideally needed.

Employee requirements, or *ideal staffing levels*, are a key input to the task of developing a workforce schedule.

One can translate demand forecasts into employee requirements in three ways, as described in the second article in this series.* To determine employee requirements by job per planning period, one can use productivity standards, which use stable productivity factors; service standards, which use consistent service levels; or economic standards, which use economic tradeoffs.—*G.M.T.*

*Gary M. Thompson, "Labor Scheduling, Part 2: Knowing How Many On-duty Employees to Schedule," *Cornell Hotel and Restaurant Administration Quarterly*, Vol. 39, No. 6 (December 1998), pp. 26–37.

costs of delivering the service and the pseudo costs or penalties associated with under- and overstaffing. As the level of overstaffing or understaffing increases, the penalty for such variance increases in two steps from a low level to a high level. With those stepwise, or tiered, costs, the K-framework usually avoids large staff shortages or surpluses. Even so, the K-framework does permit extreme understaffing—in which no employees are scheduled in a period. As with the D-framework, a manager would have to intervene at this point and recast the schedule using different assumptions.

The academic literature on workforce scheduling has examined the venerable D-framework more than twice as often as the K-framework. Interestingly, most commercial workforce-scheduling software packages implement forms of the K-framework. Moreover, I have found that the K-framework, suitably modified to prohibit extreme understaffing, is generally more effective at satisfying the chief scheduling objective of meeting customer needs at the lowest cost.[6]

Limitations of the classic frameworks. Despite the general superiority of the K-framework over the D-framework, both classic frameworks have limitations that curtail their effectiveness, as I have stated above. Both the D-framework and the K-framework suffer from what might be called a "single-period paradigm." A scheduling framework based on a single-period paradigm uses employee requirements that are set independently for each planning period. That is, the framework doesn't account for the fact that employees work discrete shifts that start and end at particular times. That is a problem, of course, because employee shifts affect a number of successive planning periods.

From a service-standard point of view, a single-period approach prohibits a schedule from delivering a lower-than-ideal, but acceptable, level of service in one period, even though doing so might better control labor costs and allow a generally better staffing fit in the other planning periods. Because of its built-in floor, the single-period paradigm makes it difficult for managers to determine employee-staffing levels that provide the exact level of service that managers seek. Another limitation of the classic frameworks in service-standard environments is their assumption that a surplus employee is of equal value no matter when the extra person is at work. In reality, however, adding an extra employee in a busy period could improve the service experienced by a great number of customers, while adding that extra employee in a slow period may have minimal benefit on customer service. An assumption of equality makes it impossible for either framework to maximize the level of service provided if the manager assumes either a fixed labor cost or fixed labor hours.

Similar limitations also exist for the D- and K-framework in an economic-standard environment.[7] The classic frameworks' use of a single-period paradigm and their assumption that surplus employees are of equal value regardless of the period in which the surplus occurs mean that those frameworks cannot

[6] For example, see: Gary Thompson, "Representing Employee Requirements in Labor-Tour Scheduling," *Omega*, Vol. 21, No. 6 (1993), pp. 657–671; and Gary Thompson, "Labor Staffing and Scheduling Models for Controlling Service Levels," *Naval Research Logistics*, Vol. 44 (1997), pp. 719–740.

[7] The differences between a service standard and an economic standard are discussed in the second article in this series. The service standard calls for a schedule that delivers a targeted level of service, while the economic standard calls for a schedule that is the most economical. See: Thompson (December 1998), p. 27.

provide the best schedule from an economic perspective.

Two new frameworks. In the past four years I have presented two new conceptual frameworks for workforce scheduling that are intended to remedy the shortcomings of the classic frameworks. The two new frameworks are variations on the same theme. One of them, the contemporary service framework (CS-framework), overcomes the limitations of the classical frameworks in service-standard environments, while the other, the contemporary economic framework (CE-framework), does the same in economic-standard environments.

The CE-framework's objective, for example, is to maximize the total economic profit from a given schedule, subject to the following provisos. For each period, the framework must **(1)** ensure that a minimum acceptable number of employees are scheduled and **(2)** measure the number of additional employees that would provide an improvement in economic performance. Note that the minimum acceptable number of employees in a period is typically less than the ideal number of employees for that period—representing the difference between getting by and doing a complete job. Thus, after setting the minimum number of employees, the CE-framework measures the benefit of increasing the number of employees scheduled over the minimum and returns a solution that best balances the monetary benefit of good service, the monetary cost of poor service, and the cost of delivering the service. The CS-framework takes a similar approach using service standards as its specification.

The chief difference between my contemporary conceptualization and the classic frameworks is the treatment of steps two and three (i.e., determining staffing requirements and creating a schedule). The D- and K-frameworks first calculate the ideal employee requirements (in step 2) and then use that calculation as essentially an independent input for drawing up the actual schedule (in step 3). In contrast, the CE- and CS-frameworks more closely link scheduling tasks two and three by accounting for specific employee requirements in both the determination of the number and actual scheduling of workers. The CS- and CE-frameworks do this by recognizing that one cannot independently determine the ideal employee requirements because of the interdependence of staffing decisions across periods. The CS- and CE-frameworks thus exhibit what might be called a "cross-period paradigm." My past experiments have shown that the CS- and CE-frameworks yield better schedules than the classic frameworks.[8] Moreover, those experiments also showed that the CS- and CE-frameworks perform better because they make more appropriate use of the information available to the manager, and not because they require more or better information. In my experience, currently available workforce-scheduling systems apply the single-period paradigms or, at best, a rudimentary cross-period paradigm. As I explain in the remainder of this article, managers can improve on that scheduling rubric.

A goal of scheduling frameworks should be to measure the actual benefit (or not) of adding more employees over the bare minimum needed.

Developing Workforce Schedules

In developing a schedule, a manager can use either a two-phase or a one-phase procedure. The two-phase approach entails the classic single-period paradigm. In the first phase of the two-phase approach, the manager develops a workforce schedule without regard to em-

[8] For example, see: Thompson (1997), pp. 719–740; and Gary Thompson, "Labor Scheduling Using NPV Estimates of the Marginal Benefit of Additional Labor Capacity," *Journal of Operations Management*, Vol. 13 (1995), pp. 67–86.

A Comparison of Single-period and Cross-period Paradigms

This example is based on a scenario presented in the second article in this series.* The assumptions are these: the prime labor driver is the number of customers; each employee can serve a maximum of 16 customers per hour; the labor rate is $10.00 per hour, including benefits; the cost of customer waiting is estimated at $13.46 per hour; and employees work only three-hour shifts. Table A gives the forecasted customer-arrival rates for the five-hour scheduling horizon. The minimum acceptable staffing level in that table is equal to the smallest number of employees who can serve customers at a faster rate than they arrive (thus, the queue size does not increase at that minimum level). For example, 29.4 customers are forecast to arrive in hour 4, necessitating at least two employees. Table A also shows the incremental reduction in customer-waiting costs for each employee added above the minimum acceptable level. For example, 33 customers are forecast to arrive in hour 1, yielding a minimum acceptable staff size of three employees. If the manager increases the number of employees scheduled in hour 1 to four, the organization benefits by reducing the estimated wait-related cost by $11.33.

Applying the single-period paradigm described in the accompanying article, the ideal staffing level for each period is the last employee whose marginal benefit exceeds his marginal cost for the period. Thus, the ideal staffing levels are one employee higher than the minimum for each period.

When one solves the D-framework representation of this problem, the optimal solution is to schedule eight shifts: four starting in period 1 (working hours 1, 2, and 3), the remainder starting in hour 3 (working hours 3, 4, and 5). This gives a total of four employees working in each hour except for hour 3, where eight people are on duty because the two sets of four employees overlap (Table B). The net benefit of this schedule can be calculated from adding the incremental reductions in customer waiting costs and subtracting the cost of the shifts. That schedule means that hours 1 and 5 have what is calculated to be the ideal

*Gary M. Thompson, "Labor Scheduling, Part 2: Knowing How Many On-duty Employees to Schedule," *Cornell Hotel and Restaurant Administration Quarterly*, Vol. 39, No. 6 (December 1998), pp. 26–37.

Table A

Expected marginal reduction in hourly customer-waiting costs with increases in the number of employees

Shift hour	Arrival rate (customers)	Minimum staffing level	Ideal staffing level	Marginal reduction in customer-waiting cost ($) with added employees					
				+1	+2	+3	+4	+5	+6
1	33.0	3	4	$11.33	$2.09	$0.48	$0.11	$0.03	$0.01
2	10.5	1	2	15.80	0.95	0.10	0.01	0.00	0.00
3	13.8	1	2	70.19	2.31	0.30	0.04	0.01	0.00
4	29.4	2	3	126.05	6.33	1.22	0.27	0.06	0.01
5	42.3	3	4	68.64	7.34	1.74	0.47	0.13	0.03

Note: The minimum staffing level is set so that employees can at least stay ahead of customer arrivals. The marginal cost reduction of adding an employee is based on a complicated calculation of the cost of customer-waiting time. For a discussion of that calculation, see: Gary M. Thompson, "Labor Scheduling, Part 2: Knowing How Many On-duty Employees to Schedule," *Cornell Hotel and Restaurant Administration Quarterly*, Vol. 39, No. 6 (December 1998), pp. 32–34.

Table B

Incremental reductions in customer-waiting cost ($) captured by the optimal D-framework schedule

Shift hour	Incremental staffing level (number of employees above minimum staff)							
	+1	+2	+3	+4	+5	+6	+7	
1	$11.33							**Incremental**
2	15.80	$0.95	$0.10					**savings**
3	70.19	2.31	0.30	$0.04	$0.01	$0.00	$0.00	**in**
4	126.05	6.33						**waiting**
5	68.64							**time**

Table C

Incremental reductions in customer-waiting cost ($) captured by the optimal CE-framework schedule

Shift hour	Incremental staffing level (number of employees above minimum staff)							
	+1	+2	+3	+4	+5	+6	+7	
1								**Incremental**
2	$15.80	$0.95						**savings**
3	70.19	2.31	$0.30	$0.04	$0.01	$0.00		**in**
4	126.05	6.33						**waiting**
5	68.64							**time**

number of employees, four. Hour 2 has two employees above the ideal and three above the minimum, while hour 4 has one above the ideal and two above the minimum. The only serious overstaffing is in hour 3, when the marginal employees are adding no value at all to the customer-waiting-time calculation (as shown in the +6 and +7 cells of Table B). Still, the net value of this schedule to the company is positive. It costs the firm only $240 (in labor) to reduce customer-waiting costs by $302.05, for a net benefit of $62.05.

Most people adept at scheduling will see that the above solution is not the most economically wise approach, although it certainly meets all customer-service targets. The optimal solution to the CE-framework representation of the problem is to schedule only seven shifts: three commencing in hour one (covering hours 1, 2, and 3), the remaining four commencing in hour three (covering hours 3, 4, and 5). The staffing level for the first two hours is thus three employees, which meets the minimum acceptable service standard, but not the ideal. In subsequent hours, the unit exceeds its ideal standard by as many as five employees (hour 3). Table C provides the incremental reductions in customer-waiting cost captured by this schedule. The total incremental reduction in customer-waiting cost is $290.62; since the seven shifts cost a total of $210.00 (in labor), the net benefit of the CE-framework schedule is $80.62.

A careful comparison of the D- and CE-framework schedules shows the CE-framework's superiority. Consider what happens as one adds an additional shift starting in hour 1 to the CE-framework schedule. Given the existing staffing levels, one reduces the incremental cost of customer waiting by $11.33, $0.10, and $0.00 in hours 1, 2, and 3, respectively, for a total reduction in waiting costs of only $11.43. The additional shift, however, costs $30.00 (in labor), which eliminates any net benefit of adding the worker ($11.43 - 30.00 = -$18.57). In other words, the marginal cost for the eighth shift exceeds its marginal benefit. The D-framework, which uses a single-period paradigm, fails to recognize this and so adds the shift. Adding the incremental shift reduces the total benefit of the CE-framework from its current $80.62 to the $62.05 of the D-framework. The CE-framework is more effective because it better determines the interaction of shifts and judges each shift on its true incremental value.—*G.M.T.*

ployee availability or preferences. In the second phase, the manager assigns shifts to specific employees, if that is possible given the scheduling pattern proposed in phase one. Two-phase approaches are relatively quick and expedient to develop. However, the two-phase approach is generally inferior to a one-phase approach, which considers employee information in the process of developing the shift schedule. Two-phase approaches often yield poorer matches between the number of employees scheduled and the number of employees needed and, additionally, two-phase approaches may not always satisfy certain schedule requirements, such as making sure an employee works a minimum number of hours.

By considering the restrictions arising from employee requirements, the one-phase, or holistic, schedule overcomes the limitations of a two-phase approach. The problem with holistic approaches, however, is that they can be slow unless they are well designed. Although a number of commercial workforce-scheduling systems still use two-phase approaches, the one-phase approach is becoming more common due to the problems associated with two-phase approaches.

Here is a sample scheduling problem that illustrates the difficulty that can arise with a two-phase approach. Exhibit 1, on the next page, shows the employee requirements in each of eight hour-long planning periods. Assume that shifts must be at least four and no more than eight hours long. Finally, assume that two employees are available for work: Employee A is available from hour one to hour 7 while Employee B is available from hour 3 to hour 8. For simplicity of presentation, the shift-based and holistic approaches will be compared using the D-framework.

As Exhibit 1 shows (on the net-staffing-level line), the shift-based schedule perfectly matches the business's staffing needs in all periods. It has neither under- nor overstaffing. The shift-based schedule puts one employee on an eight-hour shift and has the second covering hours three through six. The problem with this schedule is obvious when one attempts to perform the second phase and assign employees for the specified shifts. Either employee could work Shift 2, but neither employee can work Shift 1. A shift-based approach requires the manager either to leave Shift 1 unassigned, which would result in substantial understaffing, or to modify Shift 1 so that one of the workers could cover at least a portion of the shift. However, any such modification would require the manager to go back to the first step to ensure that service standards are met. (In this simple example, one can see the solution at a glance, but a real schedule would be far more complex.)

It makes more sense to factor in employee availability from the start as is done in the example of the holistic approach. The holistic approach develops a schedule that also has no under- or overstaffing and has two shifts. Here the resemblance ends, however, because the first shift covers hours one through six (assigned to Employee A), and the second shift covers hours three through eight (assigned to Employee B). By considering employee availability while developing the schedule, the holistic approach prevents a scheduling mechanism from proposing shifts that cannot be staffed.

Solving workforce-scheduling problems. Managers can use any of many methods, or algorithms, for developing workforce schedules. These methods can be categorized

Heuristics

A heuristic is a logic-based procedure designed to yield good schedules quickly. Consider the task of trying to construct a workforce schedule. One must start with a blank slate—no shifts—and add shifts until there are enough. How should one select shifts to add to the schedule? In an earlier paper, I evaluated a set of 20 rules for constructing schedules.* Those 20 rules were based on five criteria, each of which was a primary criterion and each of which became a tie-breaker for any of the others. Those five criteria were to add the shift that: **(1)** covers the period having the greatest single-period staff shortage; **(2)** covers the highest average short-staffing; **(3)** offers the greatest reduction in schedule cost per working period; **(4)** offers the greatest improvement in schedule smoothness (where smoothness is measured as the absolute difference in net staffing levels from period to period); and **(5)** covers the periods having the highest average ratio of the number of employees still needed in a period divided by the number of still-unassigned employees who could work the period.

To see how two of those rules would work, consider the following example. A manager is considering adding one of two four-hour shifts to the schedule. The first shift covers periods where the net staffing levels are 0, -1, -2, and 0 (i.e., employees are still needed only in the second and third hours of the shift). The second shift covers periods where the net staffing levels are -1, -1, -1, and -1 (thus, employees are needed in every period). The criterion that adds the shift covering the maximum staff shortage would select the first of these shifts, while the criterion that adds the shift covering the greatest average short staffing would select the second of these shifts.—*G.M.T.*

*Gary Thompson, "A Simulated-Annealing Heuristic for Shift Scheduling Using Non-Continuously Available Employees," *Computers and Operations Research,* Vol. 23, No. 3 (1996), pp. 275–288.

Exhibit 1
A comparison of one- and two-phase solutions for a simple scheduling problem

Two-phase (shift-based) solution

Hour		1	2	3	4	5	6	7	8
Employees needed		1	1	2	2	2	2	1	1
Employee A availability		Y	Y	Y	Y	Y	Y	Y	N
Employee B availability		N	N	Y	Y	Y	Y	Y	Y
Shift-based schedule (two-phase)	Shift 1 (employee A)	W	W	W	W	W	W	W	W
	Shift 2 (employee B)			W	W	W	W		
	Total scheduled	1	1	2	2	2	2	1	1
	Net staffing level	0	0	0	0	0	0	0	0

One-phase (holistic) solution

Hour		1	2	3	4	5	6	7	8
Employees needed		1	1	2	2	2	2	1	1
Employee A availability		Y	Y	Y	Y	Y	Y	Y	N
Employee B availability		N	N	Y	Y	Y	Y	Y	Y
Holistic schedule (one-phase)	Shift 1 (employee A)	W	W	W	W	W	W		
	Shift 2 (employee B)			W	W	W	W	W	W
	Total scheduled	1	1	2	2	2	2	1	1
	Net staffing level	0	0	0	0	0	0	0	0

Forecasts of customer activity show that during the above eight-hour shift the establishment will need one employee on duty during the first two and last two hours. To handle a mid-shift rush, both employees must be on duty, as shown in the "employees needed" row. Scheduling without regard for employee preferences—as shown in the top, shift-based schedule—puts one employee on duty for eight hours and brings in the other only for the four-hour rush. (Assume that shifts must be at least four and no more than eight hours long and that only these two employees are available for work.) The schedule meets customer-service standards by balancing the number of employees needed with those scheduled, as indicated by the zeros in the "net staffing level" line. Clearly, neither employee would be satisfied with this schedule, however, and the manager would actually be unable to staff it. Employee A would be scheduled for a time that she or he cannot work, while Employee B has the prospect of only a four-hour paycheck. In contrast, the bottom, holistic schedule, accounts for the employees' availability and also balances out the paychecks for both, while still meeting the customer-service standards.

as optimal or heuristic. (For a discussion of heuristics, see the accompanying box.) An optimal procedure is one that guarantees to find the best possible schedule. Unfortunately, the scheduling situations in which one can find an optimal schedule are usually much less complex than those occurring in real hospitality firms. Even if optimal procedures are applied during both phases of a two-phase approach, the overall solution will usually not be optimal because a two-phase approach does not consider the entire scheduling problem at one time.

A heuristic is a procedure for obtaining a good schedule quickly. Heuristic procedures may find the optimal schedule, but have no means of verifying that the result is, in fact, optimal. Developing an effective heuristic is an art. One strives to develop a solution procedure that incorporates relevant information and that is effective across a broad range of scheduling scenarios. Developing optimal one-phase schedules is impractical at this point for most real scheduling situations because finding such solutions would be too time consuming.

Consequently, most commercial workforce-scheduling systems are heuristic-based.

Scheduling Considerations

This section discusses the considerations in schedule development: controllable and uncontrollable work, scheduling flexibility, employee issues, rigid and soft constraints, forced and voluntary overtime, and scheduling-horizon duration.

Controllable and uncontrollable work. As discussed in the first article in this series, uncontrollable work is work over which there is little temporal control—the serving of customers that must be done when the customers are in the service system, for example. Controllable work, by contrast, is work over which there is some degree of temporal control. Controllable work is useful from a scheduling perspective, simply because it is controllable. The ability to schedule the work, within limits, offers a form of flexibility.[9]

A manager should schedule controllable work and develop the workforce schedule simultaneously.[10] Doing so ensures that employees are kept busier than if one schedules the controllable work only to fill periods of idle time. Ironically, this may mean scheduling controllable work during busy periods—but that makes sense when one considers that busy times are also the times with the most employees at work. The example in Exhibit 2 shows a schedule for controllable and uncontrollable work in a five-hour scheduling horizon. The example assumes that employees are properly cross-trained to handle all the tasks at hand. The charts in Exhibit 2 begin by showing the ideal number of employees needed to perform the uncontrollable work. The top chart differs from the bottom chart in the way in which it handles the schedule for three hours of controllable work. True to its name, the work can all be done in any single hour, or it can be spread across any or all of the five hours on the schedule. The scheduling situation assumes that employee shifts are three hours long and, thus, can start only in hour 1, 2, or 3.

Two-phase approach. If one were to assign the controllable work prior to scheduling the employees, one would be tempted to assign the controllable work to slow and shoulder times—the periods where the volume of uncontrollable work is lowest. The result might be the top chart in Exhibit 2. Using a D-framework for simplicity, the best schedule comprises eight overlapping shifts. Four shifts commence in hour 1 and four commence in hour 3. The staff would be at maximum strength in hour 3, which is the busiest period. Ironically, since the eight employees overlap in hour 3, that is also the time when some

Exhibit 2
Comparison of methods for scheduling controllable work

Schedule with pre-assigned controllable work (D-framework solution)

Hour	1	2	3	4	5
Employees needed for uncontrollable work	4	3	5	3	3
Assigned controllable work (total three hours)		1		1	1
Total employee requirements	4	4	5	4	4
Shift 1	w	w	w		
Shift 2	w	w	w		
Shift 3	w	w	w		
Shift 4	w	w	w		
Shift 5			w	w	w
Shift 6			w	w	w
Shift 7			w	w	w
Shift 8			w	w	w
Total scheduled employees	4	4	8	4	4
Overstaffing	0	0	3	0	0

Schedule resulting when controllable work is matched to employee availability (D-framework solution)

Hour	1	2	3	4	5
Employees needed for uncontrollable work	4	3	5	3	3
Shift 1	w	w	w		
Shift 2	w	w	w		
Shift 3	w	w	w		
Shift 4	w	w	w		
Shift 5			w	w	w
Shift 6			w	w	w
Shift 7			w	w	w
Total scheduled employees	4	4	7	3	3
Scheduled controllable work		1	2		
Total workload	4	4	7	3	3
Overstaffing	0	0	0	0	0

The example assumes that employees are properly cross-trained to handle all the tasks at hand. The scheduling situation also assumes that employee shifts are three hours long and, thus, can start only in hour 1, 2, or 3.

[9] See: Thompson (October 1998), p. 23.

[10] As explained further in: Gary Thompson, "Improving the Utilization of Front-Line Service Delivery System Personnel," *Decision Sciences*, Vol. 23, No. 5 (September/October 1992), pp. 1072–1098.

Selecting a Workforce-scheduling System

Hospitality firms have many choices for workforce-scheduling systems. Windows-based workforce-scheduling software can be purchased for as little as several hundred dollars, or firms can invest in customized systems costing hundreds of thousands of dollars. Good scheduling systems have the following characteristics, in my opinion.

An intuitive graphical interface. The interface should facilitate tinkering with or editing a schedule. Although good systems will generate schedules that require a minimum of adjustment, systems rarely incorporate all relevant factors, and managers usually must make changes.

A good scheduling engine. The scheduling engine—the algorithms that actually develop the schedule—is harder to evaluate than the graphical interface because one cannot see the engine. A good scheduling engine will incorporate effective logic. It should use a cross-period paradigm—considering the interactions among planning periods—rather than a single-period paradigm, and it should be holistic instead of shift-based to take into account employee characteristics as shifts are developed. The engine should operate quickly and provide schedules that need a minimal amount of tinkering. The best way to evaluate the engines is by giving them test problems. One can compare the speed and ability of different systems to schedule the right numbers of employees at the right times (i.e., avoiding overstaffing and understaffing) and to satisfy employees' schedule requests.

Employee preferences. Speaking of schedule requests, a good scheduling system will require that employees identify their work preferences in advance, including ranking those preferences or identifying trade-offs among their preferences. The best systems will also use this information when developing the schedule and specifically will exploit complementary preferences when they exist. By explicitly considering each employee's preferences, a good scheduling system will deliver the best possible schedule in terms of matching the number of employees scheduled to the ideal number of employees needed while at the same time satisfying employee preferences as much as possible.

Constraints. Hospitality businesses should be aware of their particular rigid and soft constraints. This will allow managers to evaluate the degree of congruence between a scheduling system's rigid and soft constraints and those identified by the hospitality firm. How a scheduling system deals with conflicting constraints can make the difference between an average and a good system. Once again, the system's schedule should not require much managerial tinkering to get it to a usable form.

Flexibility. Good scheduling systems allow a high degree of scheduling flexibility, since flexibility yields better schedules. Good systems should allow various times for controllable work, multiple shift lengths, and a variety of break times. The system should also allow employees to be cross-scheduled within and across shifts. As noted earlier, however, high flexibility greatly increases the complexity of a scheduling scenario and consequently may noticeably lengthen the time a system requires to develop a schedule.

Optional overtime. The best system schedules optional overtime only at the times employees can work it and for those employees who actually want to work it. That routine will typically provide better actual schedules (and happier employees) than one that uses generic shifts.

Costs and benefits. Finally, when evaluating workforce-scheduling systems, one should not neglect the costs and benefits. Inexpensive, off-the-shelf systems will generally require that you fit your business operations to the system. In contrast, a slightly more expensive, customized scheduling solution will fit the system to your business. Your firm's resources and the potential benefits of a system will determine the best approach.—*G.M.T.*

employees are idle. Given that only five workers are actually needed, this schedule shows three hours of idle time, all occurring in hour 3.

One-phase heuristic. In contrast, if the manager considers when the controllable work can be done as part of the employee schedule, the result is strikingly different. Looking at the lower chart in Exhibit 2, one can see that moving the controllable work away from the slow times to the times when employees are already at work makes possible a schedule with seven shifts instead of eight. Again using the D-framework, the schedule sets four shifts that commence in hour 1 and three shifts starting in hour 3. One hour of controllable work is assigned to hour 2 (when four employees are handling three hours' worth of uncontrollable work) and two hours are assigned to hour 3. Even though hour 3 is the peak time for uncontrollable work, the staff is also at peak due to the overlapping shifts. This schedule has no scheduled idle time. The total of 21 hours of scheduled time equals the total required workload.

Comparing the two schedules, one sees the advantage of flexibility in scheduling controllable work. By considering the total picture in scheduling controllable work, a manager can minimize both staff shortages and surpluses. Such flexibility can allow one to better achieve the primary economic and service-based scheduling objectives.

Other flexibility. If flexibility's boon is that it allows the manager to match the number of employees scheduled to the ideal number needed, the bane of flexibility is that it can greatly increase the scheduling complexity and slow the work of casting a schedule. In addition to arranging controllable work, flexibility comes from variable shift

lengths (including overtime), break timing, alternate start times, and using employees in different jobs within and across shifts.

Employee considerations. Employee characteristics can also be considered during schedule development. One group of characteristics, known as environmental considerations, are generally outside employees' control (at least over the duration of the scheduling horizon). Environmental characteristics include employees' skills, consideration of days off (e.g., a requirement for consecutive days off), restrictions on the minimum and maximum daily and weekly hours, and the times at which employees are not available for work.

The other group of employee characteristics, preferential considerations, can be altered. Those characteristics include preferences regarding the total daily and weekly work hours, days off, task assignments, and work position (e.g., stations in a full-service restaurant). Employees may also express preferences on times when they could but would rather not work, when they ideally would like to start and finish work, and the length of meal break they would like.

Needless to say both the environmental and preferential characteristics can vary from employee to employee. For example, the schedule focus for one employee might be to make sure she gets her desired total work hours, while another employee's focus might be to get the work station he most prefers. One should not view employee preferences as reducing management's ability to deliver service. Since employees rarely have similar schedule-preference characteristics, a manager should be able to find complementary differences that will help in schedule development.

Rigid and soft constraints. Employees' environmental considerations are part of the set of constraints that managers face in schedule development. Constraints of all kinds limit one's ability to develop an optimum schedule either from the organization's perspective or the employees' point of view. Rigid constraints are those that must be satisfied in a schedule, if at all possible. Soft constraints, on the other hand, are those that should be satisfied in a schedule, if possible. The problem with those simple definitions is that constraints can conflict with each other, and a manager sometimes must determine which rigid constraint must be violated after all. Rigid constraints should not be violated (unless by another rigid constraint) while soft constraints can be violated. Typical rigid constraints include restrictions that define the minimum and maximum acceptable daily and weekly hours for employees; ensuring the employees are assigned to a job they can actually perform; and scheduling employees only when they are actually available. Soft constraints, on the other hand, would include giving employees their ideal daily and weekly hours.

Considering the clash of constraints, it may be helpful to think of constraints as a hierarchy. High-ranking constraints (notably, rigid constraints) would take precedence over constraints lower in the hierarchy. It is also often useful to place small-scope constraints higher in the hierarchy than the large-scope constraints. In my estimation, a limit on the number of hours worked in a day is a small-scope constraint, while weekly hours is a large-scope constraint. Let's say that due to staffing considerations we need a certain employee to work 38 hours in a week. This employee is limited to

By considering the total picture in scheduling controllable work, a manager can minimize both staff shortages and surpluses.

seven hours per day and is available only five days in a week. Under the hierarchy I propose, this employee would be scheduled for only 35 hours for the week, observing the limitations on daily hours and number of days worked, while violating the larger-scope weekly constraint of 38 hours.

The reason the limits on daily and weekly work hours should be rigid constraints is that firms often have contractual or obligatory relationships with employees that define the amount of work hours the employees should receive. For example, the constraint on minimum hours is of even higher priority. I recently had the opportunity to view a hospitality purveyor's scheduling system that failed to treat employees' minimum weekly hours as a rigid constraint. Thus, other considerations would take precedence, and employees might not be scheduled for their agreed-upon minimum weekly work hours. Managers using this system would have to spend a considerable amount of time tinkering with the schedule to ensure employees were getting their specified weekly minimum hours.

Forced and voluntary overtime. Hospitality businesses often operate in high-demand situations in which the on-duty employees cannot cover the required workload during their regular hours—and the manager must schedule overtime. In such cases, the manager may schedule forced overtime. In that instance, employees would be assigned longer daily shifts or an additional work day. On the other hand, the manager might allow employees to sign up for overtime, thus creating voluntary overtime hours or shifts.

From a scheduling point of view, forcing overtime is straightforward (providing that one's scheduling system allows it). The manager simply lengthens shifts or adds work days. Scheduling voluntary overtime, though, is less straightforward and so merits further consideration.

One approach to handling voluntary overtime is to develop unfilled, generic shifts that are not assigned to a particular employee. The manager can post these unfilled shifts and allow employees to sign up for those shifts that they are willing to work. An obvious difficulty with that method is that the employees who sign up may not be a fit with the skills required for the particular shift.

A more coherent approach is for the manager to find out in advance which employees would be willing to work overtime. The manager can use this information to develop the schedule and to assign the overtime to specific employees—based on whether they would be willing to pick up an extra shift, work longer shifts, or both.

The second alternative for handling voluntary overtime is also preferable to involuntary assignment because it heads off the problem inherent in a two-phase scheduling process. The proposed, generic schedule may bear no resemblance to the actual schedule because employees may pick up only some of the generic shifts. Even if the original schedule theoretically delivers employees when needed, there is no guarantee that the generic shifts will be scheduled at times when employees can actually work. If the schedule is optimized as it stands (including the generic shifts), then a failure to fill any generic shift means the resulting schedule will fail to meet its customer-service or economic objectives.

Scheduling horizon. Most hospitality managers set their schedules for one to two weeks in advance, though sometimes the schedules are developed only quarterly. Despite the obvious difficulties in forecasting customer-service needs more than two weeks in advance, I have seen a hospitality firm get trapped with a long scheduling horizon. This firm used a two-phase approach to developing its workforce schedule. As part of the second phase (matching its employees to shifts), the company allowed employees to select their schedules in an order based on seniority and productivity. This process was so cumbersome that the firm could justify performing it only once per quarter. To reduce the scheduling horizon, the firm could use a one-phase approach that incorporated employee preferences as part of the scheduling process.

Integrated Approach

To summarize my recommendations presented in this paper, hospitality firms should be implementing one-phase-solution procedures using a cross-period paradigm. Satisfying employee preferences for work schedules is not inconsistent with delivering the service efficiently. The preferences need only be incorporated into an effective heuristic procedure that matches complementary employee preferences. Schedules should be developed for no more than one or two weeks at a time.

The tasks described in this paper and the two previous papers in the series have focused on planning: the process by which a schedule is developed for the demand that is anticipated. In the next paper of this series, I focus on the final task in workforce scheduling—controlling the schedule in real time. Real-time control ensures that the actual schedule worked by employees meets the expectations that drove the schedule as planned. **CQ**