Management Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Business Analytics for Flexible Resource Allocation Under Random Emergencies

Mallik Angalakudati, Siddharth Balwani, Jorge Calzada, Bikram Chatterjee, Georgia Perakis, Nicolas Raad, Joline Uichanco

MANAGEMENT SCIENCE
Vol. 60, No. 6, June 2014, pp. 1552–1573
ISSN 0025-1909 (print) | ISSN 1526-5501 (online)

http://dx.doi.org/10.1287/mnsc.2014.1919

Business Analytics for Flexible Resource Allocation Under Random Emergencies

Mallik Angalakudati
Pacific Gas and Electric Company, San Ramon, California 94583, m3an@pge.com

Siddharth Balwani
BloomReach, Mountain View, California 94041; and Leaders for Global Operations, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, sid@bloomreach.com

Jorge Calzada
National Grid, Waltham, Massachusetts 02451, jorge.calzada@nationalgrid.com

Bikram Chatterjee
Pacific Gas and Electric Company, San Ramon, California 94583, b2cl@pge.com

Georgia Perakis
Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, georgiap@mit.edu

Nicolas Raad
National Grid, Waltham, Massachusetts 02451, nicolas.raad@nationalgrid.com

Joline Uichanco
Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, uichanco@alum.mit.edu

In this paper, we describe both applied and analytical work in collaboration with a large multistate gas utility. The project addressed a major operational resource allocation challenge that is typical to the industry. We study the resource allocation problem in which some of the tasks are scheduled and known in advance, and some are unpredictable and have to be addressed as they appear. The utility has maintenance crews that perform both standard jobs (each must be done before a specified deadline) as well as respond to emergency gas leaks (that occur randomly throughout the day and could disrupt the schedule and lead to significant overtime). The goal is to perform all the standard jobs by their respective deadlines, to address all emergency jobs in a timely manner, and to minimize maintenance crew overtime. We employ a novel decomposition approach that solves the problem in two phases. The first is a job scheduling phase, where standard jobs are scheduled over a time horizon. The second is a crew assignment phase, which solves a stochastic mixed integer program to assign jobs to maintenance crews under a stochastic number of future emergencies. For the first phase, we propose a heuristic based on the rounding of a linear programming relaxation formulation and prove an analytical worst-case performance guarantee. For the second phase, we propose an algorithm for assigning crews that is motivated by the structure of an optimal solution. We used our models and heuristics to develop a decision support tool that is being piloted in one of the utility's sites. Using the utility's data, we project that the tool will result in a 55% reduction in overtime hours.

Keywords: resource allocation; stochastic emergencies; scheduling; gas pipeline maintenance; utility; optimization
History: Received September 23, 2012; accepted January 13, 2014, by Noah Gans, special issue on business analytics. Published online in *Articles in Advance* April 3, 2014.

1. Introduction

Allocating limited resources to a set of tasks is a problem encountered in many industries. It has applications in project management, bandwidth allocation, Internet packet routing, job shop scheduling, hospital scheduling, aircraft maintenance, air traffic management, and shipping scheduling. In the past decades, the focus has been primarily on developing methods for optimal scheduling for deterministic problems. These approaches assume that all relevant information is available before the schedule is decided and the parameters do not change after the schedule is made. In many realistic settings, however, scheduling decisions have to be made in the face of uncertainty. After deciding on a schedule, a resource may unexpectedly become unavailable, a task may take longer or shorter time than expected, or there might be an unexpected release of high-priority jobs (see Pinedo 2002 for an overview of stochastic scheduling models). Not accounting for these uncertainties may cause an undesirable impact, say, in a possible schedule interruption or in overutilizing some resources. Birge (1997) demonstrated that in many real-world applications, when using stochastic optimization to model uncertainties explicitly, the results are superior compared to using a deterministic counterpart.

In this paper, we study the problem of scheduling a known set of jobs when *there is an uncertain number of emergency jobs that may arrive in the future*. There are many interesting applications for this type of problem. For instance, Lamiri et al. (2008) describe the problem of scheduling surgeries in hospital intensive care units, where operating rooms are shared by two classes of patients: elective patients and emergency patients. Emergency cases arrive randomly but must be served immediately upon arrival. Elective cases can be delayed and scheduled for future dates. In scheduling the elective surgeries, the hospital needs to plan for flexibility (say, by having operating rooms on standby) to handle random arrivals of emergency cases.

This paper is motivated by a project with a major electric and gas utility. We worked on improving scheduling of services for the utility's gas business segment, which faces uncertainty in its daily operations. The gas business segment of the utility generates several billion dollars in revenue annually. The utility operates a distribution system (large networks of gas pipelines in several U.S. states) that delivers natural gas to its customers. A major part of daily operations of the gas utility is the maintenance of the pipeline networks. This entails executing two types of jobs: (i) *standard jobs* and (ii) *emergency gas leak repair jobs*. The first type of jobs includes new gas pipeline construction, maintenance and replacement of gas pipelines, and customer requests. The key characteristics of standard jobs are that they have deadlines by when they must be finished, they are known several weeks to a few months in advance of their deadlines, and they are often mandated by regulatory authorities or required by customers. The second type of job is to respond to reports of gas leaks. In the United States, more than 60% of the gas transmission pipes are at least 40 years old (Burke 2010). Most of them are composed of corrosive steel or cast iron. Gas leaks are likely to occur on corroding bare steel or aging cast iron pipes and pose a safety hazard especially if they occur near a populated location. If undetected, a gas leak might lead to a fire or an explosion. Such was the case in San Bruno, California, in September 2010, where a corrosive pipe ruptured, causing a massive blast and fire that killed eight people and destroyed 38 homes in the San Francisco suburb (Pipeline and Hazardous Materials Safety Administration 2011). To reduce the risk of such accidents occurring, company crews regularly monitor leak prone pipes to identify any leaks that need immediate attention. In addition, the company maintains an emergency hotline for reports of suspected gas leaks. It is the company's policy to respond to a report within 24 hours of receiving it. The key characteristics of emergency gas leak jobs are that they are unpredictable, they need to be responded to immediately, they require several hours to complete, and they happen with frequency throughout a day. The leaks that do not pose significant risk to the public are fixed later within regulatory deadlines dictated by the risk involved. These jobs are part of the standard jobs.

The company keeps a roster of *maintenance crews* to execute standard jobs and to respond to emergencies. The company has experienced significant crew overtime driven by both controllable factors (such as workforce management, scheduling processes, and information systems) and uncontrollable factors (such as uncertainty related to emergency leaks, diverse and unknown site conditions, and uncertainty in job complexity). Maintenance crews historically worked a significant proportion of their hours on overtime. From our analysis, one of the major causes of overtime is suboptimal job scheduling and planning for the occurrence of emergencies. Currently, the company has no standard procedures or does not use quantitative methods for job scheduling and crew assignment. Past studies undertaken by the company suggested that a better daily scheduling process that optimizes daily resource allocation can provide a significant opportunity for achieving lower costs and better deadline compliance.

In this paper, we study the utility's resource allocation problem along with associated process and managerial factors. However, the models proposed and insights gained from this paper may have wider applicability in settings where resources have to be allocated under stochastic emergencies.

1.1. Literature Review and Our Contributions

Our work makes theoretical contributions in several key areas, as well as contributions to the utility's operations. We contrast our contributions with previous work found in the literature.

Modeling and Problem Decomposition. We develop a multiperiod model for the utility's operations under stochastic emergencies. Before knowing the number of emergencies, the utility has to decide a standard job's schedule (which date it will be worked on) and its crew assignment (the crew assigned to it). We model the problem as a stochastic mixed integer program (MIP).

Several practical limitations discussed later in §3.1 (such as computational intractability, the utility's restrictive computing resources, and employees' wariness of a different decision-making process) prevented the utility from implementing the multiperiod stochastic MIP model. Therefore, we propose a *two-phase* *decomposition*, which addresses the original model's limitations. The first phase is a *job scheduling phase*, where standard jobs are scheduled so as to meet all the deadlines but while evenly distributing work over all days (§4). This scheduling phase solves a deterministic MIP. The second phase is a *crew assignment phase*, which takes the standard jobs scheduled for each day from the first phase and assigns them to the available crews (§5). Since the job schedules are fixed, the assignment decisions on different days can be made independently. The assignment decisions must be made before arrivals of emergencies; hence, the assignment problem on each day is solved as a two-stage stochastic MIP.

LP-Based Heuristic for the Scheduling Phase. We propose a heuristic for the NP-hard job scheduling problem based on solving its linear programming (LP) relaxation and rounding the solution to a feasible schedule. The scheduling phase problem is equivalent to scheduling jobs on unrelated machines with the objective of minimizing makespan (Pinedo 2002). In our problem, the dates are the "machines." The makespan is the maximum number of hours scheduled on any day. Note that a job can only be "processed" on dates before the deadline (the job's "processing set"). Scheduling problems with processing set restrictions are known to be NP-hard.

Other popular heuristics in the literature are list scheduling rules (Kafura and Shen 1977, Hwang et al. 2004). However these are applicable for problems with parallel machines. For unrelated machines, a well-known algorithm by Lenstra et al. (1990) performs a binary search procedure, in each iteration solving the LP relaxation of an integer program and then rounding the solution to a feasible schedule. Our algorithm is also based on solving an LP relaxation, but it applies for unrelated machines with processing set restrictions. Furthermore, it does not require initializing the algorithm with a binary search and therefore only solves an LP once. Since this heuristic is based on linear programming, in practice it solves very quickly with commercial off-the-shelf solvers.

Performance Guarantee for the LP-Based Heuristic. We prove a data-dependent performance guarantee for the proposed LP-based heuristic (Theorem 1). Lenstra et al. (1990) prove that the schedule resulting from their LP-based algorithm is guaranteed to have a makespan of no more than twice the optimal makespan. Their proof relies on graph theory. On the other hand, the bound we derive uses a novel technique based on stochastic analysis. Moreover, when the algorithm is initialized with a binary search, we can prove, using graph theoretic and stochastic arguments, a performance guarantee that is the minimum of two and a data-driven factor (Theorem EC.1 in the electronic companion). Since with real utility data, the data-driven factor is less than two, we improve upon the bound by Lenstra et al. (1990) in realistic settings.

Algorithm for Crew Assignment Under a Stochastic Number of Emergencies. The assignment phase problem is a two-stage stochastic MIP. In the first stage the assignment of standard jobs to crews is determined, and in the second stage (after the number of emergencies is known) the assignment of emergencies to crews is decided. Most literature on problems of this type develop iterative methods to solve the problem. For instance, a common method is based on Benders' decomposition embedded in a branch-and-cut procedure (Laporte and Louveaux 1993). However, if the second stage has integer variables, the second-stage value function is discontinuous and nonconvex, and optimality cuts for Benders' decomposition cannot be generated from the dual. Sherali and Fraticelli (2002) propose introducing optimality cuts through a sequential convexification of the second-stage problem. There are other methods proposed to solve stochastic models of scheduling under uncertainty. For instance, Lamiri et al. (2008) introduce a local search method to plan for elective surgeries in the operating room scheduling problem. Godfrey and Powell (2002) introduce a method for dynamic resource allocation based on nonlinear functional approximations of the second-stage value function based on sample gradient information. However, since they are developed for general two-stage stochastic problems, these solution methods do not give insights on how resources should be allocated in anticipation of an uncertain number of emergencies.

We exploit the structure of the problem and of the optimal assignment and propose a simple and intuitive algorithm for assigning the standard jobs under a stochastic number of emergencies (Algorithm Stoch-LPT). This algorithm can be thought of as a generalization of the Longest-Processing-Time First (LPT) algorithm in the scheduling literature (Pinedo 2002). We prove that this algorithm terminates with an optimal crew assignment for some special cases.

Models and Heuristics for Resource Allocation with Random Emergencies. Our paper is motivated by the specific problem of a gas distribution company. However, the models and algorithms we develop in this paper may have wider applicability to other settings where resources need to be allocated in a flexible manner to be able to handle random future emergencies. As a specific example, in the operating room planning problem described in the introduction, the resources to be allocated are operating rooms. Elective surgeries and emergency surgeries are equivalent to standard jobs and gas leak repair jobs, respectively, in our problem.

Business Analytics for a Large U.S. Utility. We collaborated with a large multistate utility on improving the scheduling of operations in its gas business segment.

The job scheduling and crew assignment optimization models described above are motivated by the company's resource allocation problem under randomly occurring emergencies. The job scheduling heuristic and the crew assignment heuristics described earlier are motivated from practical requirements, including the company's need for fast solution methods. We developed a Web-based planning tool based on these heuristics that is being piloted in one of the company's sites.

We also used our models to help the utility make strategic decisions about its operations. In simulations using actual data and our models, we highlight how different process changes impact crew utilization and overtime labor costs. In this paper, we analyzed three process changes: (i) maintaining an optimal inventory of jobs ready to be scheduled, (ii) having detailed crew productivity information, and (iii) increasing crew supervisor presence in the field. We demonstrate the financial impact of these new business processes on a hypothetical utility company.

1.2. Outline

In §2, we give a background of the gas utility and its operations. In §3, we present the job scheduling and crew assignment problem, and motivate the two-stage decomposition. In §4, we discuss the job scheduling phase, introduce an LP-based heuristic, and develop a data-driven performance guarantee of the heuristic. In §5, we discuss the crew assignment phase. In this section, we prove structural properties of the optimal solution, propose a crew assignment heuristic, and show that it terminates with an optimal solution for some cases. In §6, we discuss the development of the planning tool, the pilot project, and how we used simulation and the models we developed for business analytics at the gas business of a large multistate utility company. Proofs not shown in the paper can be found in the electronic companion (available at http://ssrn.com/abstract=2412239).

2. Gas Utility Operations and Background

In this section, we give a background of the utility's operations and organization. The discussion serves to motivate the model, assumptions, and our choice of heuristics later in the paper.

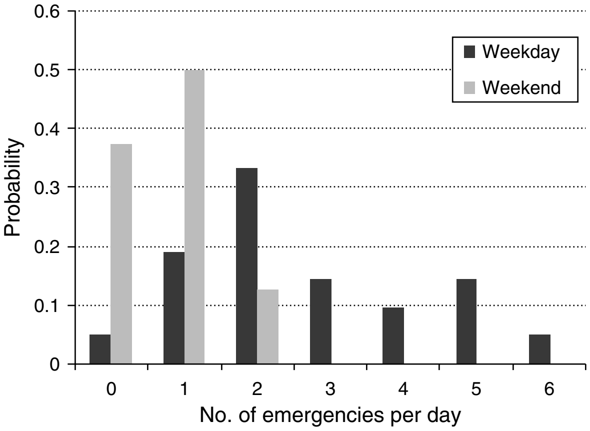
The gas utility operates and maintains a large network of gas pipes. In the United States, the natural gas industry is heavily regulated. Regulatory authorities mandate gas utilities to replace their cast iron pipes into more durable steel or PVC pipes through a main replacement program. To meet the guidelines of this program, the utility has a dedicated department called the *Resource Management Department* that sets yearly targets for standard jobs to be performed in the field and monitors the progress relative to these targets throughout the year. All targets are yearly and company-wide.

Standard jobs that occur within a geographical region (usually a town or several neighboring towns) are assigned to a *yard*. A yard is the physical company site that houses maintenance crews who are dispatched to complete the standard jobs. After the Resource Management Department decides on a company-wide target for standard jobs, it is translated into monthly targets for each yard based on yard size, number of workers available, and other characteristics of the region the yard serves. Several years ago, the utility expanded in the United States from a string of mergers of small independent local utilities operating in towns. As a result, even today, separate yards belonging to the company operate independently. Small yards can have 10 crews, and large yards can have up to 30 crews, with each crew composed of two or three crew members. Each standard job has a deadline set by the Resource Management Department to ensure that the targets are met and the company does not incur heavy regulatory fines for not meeting the requirements of the main replacement program. The company maintains a centralized database of standard jobs that lists each job's deadline, status (e.g., completed, pending or in progress); location, job type, and other key job characteristics; and also information on all past jobs completed. A large yard can complete close to 500 standard jobs in one month. The focus of the project and hence of this paper is on yard-level operations, which we describe below.

2.1. Daily Yard Operations

Each yard has a *resource planner* who is charged with making decisions about the yard's daily operations. At the start of each day, the resource planner reviews the pending standard jobs and their upcoming deadlines and decides which jobs should be done by the yard that day. The resource planner also determines which crews should execute these jobs. Shortly after, the maintenance crews are dispatched to their first assignments. Throughout the day, the yard might receive reports of emergency gas leaks that also need to be handled by maintenance crews. These gas leaks are found by company crews (operated by a department independent from the yards) dedicated solely to monitoring leak prone pipes to identify any leaks that need immediate attention. Leaks found that do not pose significant risk to the public are fixed later within regulatory deadlines (usually within 12 months) dictated by the risk involved. These less severe leaks are categorized as standard jobs. Emergencies are highly unpredictable; a given yard can have between zero to six emergencies per day (Figure 1). They also are

Figure 1 **Historical Distribution of the Number of Emergencies in a Given Yard for April 2011**



Note. Since most emergencies are found by monitoring, there are often more emergencies discovered during weekdays when more monitoring crews are working.

long duration jobs since, for regulatory compliance, the utility requires its crews to dedicate eight hours for responding to emergencies.

Once an emergency is reported to a yard, any idle crew is immediately dispatched to it. If there are no idle crews, the first crew to become idle after finishing its current job is immediately dispatched to the emergency. Once started, a standard job will not be paused even when an emergency arrives because of the significant startup effort for the job. Startup activities include travel to the site, excavating the street to access the pipe, and arranging mandatory police presence at the site. This established procedure for handling emergencies complies with the regulatory guideline that any report of an emergency has to be responded to within 24 hours of receiving it.

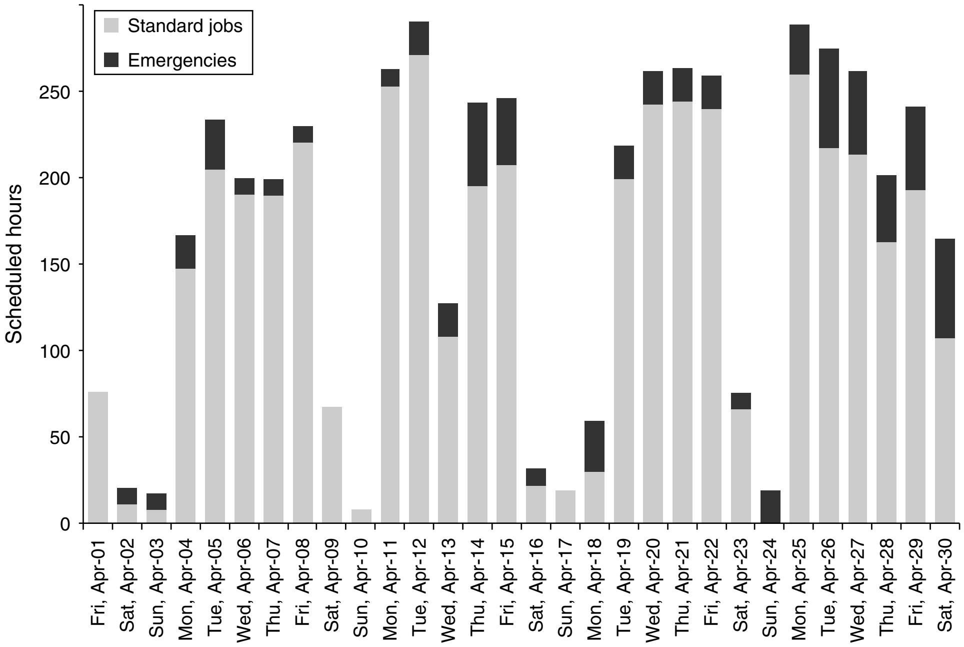
Resource planners make crew assignment decisions (assignment of standard jobs to crews) at the beginning of the day. Then they monitor the arrival of emergencies throughout the day. However, once the crew assignment is made, it is usually fixed for the rest of the day. They do not reassign a standard job to another crew once it has been initially assigned. Yards rarely postpone standard jobs in the case of multiple emergencies. This is because the set of jobs to be done in a given day needs to be known ahead of time to arrange for work permits, police detail protection at the work site, crew equipment, and other logistical requirements for performing the job.

2.2. Costs of Operations at Yards

Maintenance crews have eight-hour shifts but can work beyond their shifts if necessary. Any hours worked in excess of the crew's shift is billed as *overtime* and costs between 1.5 to 2 times as much as the regular hourly wage. Discussions with management reveal that it is preferable for maintenance crews to work overtime to complete standard job assignments rather than postpone standard jobs and risk incurring any regulatory fines for not meeting deadlines. Based on data from the company's yards, maintenance crews have been working a significant proportion of their hours at overtime. Between 25% to 40% of an average crew member's total work hours is on overtime.

Figure 2 shows the actual crew hours worked in April 2011 for one of the company's average-sized

Figure 2 **Crew Hours Worked Based on an Actual Schedule by a Yard's Resource Planner**



yards (with 25 weekday crews and 4 weekend crews). From this figure, we observe that the hours spent working on standard jobs are unevenly divided among the workdays. This variability of scheduled standard job hours from day to day observed in the figure is due to several factors, including variation in the number of crews (e.g., reduced operations during weekends, leaves, or unplanned absences) and standard jobs taking longer to finish than anticipated due to inclement weather or a complication during excavation (e.g., difficulty in locating the gas line). Resource planners are unable to anticipate variation in standard job durations since they only rely on experience and intuition to plan yard operations rather than historical data. As part of our project, our team built a regression model to predict standard job durations using job characteristics such as the job type, the size and diameter of the pipe, the age of the pipe, and the job location.

Another major cause of overtime is the stochastic number of emergencies per day. Resource planners decide on crew assignments without accounting for unplanned emergencies. As seen in Figure 1, there is a huge variation in the number of emergencies even within the same month. The crew assignment decisions are made without the use of quantitative models, based only on the resource planner's experience and subjective input from supervisors. Therefore, often resource planners do not provide slack capacity (i.e., idle crew hours) to respond to any emergencies that might occur later in the day. The variability of emergencies put resource planners in a reactive mode to meet deadlines as well as to handle emergencies, resulting in suboptimal resource allocation.

3. Modeling and Problem Decomposition

In this section, we discuss how we developed a stochastic optimization model for multiperiod planning of yard operations under random emergencies. The model decides the job schedule (i.e., determining which date each standard job is done) and at the same time decides the crew assignment (i.e., once a standard job is scheduled on a date, determining which crew is assigned to complete the job). Later in this section, we discuss a novel decomposition motivated by the practical limitations encountered during the project.

In what follows, we discuss all the assumptions in our model, motivated from the yard operations.

Assumption 1. *The number of crews available per day is deterministic, although this number can vary daily.*

Assumption 2. *There is no preemption of standard jobs.*

Assumption 3. *Standard jobs have deterministic durations. They do not necessarily have equal durations.*

Assumption 4. *The number of emergencies per day is stochastic. Emergencies have equal durations.*

Assumption 5. *Crew assignment does not take distances (geography) into consideration.*

Assumption 6. *The day can be divided into two parts (pre-emergency and post-emergency). In pre-emergency, the standard jobs are assigned to the available crews. Then the number of emergencies are realized. In post-emergency, these emergencies are assigned to the crews.*

Some of these assumptions were imposed for simplicity of the model. One of the requirements of the utility was to have a simple model for reasons we discuss later in §3.1.

Assumption 1 is due to staffing decisions not being part of yard operations since they are being made by the Resource Management Department based on company-wide projections of work for the year. Assumption 2 reflects the actual situation in yard operations due to significant startup effort for standard jobs (see §2.1 for further discussion). Assumption 3 is because standard job durations are accurately predicted by a regression model using factors such as the job type and the age of the pipe. We observe minimal variation between the regression's predicted values and actual values of job durations. Assumption 4 is due to the utility requiring its crews to devote a fixed amount of time on emergencies. However, the total number of emergencies in each day is stochastic based on the variation seen in yards (Figure 1). Assumptions 5 and 6 are reasonable from a practical point of view since the factors they ignore are second order in the model. Assumption 5 is assumed since travel time between jobs is usually much less than the duration of jobs. Assumption 6 means that we ignore the specific time that an emergency arrives. This is a reasonable assumption since regulation only requires a crew to be dispatched to an emergency within 24 hours (and not immediately as the emergency arrives). Therefore, if crews are already working on standard jobs when an emergency arrives, the emergency does not have to be responded to until a crew finishes its current job. However, it is possible that a crew might be idle if an emergency arrives after the crew finishes all its standard job assignments. Therefore, the problem setting can include dynamics in a given day and account for specific arrival times of emergencies. In §5.2, we provide a rolling horizon implementation for a dynamic assignment of standard jobs and emergencies that depends on specific arrival times of emergencies.

Next, we present our model for yard operations. Consider a set of standard jobs that need to be completed within a time horizon (e.g., one month). Each standard job has a known duration and a deadline. Without loss of generality, the deadline is assumed to be before the end of the planning horizon. Within a given day, a random number of emergencies may be reported. Reflecting actual yard operations, the number of emergencies is only realized once the standard job schedule and crew assignments for that day have been made. The following is a summary of the notation used in our model.

T: length of planning horizon;
K_t: number of crews available for work on day t, where $t = 1, \ldots, T$;
n: total number of known jobs;
d_i: duration of job i, where $i = 1, \ldots, n$;
τ_i: deadline of job i, with $\tau_i \le T$, where $i = 1, \ldots, n$;
d_L: duration of each emergency;
$L(\omega)$: number of emergencies under outcome ω;
Ω_t: (finite) set of all outcomes in day t, where $t = 1, \ldots, T$;
$P_t(\cdot)$: probability distribution of events on day t, $P_t\colon \Omega_t \mapsto [0, 1]$.

We can estimate the probability distribution of the number of emergencies, which is different for each yard and each month, from historical yard data. For example, Figure 1 can be used as the probability distribution for a yard in the month of April.

At the start of the planning horizon, the job schedule has to be decided. At the start of each day, the crew assignments need to be decided before the number of emergencies is known. This is because the calls for emergencies occur later in the day, but the crews must be dispatched early in the morning to their assigned standard jobs before these reports are received. After the number of emergencies is realized, the model decides on an assignment of the emergencies to the crews.

Let the binary decision variable X_{it} take a value of 1 if and only if the job i is scheduled to be done on day t. Let the binary decision variable Y_{itk} take a value of 1 if and only if job i is done on day t by crew k. If outcome ω is realized on day t, let $Z_{tk}(\omega)$ be the second-stage decision variable denoting the number of emergencies assigned to crew k. It depends on the number of standard jobs that have already been assigned to all the crews on day t. The variables $(X_{it})_{it}$, $(Y_{itk})_{itk}$ are the first-stage decision variables. The variables $(Z_{tk}(\omega))_{tk\omega}$ are the second-stage decision variables.

For each day t, a recourse problem is solved. In particular, given the day t crew assignments, $Y_t \triangleq (Y_{itk})_{ik}$, and the outcome of the number of emergencies, $L(\omega)$, the objective of the day t recourse problem is to choose an assignment of emergencies, $Z_t(\omega) \triangleq (Z_{tk}(\omega))_k$, so as to minimize the maximum number of hours worked over all crews. Thus, the day t recourse problem is

$$\begin{aligned} &F_t(Y_t, L(\omega)) \\ &\quad \triangleq \underset{Z_t(\omega)}{\text{minimize}} \quad \max_{k=1,\ldots,K_t} \left\{ d_L Z_{tk}(\omega) + \sum_{i=1}^{n} d_i Y_{itk} \right\} \\ &\quad\quad \text{subject to} \quad \sum_{k=1}^{K_t} Z_{tk}(\omega) = L(\omega) \\ &\quad\quad\quad\quad Z_{tk}(\omega) \in \mathbb{Z}^+, \quad k = 1, \ldots, K_t, \end{aligned} \tag{1}$$

where the term in the brackets of the objective function is the total hours (both standard jobs and emergencies) assigned to crew k. We refer to F_t as the day t recourse function. The constraint of the recourse problem is that all emergencies must be assigned to a crew.

The objective of the first-stage problem is to minimize the maximum expected recourse function over all days in the planning horizon:

$$\begin{aligned} &\underset{X,Y}{\text{minimize}} \quad \max_{t=1,\ldots,T} E_t[F_t(Y_t, L(\omega))] \\ &\text{subject to} \quad \sum_{t=1}^{\tau_i} X_{it} = 1, \quad i = 1, \ldots, n, \\ &\quad \sum_{k=1}^{K_t} Y_{itk} = X_{it}, \quad i = 1, \ldots, n,\ t = 1, \ldots, T, \\ &\quad X_{it} \in \{0, 1\}, \quad i = 1, \ldots, n,\ t = 1, \ldots, T, \\ &\quad Y_{itk} \in \{0, 1\}, \quad i = 1, \ldots, n,\ t = 1, \ldots, T, \\ &\quad\quad\quad k = 1, \ldots, K_t, \end{aligned} \tag{2}$$

where $F_t(Y_t, L(\omega))$ is described in (1). The constraints are (i) job i must be scheduled before its deadline τ_i, and (ii) if a job is scheduled for a certain day, a crew must be assigned to work on it. The optimization problem (2) can be rewritten as a mixed integer program. Section EC.1 of the electronic companion provides the MIP formulation.

The min–max objective function in (1) was chosen after many discussions with the utility management, the yard employees, and other key stakeholders. An earlier version of the model had the objective of minimizing the total expected labor cost of completing the jobs (which is equivalent to minimizing total expected overtime since straight hours are a sunk cost). However, the min–max objective led to better acceptance among all the key company stakeholders compared to a cost-related or overtime-related objective. Moreover, management views high overtime labor cost as a *symptom* of a problem in their operations and not the root cause that it wanted to solve. Management believes that the underlying problem that the company is faced with is an uneven distribution of both planned and unplanned work to the yard's crews. Another reason for the choice of objective is that, in the earlier version of the model that minimizes the expected overtime hours, the resulting solution was not acceptable to the company because of "fairness" issues and due to the risk of noncompliance with regulation. Even though the solution with the expected overtime objective has less overtime cost, in events with multiple leaks, it sometimes assigns one crew significantly more work hours compared to others (this crew incurs all the overtime hours while the rest do not).[1] In practice, yards need to comply with union rules on the amount of work a single crew can do; therefore, yards prefer to distribute work as evenly as possible among the available crews.[2] Motivated by these considerations, we decided to choose the objective of minimizing the expected maximum work hours.

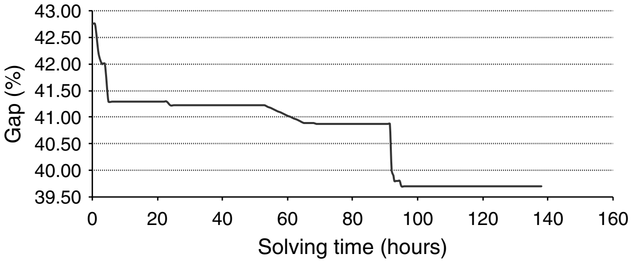
Our model assumes that standard jobs cannot be postponed if multiple emergencies appear in one day. However, it is possible to explicitly incorporate job postponement in a dynamic programming model. Note that such models are difficult to solve computationally (see a discussion by Godfrey and Powell 2002 on difficulties of solving multistage problems). In practical applications, the most natural solution strategy is to use a rolling-horizon, solving the static problem at each time period using what is known at that period and a forecast of future events over some horizon. In §4.3, we compare the rolling-horizon implementation to other dynamic models for job scheduling.

3.1. Practical Limitations of the Joint Job Scheduling and Crew Assignment Problem

Crew utilization is impacted by the combination of the job schedule and of the daily crew assignments. Hence, in model (2) the job scheduling decisions and crew assignment decisions are made jointly. However, there were several practical issues that prevented the implementation of the joint job scheduling and crew assignment problem in yard operations, which we discuss below.

First, the full optimization problem is intractable to solve with actual yard data within a reasonable amount of time. In actual yard settings, crew assignments need to be determined within at most a few minutes. If there are no emergencies, the problem is known to be NP-hard (Pinedo 2002). The presence of a stochastic number of emergencies makes the problem even more computationally intractable when solving the deterministic equivalent problem[3] using commercial off-the-shelf solvers. This is due to the structure of a stochastic MIP (Ahmed 2010). We demonstrate this by solving the deterministic equivalent problem with actual yard data[4] using Gurobi. Gurobi uses a branch-and-bound algorithm that systematically enumerates all candidate solutions, where large subsets of suboptimal candidates are discarded as a group, by using upper and lower estimated bounds of the optimal cost. The best integer solution found at any point in the search is called the incumbent solution. The cost of the incumbent is a valid upper bound to the optimal cost. The difference between the current upper and lower bounds is known as the gap. A gap of zero means that the incumbent solution is optimal. A large gap means that the solver cannot determine if the incumbent solution is optimal, and hence the branch-and-bound search continues. Figure 3 shows the gap in Gurobi's branch-and-bound search plotted against the solving time. Note that even after 140 hours, Gurobi still is only able to reduce this gap to about 40%. Therefore, for a yard that requires good solutions within a few minutes, implementing the joint problem (2) is impractical.

Figure 3 **Relative MIP Gap in Gurobi's Branch and Bound**



Notes. The gap represents the difference between the current upper and lower bounds on the optimal cost in the branch-and-bound procedure. When the gap is zero, the current solution is optimal.

A second practical issue our project faced is that at the onset of the project, the resource planners were hesitant to use quantitative decision models in replacing decisions that they have previously been making without guidance from any data or models, especially if the model follows a different decision-making process from their own.

[1] To illustrate this, consider a yard with a one-day horizon; three maintenance crews; eight standard jobs (with three hour durations each); and three emergency outcomes (zero emergencies with probability 0.2, one emergency with probability 0.6, or two with probability 0.2). If the objective is to minimize expected total overtime, then for the event of two emergencies, one crew will be working for 22 hours, whereas the other two crews work for 8 hours. If the objective is to minimize the expected maximum work hours, then no crew is working more than 16 hours in all outcomes.

[2] The number of crews is not a decision of the yard. See Assumption 1.

[3] With a finite number of scenarios, two-stage stochastic linear programs such as (2) can be modeled as large linear programming problems with many variables and constraints by enumerating all scenarios and introducing a second-stage decision variable for each scenario. This formulation is called the deterministic equivalent problem.

[4] Actual yard data had 481 standard jobs, 20 crews per weekday, 5 crews per weekend, 0 to 6 emergencies per weekday, 0 to 3 emergencies per weekend.

A third issue is that because of concerns about integration with the company's current databases and other strategic matters, the company chose not to invest in a commercial integer programming solver for a implementation of the project throughout the whole company. Therefore, any models and heuristics we develop needed to be solved using Excel's Solver or Premium Solver.

These practical issues motivated us to consider a decomposition of the joint problem into one in which the two decisions (job scheduling and crew assignment) are made sequentially. First is the job scheduling phase, which approximately schedules the jobs on the planning horizon assuming only an *average number of emergencies* on each day. The goal is to meet all the standard job deadlines while evenly distributing work (i.e., the ratio of scheduled work hours to the number of crews) over the planning horizon. Once the schedule of jobs is fixed, then the crew assignment problem can be solved independently for each day. In the crew assignment phase, the standard jobs are assigned to crews assuming a *stochastic number of emergencies*. The goal is to minimize the expected maximum hours worked by any crew. Note that the two-phase decomposition results in two layers of resource allocation problems. The first layer is a longer-term problem where the "resources" are the days that needed to be allocated to the standard jobs. The second layer is a one-day planning problem where the "resources" are the crews that needed to be allocated to the standard jobs and the random emergencies.

We discuss each phase of the decomposition in §§4 and 5. Note that the optimization problems resulting from the decomposition are more tractable because of the smaller problem dimensions. The decomposed problem also has better acceptance among the yard employees. This is because the decomposition mimics the sequential decision-making process done by the resource planners. Resource planners usually make scheduling decisions based on average emergencies because of the longer time horizon, whereas they consider variability as a problem that needs to be addressed in crew assignments within the day. Finally, we developed heuristics for solving each stage in the decomposition that can be implemented in Excel.

4. Phase I: Job Scheduling

In this section, we discuss the job scheduling phase, where standard jobs of varying durations and deadlines have to be scheduled on a planning horizon. We present a deterministic mixed integer program (MIP) whose solution is a feasible job schedule that evenly distributes work over the horizon. We also present a tractable algorithm for producing a job schedule. The algorithm is based on solving the LP relaxation, which, based on actual problem sizes, can be solved using Excel Premium Solver. The schedule resulting from the heuristic is near optimal in computational experiments and in actual yard problems.

In yard operations, there is a random number of emergencies per day, and the number of crews can change for different days. For instance, yards usually have fewer crews working during weekends compared to weekdays. Moreover, there are fewer company crews monitoring gas leaks during weekends, so there are usually fewer emergencies discovered during weekends. We chose to model the job scheduling phase to schedule standard jobs assuming a deterministic number of emergencies (equal to the average). That is, the standard jobs are scheduled to meet all the deadlines, while balancing (over all the days) the *average* hours scheduled scaled by the number of crews. The job scheduling phase solves the following optimization problem:

$$\begin{aligned}
\underset{X}{\text{minimize}}\quad & \max_{t=1,\ldots,T}\left\{\frac{1}{K_t}\left(d_L E_t[L(\omega)]+\sum_{i=1}^{n} d_i X_{it}\right)\right\}\\
\text{subject to}\quad & \sum_{t=1}^{\tau_i} X_{it}=1,\quad i=1,\ldots,n, \qquad (3)\\
& X_{it}\in\{0,1\},\quad i=1,\ldots,n,\ t=1,\ldots,T.
\end{aligned}$$

The motivation behind scaling the average scheduled hours per day by the number of crews is so that the optimal solution will schedule fewer hours on days when there are only a few crews.

Note that the scheduling decisions are made without a detailed description of the uncertainties. Rather, this phase simply takes the expected value of the number of emergencies per day. The stochasticity in emergencies will be handled in the crew assignment phase described in §5. Because of these modeling assumptions, the problem can be cast as an MIP with only a small number of variables and constraints.

Proposition 1. *Scheduling phase problem* (3) *can be cast as the mixed integer program*:

$$\begin{aligned}
\underset{C,X}{\text{minimize}}\quad & C\\
\text{subject to}\quad & d_L E_t[L(\omega)]+\sum_{i=1}^{n} d_i X_{it}\le K_t C,\\
& \qquad\qquad t=1,\ldots,T, \qquad (4)\\
& \sum_{t=1}^{\tau_i} X_{it}=1,\quad i=1,\ldots,n,\\
& X_{it}\in\{0,1\},\quad i=1,\ldots,n,\ t=1,\ldots,T.
\end{aligned}$$

This problem is related to scheduling jobs to *unrelated* machines with the objective of minimizing makespan when there are processing set restrictions (Pinedo 2002). The makespan is the total length of the schedule when all machines have finished processing the jobs.

In our setting, "machines" are equivalent to the dates $\{1, 2, \ldots, T\}$. Each job i is restricted to be scheduled only on dates (or "machines") before the deadline, i.e., on "machines" $\{1, 2, \ldots, \tau_i\}$. In our setting, the makespan of machine t is the ratio of scheduled hours to number of crews for day t.

Note that even the simpler problem of scheduling jobs on *parallel* machines is well known to be NP-hard (Pinedo 2002). List scheduling heuristics (where standard jobs are sorted using some criterion and scheduled on machines one at a time) are commonly used to approximately solve scheduling problems with parallel machines (Kafura and Shen 1977, Hwang et al. 2004, Glass and Kellerer 2007, Ou et al. 2008). For the case of unrelated machines, a well-known algorithm by Lenstra et al. (1990) performs a binary search procedure, in each iteration solving the linear programming relaxation of an integer program, and then rounds the solution to a feasible schedule. Using a proof based on graph theory, they show that the schedule resulting from their algorithm is guaranteed to have a makespan of no more than twice the optimal makespan.

4.1. LP-Based Job Scheduling Heuristic

In what follows, we describe an algorithm for approximating the solution for the job scheduling problem (3), which we refer to an *Algorithm LP-schedule*. Similar to Lenstra et al. (1990), this algorithm is also based on solving the LP relaxation and rounding to a feasible schedule. However, we do not require initializing the algorithm with a binary search procedure, therefore solving the LP relaxation only once. We are able to provide a data-dependent performance guarantee for LP-schedule (Theorem 1) that we derive using a novel technique based on stochastic analysis.

The following is a high-level description of Algorithm LP-schedule. (For details, refer to the electronic companion.) Consider the LP relaxation of the scheduling phase MIP (4) where all constraints of the form $X_{it} \in \{0, 1\}$ are replaced by $X_{it} \geq 0$. Let (C^{LP}, X^{LP}) be the solution to the relaxed problem. The algorithm takes X^{LP} and converts it into a feasible job schedule using a rounding procedure. The idea is to fix the jobs that have integer solutions while re-solving the scheduling problem to find schedules for the jobs that have fractional solutions. However, a job i with a fractional solution can now only be scheduled on a date t when the corresponding LP solution is strictly positive; i.e., $X_{it}^{LP} \in (0, 1)$. The rounding step solves an MIP; however, it only has $O(n+T)$ binary variables instead of the original scheduling phase integer problem, which had $O(nT)$ binary variables (the proof of this is similar to that in Lenstra et al. 1990).

The following theorem states that the schedule resulting from Algorithm LP-schedule is feasible (in that it meets all the deadlines), and its maximum ratio of hours scheduled to number of crews can be bounded.

Theorem 1. *Let C^{OPT} be the optimal objective cost of the scheduling phase problem* (4), *and let C^{LP} be the optimal cost of its LP relaxation. If X^H is the schedule produced by Algorithm LP-schedule, then X^H is feasible for the scheduling phase problem* (4) *and has an objective cost C^H, where*

$$C^H \leq C^{OPT} \times \left(1 + \frac{1}{C^{LP}}\left(\min_{t=1,\ldots,T} K_t\right)^{-1} \cdot \sqrt{\frac{1}{2}\left(\sum_{i=1}^{n} d_i^2\right)(1+\ln\delta)}\right), \tag{5}$$

where $\delta = \max_{t=1,\ldots,T} \delta_t$ and $\delta_t \triangleq |\{r = 1, \ldots, T\colon X_{ir}^{LP} > 0$ and $X_{it}^{LP} > 0\}|$.

The proof is based on stochastic analysis and can be found in the electronic companion. We provide an outline of the proof: Introduce $\tilde{X}$ as the randomized schedule derived by interpreting the LP solution X^{LP} as probabilities. For example, if $X_{i1}^{LP} = X_{i2}^{LP} = 0.5$, then job i is equally likely to be scheduled on day 1 and day 2 in the random schedule. All outcomes of $\tilde{X}$ are all the possible roundings of X^{LP}. Note that the algorithm produces the rounding X^H with the smallest cost (the maximum ratio of scheduled hours to number of crews). Therefore, if we can prove that there exists a *positive probability* that $\tilde{X}$ has a cost bounded by the right-hand side of (5), then this proves Theorem 1. We do this by defining B_t as the "bad" event that $\tilde{X}$ has a day t ratio of scheduled hours to number of crews greater than the right-hand side of (5). We need to show that with positive probability none of the bad events $B_1, B_2, \ldots, B_T$ occur. Note that each bad event is mutually dependent on at most δ other bad events. Then if there exists an upper bound on $\Pr(B_t)$ for all t, we can use Lovász's Local Lemma (Erdős and Lovász 1975) to prove that there is a strictly positive probability that none of the "bad" events occur. We use McDiarmid's Inequality (McDiarmid 1989) to derive an upper bound on $\Pr(B_t)$.

We now demonstrate using (5) that the schedule resulting from the algorithm has a cost closer to optimal if the job deadlines are more restrictive (i.e., more standard jobs are due earlier in the horizon) or if the job durations are less variant (i.e., standard jobs have very similar durations). Suppose that there are K crews per day. Note that C^{LP} takes its smallest value when all jobs are due on the last day, with $C^{LP} = (\sum_{i=1}^{n} d_i)/(KT)$. On the other extreme, if all the deadlines are on the first day, C^{LP} takes its largest value with $C^{LP} = (\sum_{i=1}^{n} d_i)/K$. Hence, the bound is smaller under more restrictive deadlines. Furthermore, note that δ_t represents (based on the LP solution) the number of days that share a fractional job with day t. With more restrictive deadlines, we would expect δ_t to be smaller, implying that $\delta = \max_t \delta_t$ is smaller. Finally, consider the case where $C^{LP} = (\sum_{i=1}^{n} d_i)/(\alpha K)$, for some constant $\alpha > 0$ (note that this is the case when deadlines are either all on the first day or all on the last day). Then the bound simplifies to $1 + \alpha(\|d\|_2/\|d\|_1)\sqrt{\frac{1}{2}(1+\ln\delta)}$, where d is the vector of job durations. Interpreting $\|d\|_2/\|d\|_1$ as the coefficient of variation in job durations, we can infer that the bound is smaller if there is less variance in the job duration data.

In both randomly generated job scheduling problem instances as well as actual yard problems, we observe that the data-dependent multiplicative factor in (5) is less than two. But in some cases, the factor might become large, for instance as T increases. However, we can modify the algorithm, ensuring that the resulting schedule has a cost of no more than αC^{OPT}, where α is the minimum of two and a data-dependent expression (Theorem EC.1 in the electronic companion). Hence, the bound will not explode in asymptotic regimes. The modification is to initialize the algorithm with binary search procedure (described in §EC.4 of the electronic companion).

We next compare Algorithm LP-schedule to the optimal schedule on actual yard data for one month. In that month, there were 481 standard jobs with durations ranging from three hours to nine hours. On weekdays, there were 20 crews available per day, and the number of emergencies ranged from zero to six per day. On weekends, there were five crews available per day, and emergencies ranged from zero to three per day. The job scheduling problem (4) implemented in Gurobi did not terminate with an optimal solution after several days. However, since C^{OPT} is bounded below by C^{LP}, we used the LP relaxation solution to determine that our algorithm results in a schedule that is at most 5.6% different from the optimal job schedule. In the remainder of this section, we compare LP-schedule to other scheduling heuristics on simulated data and on actual yard data.

4.2. Comparing Algorithm LP-Schedule to a Sensible Resource Planner

We randomly generate 100 problem instances and compare the schedule resulting from Algorithm LP-schedule to a schedule that a sensible resource planner might otherwise produce following some rules-of-thumb. An algorithm that follows the sensible resource planner's rules will be referred to as *Algorithm SRP*. SRP's rules have been determined after consulting with several of the utility's resource planners. In SRP, the standard jobs are sorted with increasing deadlines so that the job with the earliest deadline comes first in the list. Then SRP will determine a cutoff value for work hours per day. Starting from the first day in the horizon, SRP will go through the sorted list of jobs. If the current job has a deadline of today *or* if the current work hours scheduled for today is less than the cutoff, SRP will schedule the current job for today and remove it from the list. Otherwise, it does not schedule it today and moves on to the next day. The cutoff used by SRP is the total work hours divided by the number of days.

In each problem instance, there are seven days in the planning horizon, and three crews available each day. There are 70 standard jobs to be scheduled (with durations randomly generated between zero to eight hours, and deadlines randomly chosen from the seven days). The size of the problem instance is chosen so that the job scheduling problem solves to optimality within a reasonable amount of time. For each problem instance, we apply both LP-schedule and SRP, noting the cost of both schedules, i.e., the maximum ratio of average scheduled hours to number of crews. A schedule is near optimal if its cost is close to the optimal cost from solving the scheduling problem (4). For each problem instance, we compute the percentage difference of the heuristics' cost to the optimal cost. Algorithm LP-schedule has a sample mean (taken over 100 instances) for the percentage difference equal to 3.6%. The 95% confidence interval for this sample mean is [2.9%, 4.2%]. On the other hand, SRP has a sample mean for the percentage difference equal to 9.7%. The 95% confidence interval for this sample mean is [9.1%, 10.2%]. We can infer that scheduling with Algorithm LP-schedule results in a cost closer to the optimal cost than scheduling by SRP since the range of values of LP-schedule's confidence interval falls below the range of values for SRP's confidence interval. Additionally, Algorithm LP-schedule manages to improve computational efficiency. Solving for the optimal schedule in (3) often requires several hours, which is not viable in actual yard operations. On the other hand, Algorithm LP-schedule only takes a few seconds to solve.

4.3. Dynamic Job Scheduling

The Phase I model results in a static job schedule. In the case of the utility, yards rarely postpone standard jobs in the case of multiple emergencies (see discussion in §2). However, one can potentially solve the job scheduling problem with a rolling horizon so that standard jobs can be rescheduled as more information is revealed. In practice, static models are often solved with a rolling horizon rather than solving a dynamic program. This is because dynamic resource allocation models are computationally intractable, with solution methods often only approximating the value function (Godfrey and Powell 2002, Huh et al. 2013).

Using actual yard data for one month, we compare three job scheduling heuristics: the static Algorithm LP-schedule, the rolling horizon implementation of LP-schedule, and Algorithm SRP. To evaluate the cost of these schedules, we compare their costs relative to the cost of a *perfect hindsight* job schedule, which is the optimal job schedule after knowing the sequence of emergencies occurring each day. The cost of the perfect hindsight job schedule is clearly smaller than any dynamic job schedule since it has the advantage of complete information. The cost of the perfect hindsight model is unachievable in reality. However, we use it to benchmark against the costs of the three heuristics.

The experiments use actual yard data for one month (481 standard jobs, 20 weekday crews, 5 weekend crews). The static job schedule is the output of Algorithm LP-schedule applied to the one-month horizon. To implement the rolling horizon schedule, Algorithm LP-schedule is reapplied every day, with a horizon starting from the current day until the end of the month. On the current day, the number of emergencies is known (for our experiments, it is randomly drawn from the empirical probability distribution shown in Figure 1). The algorithm is applied using the known number of emergencies for the current day and an expected number of emergencies for the remaining days. The SRP job schedule uses the rules described in §4.2. We compare the three schedules to the perfect hindsight schedule where the sequence of emergencies is known.

Figure 4 shows the histograms of the three heuristics' costs relative to the perfect hindsight cost generated from 100 sample paths for emergencies. In the experiments, the total number of emergencies for the month varied between 48 to 88. Note that the cost of the schedule produced by the sensible resource planner is the highest of the three, with an average percentage difference of 123%. The average percentage difference of the rolling horizon schedule is only 11.4%, whereas for the static schedule it is 27.3%. For 80% of the sample paths, the percentage difference of the rolling horizon schedule is no more than 16%. On the other hand, the 80th percentile of the static schedule percentage difference is 39%. Based on these experiments, we conclude that if the yard could postpone standard jobs, then it can reap significant cost savings from implementing a dynamic schedule rather than a static schedule. Moreover, its small percentage cost difference compared to the perfect hindsight cost implies that a simple rolling horizon implementation of the job scheduling problem already achieves most of these gains compared to the best dynamic schedule or a schedule obtained from solving a dynamic program.

5. Phase II: Crew Assignment

In this section, we focus on the second phase of the decomposition, i.e., the crew assignment problem within one day. The basic problem is determining which crews should execute which standard jobs and which crews to reserve for emergencies, given the stochastic number of emergencies. We develop an algorithm for crew assignment, which we motivate from a property of the optimal crew assignment. We can show that, in some special cases, the algorithm terminates with an optimal crew assignment. We also modify the algorithm for a multiperiod setting that allows reassignments and evolution of forecasts for the emergencies within the day.

Denote by I the set of standard job indices to be assigned in one day (as determined in the job scheduling phase). Let L be the stochastic number of emergencies on that day. Suppose there are K crews available on that day. The crew assignment problem assigns all standard jobs in set I to the available crews. However, these assignments must be made before the number of emergencies for the day is realized. After the number of emergencies is known, all emergencies must be assigned to the crews. The objective is to minimize the expected maximum hours worked on that day.

The crew assignment problem for one day solves a *two-stage stochastic mixed integer program*. The first-stage problem is

$$\begin{aligned} \underset{Y}{\text{minimize}} \quad & E[F(Y, L(\omega))] \\ \text{subject to} \quad & \sum_{k=1}^{K} Y_{ik} = 1, \quad i \in I, \\ & Y_{ik} \in \{0, 1\}, \quad i \in I,\ k = 1, \ldots, K, \end{aligned} \tag{6}$$

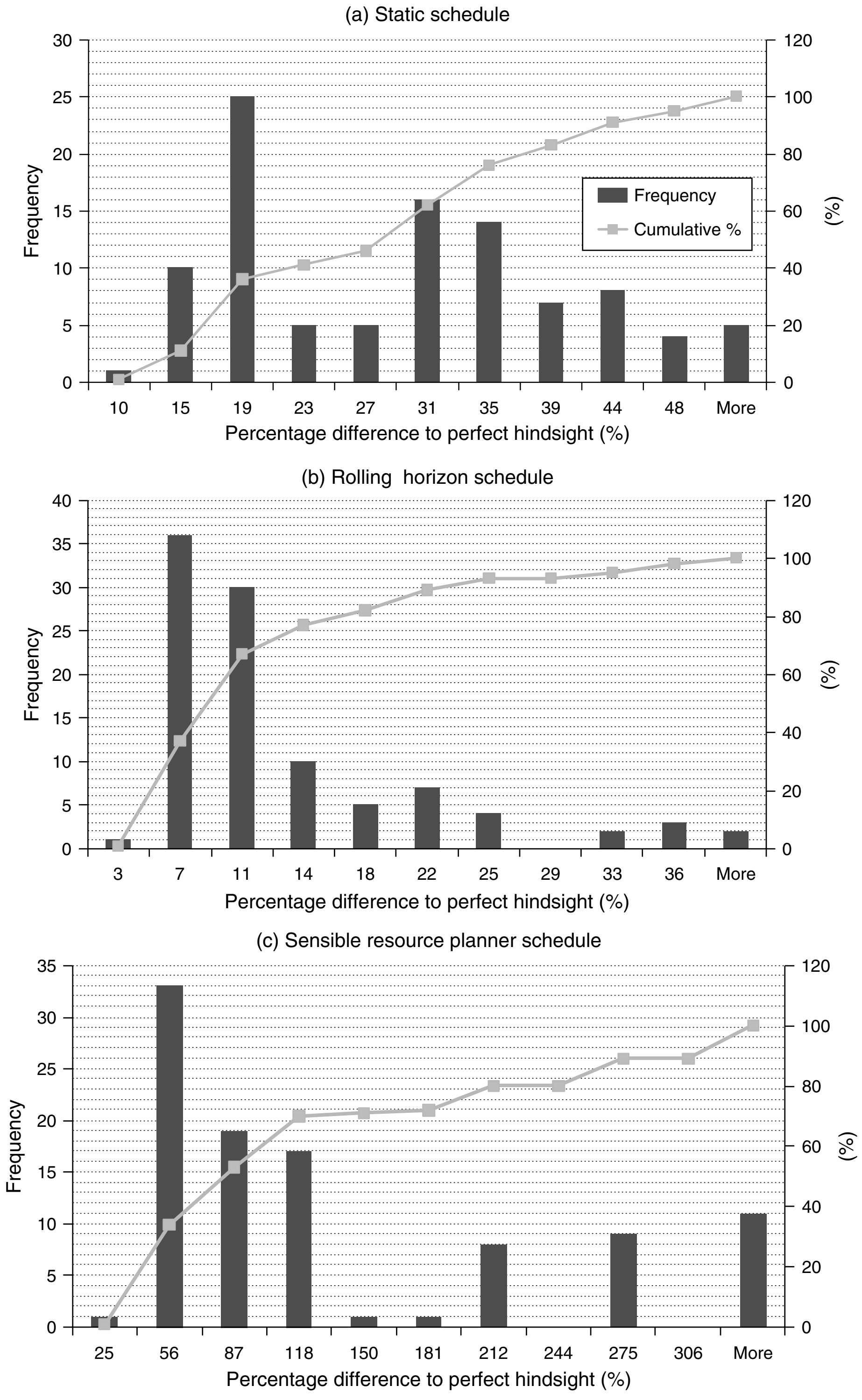
where $F(Y, L(\omega))$ is defined as

$$\begin{aligned} F(Y, L(\omega)) \triangleq \underset{Z}{\text{minimize}} \quad & \max_{k=1,\ldots,K} \left\{ d_L Z_k + \sum_{i \in I} d_i Y_{ik} \right\} \\ \text{subject to} \quad & \sum_{k=1}^{K} Z_k = L(\omega), \\ & Z_k \in \mathbb{Z}^+, \quad k = 1, \ldots, K. \end{aligned} \tag{7}$$

Note that the term in the brackets of the objective function is the number of hours assigned to crew k during outcome ω and under the standard job assignments Y. The assignment phase problem can also be solved as a mixed integer program (see §EC.6 of the electronic companion).

Even if the number of emergencies is known, the deterministic crew assignment problem is NP-hard. With a stochastic number of emergencies, the problem is even more computationally intractable (Ahmed 2010). The difficulty with stochastic integer programming is that some constraints are repeated with small changes for each scenario. One of the important features of commercial off-the-shelf solvers is the generation of cutting planes by analyzing the polyhedral relaxation of the problem and expediting branch-and-bound

Figure 4 Histogram of Static, Dynamic, and SRP Cost Percentage Difference Relative to the Perfect Hindsight Cost



search. Typically, such cuts are generated by analyzing a single row of the constraint system. A stochastic IP is computationally difficult since information from multiple rows in its constraint set is lost (Ahmed 2010). With actual yard data, the crew assignment problem (6) implemented in Gurobi terminates with an optimal solution only after several hours.

Motivated by the intractability of solving the problem to optimality, we developed a crew assignment algorithm that exploits the specific structure of the crew assignment problem, which we will describe later in §5.1. This algorithm is simple and intuitive since it can be viewed as a stochastic variant of the LPT rule. The algorithm we developed also resulted in natural guidelines for resource planners to follow in making yard operations under a stochastic number of emergencies.

5.1. Crew Assignment Heuristic

We conducted computational experiments on several examples to gain insight into the structure of the optimal crew assignment solution to (6). Section EC.11 of the electronic companion explains in detail the experiments we conducted. An observation we make from the experiments is that in the optimal solution, if a crew is assigned to work on an emergency in a given emergency outcome, that crew should also be assigned to work on an emergency under outcomes with more emergencies. This is formalized in the following proposition.

Proposition 2. *There exists an optimal solution* $(Y^*, Z^*(\omega), \omega \in \Omega)$ *to the stochastic assignment problem* (6) *where each crew's emergency assignment is monotonic in the number of emergencies. That is, if* $L(\omega_1) < L(\omega_2)$ *for some* $\omega_1, \omega_2 \in \Omega$, *then* $Z_k^*(\omega_1) \leq Z_k^*(\omega_2)$ *for all* $k = 1, \ldots, K$.

This proposition motivates our heuristic for crew assignment under stochastic emergencies. The heuristic results in crew emergency assignments that are monotonic in the number of emergencies. We refer to the heuristic as *Algorithm Stoch-LPT* since it is is a variant of the LPT algorithm under a stochastic number of emergencies. LPT applies when there are no emergencies, and the objective is to minimize the maximum work hours of the crews. In each iteration of LPT, it keeps track of the current number of assigned work hours (current load) for each crew. LPT initializes the current load for each crew to be zero. Then it sorts the standard jobs in decreasing duration. Starting with the longest duration job, each iteration of LPT assigns the current standard job to the crew with the smallest current load, updating the current load after an assignment is made.

In what follows, we describe Algorithm Stoch-LPT for stochastic emergencies, with the objective of minimizing the *expected* maximum work hours of the crews. The algorithm begins by first making assignments of emergencies in each outcome of L (the stochastic number of emergencies). For example, in an outcome with two emergencies, the algorithm needs to assign two emergencies to the crews. For each outcome, the algorithm assigns emergencies, starting with the outcome with the least emergencies, then the one with the second least, continuing until it assigns all emergencies under all outcomes. For the current outcome, Algorithm Stoch-LPT uses a procedure for assigning the emergencies that preserves the property of monotonic crew emergency assignments described in Proposition 2. It assigns the emergencies in the current outcome to the crews by the LPT rule. But in case of ties (where more than one crew has the smallest current load), it chooses a crew whose current load is strictly smaller than its load in the previous outcome's assignment.

After emergencies have been assigned for all outcomes, the next step in Stoch-LPT is to assign the standard jobs. The algorithm keeps track of the current load of each crew *in each outcome*, which is initialized after the crews' emergency assignments. Then, like LPT, the algorithm sorts the standard jobs in decreasing order of duration. Starting with the longest duration job, each iteration of Stoch-LPT assigns the current standard job to a crew according to the following rule. Under each crew, determine the increase in expected maximum load that results from assigning the current job to that crew. Note that different assignments result in different loads for crews in a given outcome of L. The expected maximum load is computed by summing over all outcomes the maximum load in the outcome multiplied by the probability. The standard job is assigned to the crew that has the smallest amount of increase in the expected maximum load. If there are any ties, the standard job is assigned to the crew with the smallest expected current load. After a standard job is assigned, the current load of each crew in each outcome is updated.

We next discuss an implication of having crew emergency assignments that are monotonic in the number of emergencies. In reality, the leak outcome reveals itself over time since leaks are discovered throughout the day. However, as the proposition states, if a crew is assigned to an emergency for an outcome with one leak, then this same crew is assigned at least one emergency for outcomes with two, three, and more leaks. Therefore, the first leak that appears in any outcome is always assigned to that crew. This way, one can "rank" the crews that handle the emergencies. Thus, the crew assignment solution of the static model can be easily implemented in a real-time setting where leaks arrive throughout the day. This ranking of crews based on the optimal leak assignment is formalized in the following proposition.

Proposition 3. *Suppose* $(Y, Z(\omega), \omega \in \Omega)$ *is a feasible solution to the stochastic assignment problem* (6) *with crew emergency assignments that are monotonic in the number of emergencies; i.e., if* $L(\omega_1) < L(\omega_2)$ *for some* $\omega_1, \omega_2 \in \Omega$, *then* $Z_k(\omega_1) \leq Z_k(\omega_2)$ *for all* $k = 1, \ldots, K$. *Then crews can be relabeled as* $k_1, k_2, \ldots, k_K$ *so that* $Z_{k_{j-1}}(\omega) \geq Z_{k_j}(\omega)$ *for all* $\omega \in \Omega$.

Corollary 1. *There exists an optimal solution* $(Y^*, Z^*(\omega), \omega \in \Omega)$ *to the stochastic assignment problem* (6) *where the crews can be relabeled as* $k_1, k_2, \ldots, k_K$ *so that* $Z^*_{k_{j-1}}(\omega) \geq Z^*_{k_j}(\omega)$ *for all* $\omega \in \Omega$ *and* $\sum_{i\in I} d_i Y^*_{i,k_{j-1}} \leq \sum_{i\in I} d_i Y^*_{i,k_j}$.

In what follows, we investigate in several examples how the crew assignment produced by Algorithm

Stoch-LPT compares to the optimal crew assignment solution to (6) and to crew assignments from other heuristics. All examples use the same 15 standard jobs and seven crews but a different probability distribution of the number of emergencies. (See Tables EC.1 and EC.2 in the electronic companion for the data.) All distributions have an expected number emergencies equal to one. We refer to the optimal crew assignment as *OPT*. We consider two other crew assignment heuristics aside from Stoch-LPT. First is the heuristic that solves a deterministic crew assignment model assuming that *the number of emergencies is equal to the expectation* $E[L]$ (in the examples, the $E[L]=1$). This heuristic is motivated from practice since often practitioners facing uncertainty in their operations choose to ignore it by planning their operations based on the average uncertainty. We refer to this heuristic as *AVG*. The second heuristic we consider approximates the deterministic crew assignment optimization problem AVG (whose set of jobs includes the standard jobs and one emergency) by applying the LPT rule. We refer to this heuristic as *1-LPT*. Note that AVG and OPT both solve NP-hard optimization problems. 1-LPT and Stoch-LPT are heuristics that approximate the problems, respectively. Moreover, 1-LPT is a method that is a closer approximation of how resource planners currently make their assignment decisions since they make decisions based on averages and they do not have the computational resources to solve optimization problems.

Table 1 summarizes the expected maximum work hours. It illustrates under which emergency leak distributions each heuristic performs best. Note that AVG has near-optimal expected maximum hours in most examples, except under leak distribution 4. This distribution has the highest probability of having more than one leak. The assignment that AVG produces is not robust to an outcome with a large number of leaks. Now, let us compare Stoch-LPT to 1-LPT. Note that in most of the examples, Stoch-LPT has a smaller expected maximum hour than 1-LPT. Moreover, 1-LPT also does poorly when there is a high probability of an outcome with many leaks.

Table 1 **Comparing Algorithm Stoch-LPT to the Optimal Crew Assignment and Other Heuristics**

	Expected maximum hours				% difference to OPT		
	OPT	AVG	1-LPT	Stoch-LPT	AVG	1-LPT	Stoch-LPT
Leak distribution 1	10.66	10.66	11.50	11.50	0.00	7.96	7.96
Leak distribution 2	11.41	11.42	12.13	12.06	0.06	6.28	5.72
Leak distribution 3	11.78	12.18	12.75	12.75	3.39	8.24	8.23
Leak distribution 4	11.79	13.70	14.00	12.67	16.23	18.73	7.50
Leak distribution 5	12.18	12.18	12.77	12.19	0.03	4.80	0.11
Leak distribution 6	12.50	12.95	13.39	13.44	3.59	7.13	7.52
Leak distribution 7	12.85	12.95	13.40	13.34	0.75	4.27	3.79

Section EC.10 of the electronic companion provides analytical results on the performance of Algorithm Stoch-LPT relative to the optimal crew assignment solution when there are only two crews and two emergency outcomes. We prove that in the special case of equal job durations, the algorithm terminates with an optimal crew assignment.

5.2. Dynamic Crew Reassignment

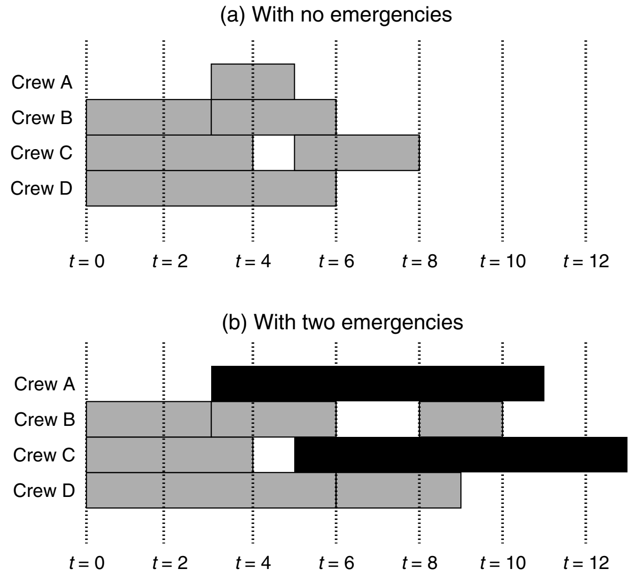
Motivated by yard operations where assignment of standard jobs is determined once in the beginning of the day and cannot be changed, our model for the crew assignment is static. We now modify Algorithm Stoch-LPT such that standard jobs not yet started can be reassigned later in the day as more information about the emergencies becomes available.

We assume that the standard jobs can be reassigned every hour. We also assume that the arrival of emergencies follow a Poisson process with an arrival rate λ. Any arrival process can be used; however, we chose a Poisson process for the purpose of illustration. Recall that the emergencies are found by dedicated company crews that monitor for gas leaks in a shift of eight hours. Then there is a natural update rule for the belief on the number of emergencies. Suppose there are s hours remaining until company crews stop monitoring for leaks. Then the probability that n emergencies are found within s hours is $P(L=n)=((\lambda s)^n/n!)e^{-\lambda s}$, for $n=0,1,2,\ldots$.

What we refer to next as the dynamic crew reassignment is the following. At the start of the day, determine the standard job and emergency assignments according to Algorithm Stoch-LPT. We make a distinction between Stoch-LPT's assignments and the *actual* assignments that are based on the actual outcome of the number of emergencies. In the first hour, the number of emergencies are realized according to the Poisson distribution. Make the actual emergency assignment based on Algorithm Stoch-LPT. For example, if there is one emergency, assign that emergency based on the algorithm's emergency assignment under the outcome with one emergency. If a crew has an actual emergency assignment, it starts work on that emergency in the current hour. Otherwise, choose an actual standard job assignment from the set of standard jobs that Stoch-LPT assigns to that crew. We choose the longest duration job in the set.[5] The crew starts work on the chosen standard job (if any) in the current hour. Moving to the next hour, we again apply Algorithm Stoch-LPT, but (i) with only the standard jobs not yet started,

[5] It is possible to have a different rule for choosing the actual standard job assignments. For instance, one can choose the shortest duration job in the set.

Figure 5 An Example of a Dynamic Assignment



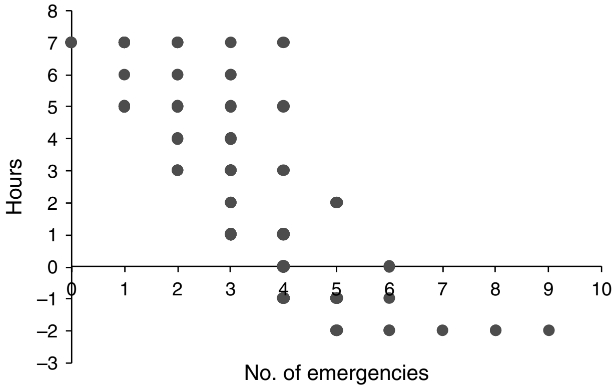
Notes. The horizontal axis represents time. A rectangle represents a job; its duration is proportional to the length of the rectangle. Grey rectangles represent standard jobs. Black rectangles represent emergency jobs.

(ii) with the current load of some crews reflecting the job they started from the previous hour, and (iii) with an updated probability distribution of the number of emergencies. Emergencies are realized for that hour according to the Poisson distribution, and actual assignments of emergencies and standard jobs are determined as before. Then continue the reassignment at the start of each hour until either there are no more standard jobs left or the end of eight hours is reached. At the end of the last hour, the LPT rule is applied for the remaining standard jobs (if any).

Figure 5 is an example of how the crew assignment evolves as the emergencies arrive. The example uses four crews, six jobs, and an emergency arrival rate of 0.2 per hour. Grey rectangles represent standard jobs. Black rectangles represent emergency jobs. Figure 5(a) shows the job assignment when there are no emergencies. Figure 5(b) shows the job assignment when two emergencies arrive at $t = 3$ and $t = 5$. White space between jobs shows that the crew is idle during that period. Note that, depending on the arrival of emergencies, the standard jobs assignments are different.

We compare the dynamic reassignment solution to the static solution using simulation experiments. Consider a yard with 17 crews and 21 standard jobs with durations varying from three to nine hours. An emergency has a duration of eight hours. Emergency arrivals follow a Poisson process with rate 0.352 per hour. Emergencies can only arrive in an eight-hour period, during which there is an expected number of 2.8 emergencies. We simulate 100 sequences of emergency arrivals. For each sequence, we apply both the static Stoch-LPT and the dynamic Stoch-LPT. Figure 6 shows the number of work hours saved by the dynamic reassignment plotted against the number of emergencies in the sequence. The static assignment, which is decided at the start of the day, is conservative. It does not adjust its solution and maintains some idle crews even toward the end of the day if no or few emergencies arrive. Therefore, some of the hours logged by crews are actually idle time while waiting for emergencies. For these emergency sequences, the dynamic reassignment converts some of the idle time into time actually working, thereby reducing the total number of work hours. However, in sequences with many emergencies, dynamic reassignment often results in more work hours compared to the static assignment. A disadvantage of the dynamic reassignment is that the resource planner has to coordinate changing the crew assignments around and calling supervisors to update them about the new assignments. Therefore, in the yard setting, the utility has chosen to implement a dynamic model, but with reassignment only occurring once midday.

Figure 6 Crew Work Hours Saved by Dynamic Reassignment



Note. Each data point corresponds to a different sequence of emergency arrivals.

6. Business Analytics for a Utility's Gas Business

In this section, we describe how the research above applies to the scheduling of operations at the gas business of a large multistate utility. This is based on a joint project between the research team and the company that gave rise to the results of this paper. Refer to §2 for the details of the utility's operations. We discuss how we used the optimization models and heuristics described in this paper so that the company could develop better strategies to create flexibility in its resources to handle emergencies.

6.1. Overview of the Project

At the onset of the project, our team analyzed sources of inefficiency in yard operations by mapping in detail the existing yard processes. We visited several company yards and interviewed a number of resource planners, supervisors, and crew leaders as well as members of the Resource Management Department. Our team shadowed crews from multiple yards performing different types of jobs and documented the range of processes followed. We also constructed historical job schedules based on data gathered from the company's job database.

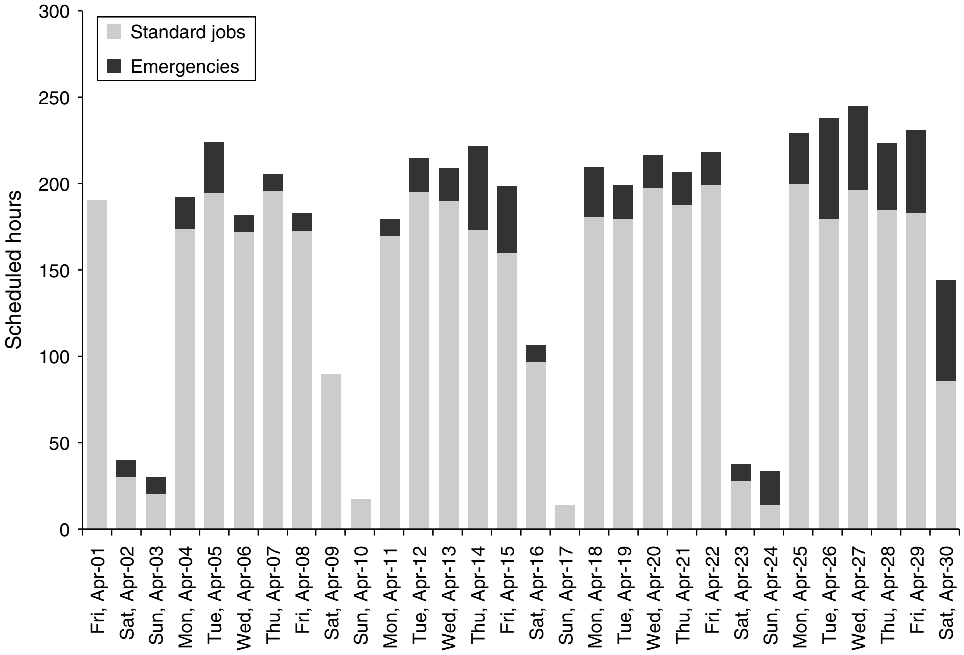
Our project with the utility had three main objectives. The first was to develop a tool that can be used with ease in the company's daily resource allocation. Based on the models and heuristics we discuss in this paper, we created a tool—the Resource Allocation and Planning Tool (RAPT)—to efficiently schedule jobs and to assign them to crews while providing flexibility for sudden arrival of emergencies. RAPT has access to the job and time-sheet databases and uses this information to estimate the distribution of the number of emergency jobs and their durations. The resource planner can view a webpage showing RAPT's output of the weekly schedule for each crew and detailed plans under different emergency outcomes. This tool is being piloted in one of the company's yards.

The second objective was to create and improve processes related to daily resource allocation so that the tool could be easily embedded into daily scheduling process. We observed that a lot of the data in the database were either missing, inappropriately gathered or not vetted before entry into the system. Having missing or inaccurate data makes it difficult to apply a data-driven tool such as RAPT. Processes were created to ensure that when new jobs were added to the database, they had the right database fields set in a consistent manner across all jobs and yards.

The third objective was to analyze the impact of key process and management drivers on operating costs and the ability to meet deadlines using the optimization model we developed. Results from this analysis will help the company deploy the optimization model with all the necessary process and management changes to capture the potential benefits outlined in this paper. This analysis of process improvement is discussed in §6.2. For yard-level process improvements, RAPT also has an option for resource planners to analyze the impact of different process changes to the yard's key performance indicators using simulation, thereby enabling better informed decisions.

Finally, we set out to determine the potential impact of the RAPT tool to the company. Refer to Figure 2, which shows the actual crew hours worked in April 2011 for one of the company's average-sized yards (with 25 weekday crews and 4 weekend crews). Figure 7 shows the profile for the same set of jobs if RAPT is used to schedule jobs and assign them to crews. The result is a 55% decrease in overtime crew hours for the month. Clearly, the schedule and crew assignments produced by RAPT are superior to those produced previously by the resource planner. However, even if compared to the best possible schedule where uncertainty is removed, the decisions produced by RAPT compare favorably. The "perfect hindsight" scheduling and assignment decisions are based on complete knowledge of the outcomes of emergencies that occur

Figure 7 **Hypothetical Scenario: Crew Hours Worked if Optimization Model Is Used to Schedule Jobs**



in the month. The "perfect hindsight" model results in the maximum possible reduction in overtime since the yard can plan completely for emergencies. Even though the RAPT model assumes a random number of emergencies, it still is able to capture 98.6% of the maximum possible overtime reduction by "perfect hindsight."

6.2. Using the Model to Recommend Process Improvements

Using the models from this paper, we conducted a study to understand the impact of changes in yard processes on yard productivity. Based on past studies the company had conducted, the company understands that yard productivity is driven by process settings such as the size of work queues (i.e., jobs available for scheduling), effective supervision, incentives, and cultural factors. Our team analyzed three drivers of productivity: work queue level, use of crew-specific productivity data, and the degree of field supervision. The analyses were used to recommend specific process changes to management and their resulting productivity and cost improvements.

6.2.1. Optimal Work Queue Level. Jobs need to be in a "workable" state before crews can begin executing them. For instance, one of the steps in getting a job to a workable state is to apply for a permit with the city of jurisdiction. Jobs in a "workable jobs queue" are jobs ready to be scheduled by RAPT. A queue is maintained since "workable" jobs are subject to expiration and require maintenance to remain in a workable state (e.g., permits need to be kept up to date). We observed some yards kept only a few workable jobs in the queue at times. The short workable jobs queue adversely impacted the RAPT output by not fully utilizing the tool's potential. The team decided to run simulations to determine a strategic level of workable jobs in the queue to maximize the impact of RAPT while minimizing the efforts to maintain the workable jobs queue.

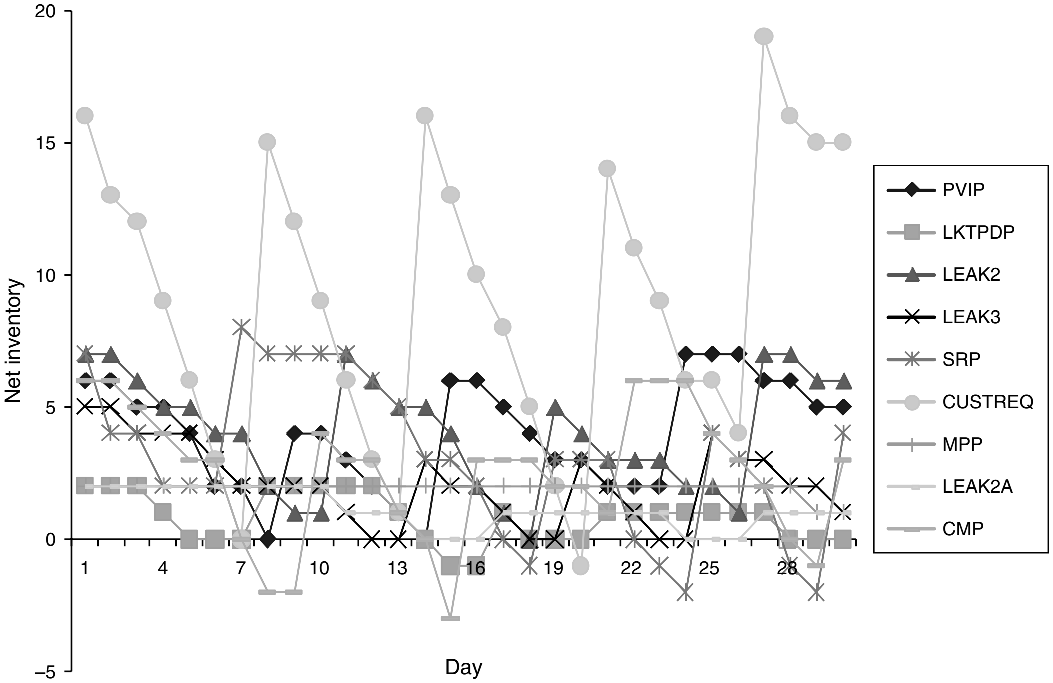
In what follows, we describe simulations using data from one of the company's yards. Five crews, each with eight-hour shifts, are available to work in each simulated day. There are 10 types of jobs to be executed (9 standard job types and 1 emergency job type). Table EC.17 in the electronic companion shows the average durations of the job types. On each day, the Resource Management Department announces a minimum quota of the number of jobs required to be done for each standard job type. These quotas are random and depend on various factors beyond the yard's control. Based on historical quotas, we estimate the probability distribution of daily quotas for each job type. Table EC.17 gives the probability distribution of quotas for the standard job types. To meet these quotas, the yard maintains a workable jobs queue for each standard job type. For example, consider the CMP job type in Table EC.17 with an average duration of 5.85 hours. Suppose today the quota for CMP jobs is three, but there is only one job in the workable CMP job queue. Then today the yard will execute one CMP job and will carry over the remaining two CMP jobs as a backlog for the next day.

Getting jobs to a workable state (and into the queue) is done by the yard's resource planner, who processes the paperwork and applies for city permits. Since resource planners have many responsibilities in the yard (including scheduling, dispatching, and monitoring crews), they only process paperwork for jobs in batches and only when the jobs in the queue reach a low number. Hence, the workable jobs queue is replenished through a continuous review policy specified by a *reorder point* and an *order quantity*. Each time the total workable jobs (both in the queue and in the pipeline) drops below the reorder point, the yard requests new workable jobs. The size of the request is equal to the order quantity. The request is added to the pipeline and arrives after a lead time of three days. This lead time includes time used for administrative work to apply for a permit. Suppose the yard chooses a reorder point of two and an order quantity of five for the CMP workable jobs queue. Then each time the total CMP workable jobs drops below two, the yard places an additional request for five workable CMP jobs.

In the simulations, the order quantity is set for each job type queue so that, on average, new requests are made every week. The reorder point is determined from a service level the yard chooses, where the service level is the probability that there is enough jobs in the workable jobs queue to meet new quotas during the lead time period (i.e., while waiting for new workable jobs to arrive). For each simulated day, quotas are randomly generated and met to the maximum extent possible from the workable jobs queue. The jobs are assigned to the five crews using the RAPT crew assignment model. Figure 8 shows the evolution of the workable jobs queue in one simulated month for a 50% service level. The net inventory level corresponds to the total number of workable jobs currently in the queue. When the net inventory is negative, then there is a backlog of workable jobs for that job type (i.e., there are not enough workable jobs to meet the quotas).

Table 2 summarizes the results of the simulation for different service levels. Note that increasing the service level increases the average size of the workable jobs queue. With 50% service level, the average inventory per day in the workable jobs queue is 28.8. However, seven quotas have not been met in time. Increasing the service level to 75% requires increasing the average inventory per day to 35.6, resulting in eliminating any backlogged jobs. Increasing the service level further to 90% or 99% results in higher average inventories

Figure 8 Workable Jobs Queue over One Simulated Month with 50% Service Level



Note. Each line is the net inventory of workable jobs for a job type in Table EC.17 of the electronic companion.

of workable jobs, but with essentially the same effect on backlogged jobs as the smaller service level 75%. A backlog of quotas also affects the daily crew utilization. If there is an infinite supply of workable jobs, crews will be working on average the same amount of hours. Having a finite supply of workable jobs means that crew utilization is variable from day to day (crews work less during backlogged days but work more in ensuing days). Note that with a 50% service level, on average a three-member crew is logging 10.8 hours more overtime per month than with the other service level choices.

We have embedded in RAPT an option to conduct an analysis of the workable queue inventory policy for the yard. This analysis includes a workable queue simulation and a summary of the service level's effect on crew utilization and on the yard's ability to meet quotas. A yard's resource planner can use RAPT to choose an appropriate policy for the yard.

6.2.2. Using Crew Productivity Data. At present, detailed crew productivity data are not available in the company's database. As such, it is not possible to make crew assignments to take advantage of the inherent job-specific productivity differences between crews in the crew assignment phase. We used simulations to aid company management in understanding the impact of using job-specific productivity data in crew assignment versus assigning jobs based on average productivity. This analysis was used to motivate upper management to provide the appropriate resources to keep track of crew productivity.

In the simulations, a yard has five crews that are each "experts" in one of the job types. We let $\gamma \in [0, 1)$ be an *expertise factor*, which is the percentage reduction in a job's duration if an expert works on it. Larger values of γ mean that experts are more productive relative to regular crews. In our experiments, we let $\gamma = 0\%$ (base case), 5%, and 10%. We run the simulation for 30 days. In each day, the work that has to be assigned is randomly generated. (The distribution of quotas is given in §EC.13 of the electronic companion.) We observe that the assignment model assigns most jobs to crews that have expertise in them. Table 3 shows the total expected overtime crew hours over a one-month period. By having expert crews who work

Table 2 Effect of Service Levels on Average Workable Jobs Inventory, Backlogged Jobs, and Overtime Crew Hours for One Simulated Month

Service level	50%	75%	90%	99%
Average inventory per day	28.8	35.6	37.6	50.6
Total backlogged jobs	7	0	0	0
Average overtime per day (crew hours)	14.67	14.55	14.55	14.55

Table 3 Total Expected Overtime Crew Hours for Different Expertise Factors

	Base case	$\gamma = 5\%$	$\gamma = 10\%$
Total expected overtime crew hours	340.4	329.4	302.6
% improvement over base case	—	3.23	11.1

Table 4 Simulation Results for Increasing Supervisor Presence in the Field

	Base case	5% reduction in job durations	10% reduction in job durations	25% reduction in job durations
Average overtime per day per crew (crew hours)	3.09	2.81	2.48	1.51
% improvement over base case	—	9.2	19.7	51.1

with 5% reduced durations, overall overtime hours can decrease by 3.23%. The decrease in overtime hours is nonlinear since if expert crews can work with 10% reduced durations, the total overtime hours in one month are reduced by as much as 11.1%.

6.2.3. Increasing Supervision over Crews. A prior study conducted by the company observed that crew productivity is directly related to field supervision. More time spent overseeing crews in the field results in more productive crews. We used simulations to validate and measure the appropriate level of supervision to maximize productivity since field supervision has a cost.

In these simulations, we compare the effect of having an increased supervisor presence in the field to the average expected overtime incurred by crews. Consider the work types given in §EC.13 of the electronic companion. Assume that by having increased supervisor presence, the durations of work types can be decreased. We will compare different cases: the base case (no reduction), 5% reduction, 10% reduction, and 25% reduction. We simulate a yard with five crews and with the daily quotas randomly generated based on Table EC.17. We assume that there is an infinite supply of permitted work (so the inventory policy is not a factor). Each day, we assign the work to the five crews using RAPT and note the total expected overtime incurred by the five crews during that day. For the different cases, we run this simulation for 30 days and calculate the total expected overtime averaged over 30 days.

Table 4 reports the result of the simulation for the different cases. We can infer that each 5% decrease in job duration (by increasing supervisor presence) results in a reduction of 1.6 overtime crew hours each day for the five-crew yard. Therefore, assuming that there are three members in a crew, a 5% increase in productivity results in reducing a total of 143 overtime hours charged for the yard in one month.

6.2.4. Projected Financial Impact from Changes. In our project, we used the previous analyses to compute the projected financial impact of implementing process changes in the utility company. We demonstrate this with a hypothetical utility that has an operating profit of $3.5 billion per year. The hypothetical utility employs 10,000 field personnel. The straight-time hours per person per year are 2,000, with an additional 500 overtime hours per person per year. The average wage of a field personnel is $50 per hour. Overtime is paid out at $75 per hour. The hypothetical utility spends $1 billion in straight-time labor costs (20 million hours), with an additional $375 million in overtime labor costs (5 million hours). We estimate the financial impact to this hypothetical utility of introducing the business process changes described earlier in this section. The percentage savings in overtime costs are all based on the analyses in §§6.2.2 and 6.2.3.

If the utility were to keep crew-specific productivity data as described in §6.2.2 (assuming expert crews are 5% more productive), we would anticipate annual savings of about $12 million, which represents 0.3% of the utility's annual operating profit. Suppose the company were to increase crew supervision as described in §6.2.3. Based on previous company studies, increased supervisor presence reduces job durations by at least 10%. This results in annual savings of $74 million (or 2% of the annual operating profit). If the company is able to implement both changes, this has a cumulative savings of about $84 million per year, which represents 2.4% of the annual operating profit.

7. Conclusions

In many industries, a common problem is how to allocate a limited set of resources to perform a specific set of tasks or jobs. However, sometimes these resources are also used to perform emergencies that randomly arrive in the future. For example, in hospitals, operating rooms are used both for elective surgeries (that are known in advance) and emergency surgeries (which need to be performed soon after they arrive). Another example, which motivated this paper, is scheduling maintenance crews in a gas utility company. Maintenance crews have to execute standard jobs (pipeline construction, pipe replacement, customer service) and respond to reports of emergency gas leaks. Emergencies arrive randomly throughout the day. With randomly arriving emergencies, the problem becomes more complicated since the resources need to be allocated before realizing the number of emergencies that have to be performed. Thus, a schedule needs to be flexible in that there must be resources available to perform these future emergencies.

We use stochastic optimization to model the problem faced by the gas utility. The problem is decomposed into two phases: a job scheduling phase and a crew assignment phase. The optimization problems resulting from each phase are computationally intractable, but we provide tractable heuristics for solving each of them. The job scheduling phase heuristic solves a mixed integer program, for which we propose an LP-based heuristic. We are able to prove a data-driven performance guarantee for this heuristic. The crew assignment phase solves a two-stage stochastic mixed integer program. Here, we propose an algorithm that replicates the structure of the optimal crew assignment. We demonstrate how the two heuristics can be implemented in a rolling horizon for rescheduling and reassignment in response to the state of emergencies.

We used our models and algorithms to improve job scheduling and crew assignment in the gas business of a large multistate utility company that faced significant uncertainty in its daily operations. Our models were also used to help the utility make strategic decisions about changes in its business and operations. In simulations using actual data and our models, we project the impact of different process changes to crew utilization and overtime labor costs.

7.1. Future Directions

There are several future directions that go beyond the scope of this paper and could be pursued.

In this paper, we focused on the job scheduling and crew assignment problems assuming that there is no travel time between two jobs. In the real-world application, this simplifying assumption makes sense because of the small distances between jobs. However, a future direction might be considering geography in making decisions. For instance, the job scheduling model can include a penalty proportional to the distance between jobs scheduled for the same day.

Another possible direction is to have emergencies with random durations. This is related to literature on scheduling under stochastic job durations, where jobs need to be processed on parallel machines without preemption. The number of jobs is known (unlike our setting), but the processing time of each job is an independent random variable. The objective is to minimize expected makespan (like our setting). It is known that the longest-expected-processing-time (LEPT) rule minimizes the expected makespan for exponential jobs or for remainders of i.i.d. decreasing hazard rate jobs (Pinedo and Weiss 1979, Weber 1982). In general, LEPT is a good but not optimal heuristic (Pinedo and Weiss 1979). For this reason, and based on preliminary experiments, we believe that Algorithm Stoch-LPT would perform well under the case where emergency durations are random.

Another potential direction is an analytical performance guarantee for the crew assignment heuristic, Stoch-LPT. We are able to prove that Stoch-LPT terminates with the optimal crew assignment under special cases. However, establishing a guarantee for the general case is an interesting direction.

We demonstrated the potential impact of the resource allocation planning tool in managing uncertainty in yard operations and decreasing labor costs. However, there is still some further work to be done for the yards to achieve these results. These include gaining grassroot support from the workers' union and continuing with strong management leadership. Some new processes need to be also introduced in all of the company's yards to ensure that the tool can be implemented successfully. The purpose of these new processes is to ensure integrity of the data fed into the model, and to create multiple levels of accountability for better oversight and cost control.

Acknowledgments

The authors thank Noah Gans (special issue coeditor), the associate editor, and the anonymous referees for their insightful comments, which have helped to significantly improve this paper. The authors also thank the anonymous utility that inspired and supported this work as well as the Leaders for Global Operations Program for their support. Georgia Perakis and Joline Uichanco thank Retsef Levi, Melvyn Sim, and Chung-Piaw Teo for insightful comments on this paper. Georgia Perakis also acknowledges the National Science Foundation [Grants CMMI-1162034, CMMI-0824674, and CMMI-0758061].

References

Ahmed S (2010) Two-stage stochastic integer programming: A brief introduction. Cochran JJ, Cox LA, Keskinocak P, Kharoufeh JP, Smith JC, eds. *Wiley Encyclopedia of Operations Research and Management Science*, Vol. 8 (John Wiley & Sons, New York).

Birge JR (1997) Stochastic programming computation and applications. *INFORMS J. Comput.* 9(2):111–133.

Burke G (2010) Aging gas pipe at risk of explosion nationwide. *AP News Archive* (September 14), http://www.apnewsarchive.com/2010/Aging-gas-pipe-at-risk-of-explosion-nationwide/id-d3e64ab3509a445285f27f4f81f73e4c.

Erdős P, Lovász L (1975) Problems and results on 3-chromatic hypergraphs and some related questions. Hajnal A, Rado R, Sós VT, eds. *Infinite and Finite Sets: To Paul Erdős on His 60th Birthday*, Vol. 2 (North-Holland, Amsterdam), 609–628.

Glass CA, Kellerer H (2007) Parallel machine scheduling with job assignment restrictions. *Naval Res. Logist.* 54(3):250–257.

Godfrey G, Powell WB (2002) An adaptive, dynamic programming algorithm for stochastic resource allocation problems, I: Single period travel times. *Transportation Sci.* 36(1):21–39.

Huh WT, Liu N, Truong V-A (2013) Multiresource allocation scheduling in dynamic environments. *Manufacturing Service Oper. Management* 15(2):280–291.

Hwang H-C, Chang SY, Lee K (2004) Parallel machine scheduling under a grade of service provision. *Comput. Oper. Res.* 31(12):2055–2061.

Kafura DG, Shen VY (1977) Task scheduling on a multiprocessor system with independent memories. *SIAM J. Comput.* 6(1):167–187.

Lamiri M, Xie X, Dolgui A, Grimaud F (2008) A stochastic model for operating room planning with elective and emergency demand for surgery. *Eur. J. Oper. Res.* 185(3):1026–1037.

Laporte G, Louveaux FV (1993) The integer *L*-shaped method for stochastic integer programs with complete recourse. *Oper. Res. Lett.* 13(3):133–142.

Lenstra JK, Shmoys DB, Tardos E (1990) Approximation algorithms for scheduling unrelated parallel machines. *Math. Programming* 46(1–3):259–271.

McDiarmid C (1989) On the method of bounded differences. *Surveys in Combinatorics* (Cambridge University Press, Cambridge, UK), 148–188.

Ou J, Leung JY-T, Li C-L (2008) Scheduling parallel machines with inclusive processing set restrictions. *Naval Res. Logist.* 55(4): 328–338.

Pinedo M, Weiss G (1979) Scheduling stochastic tasks on two parallel processors. *Naval Res. Logist. Quart.* 26(3): 527–536.

Pinedo ML (2002) *Scheduling: Theory, Algorithms, and Systems*, 2nd ed. (Prentice-Hall, Upper Saddle River, NJ).

Pipeline and Hazardous Materials Safety Administration (2011) Facts and stats—Pacific Gas and Electric pipeline rupture in San Bruno, CA. Accessed July 16, 2012, http://opsweb.phmsa.dot.gov/pipelineforum/facts-and-stats/recent-incidents/sanbruno-ca/.

Sherali HD, Fraticelli BMP (2002) A modification of Benders' decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse. *J. Global Optim.* 22(1–4):319–342.

Weber RR (1982) Scheduling jobs with stochastic processing requirements on parallel machines to minimize makespan or flow time. *J. Appl. Probab.* 19(1):167–182.