Submitted to *Management Science*
manuscript

Business analytics for flexible resource allocation under random emergencies

Mallik Angalakudati*, Siddharth Balwani†, Jorge Calzada*, Bikram Chatterjee*

Georgia Perakis‡, Nicolas Raad*, Joline Uichanco§

In this paper, we describe both applied and analytical work in collaboration with a large multi-state gas utility company. The project addressed a major operational resource allocation challenge that is typical to the industry and to other application domains. In particular, we developed analytical decision support tools to address 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 company has maintenance crews that perform both standard jobs (each must be done before a specified deadline) as well as repair 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 service 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 to replicate the optimal solution structure.

We used our models and heuristics to develop a web-based planning tool for the utility which is currently being piloted in one of the company's sites. Using the utility's data, we project that the tool will result in 55% reduction in overtime hours. This represents dramatic potential savings in annual labor cost for the company. Additionally, we demonstrate the financial impact of these new business processes on a hypothetical utility. Simulations reveal that creating new business processes can potentially reduce annual overtime hours by 22%.

---

* National Grid, 40 Sylvan Road, Waltham, MA

† MIT Leaders for Global Operations, Cambridge, MA

‡ MIT Sloan School of Management, Cambridge, MA

§ MIT Operations Research Center, Cambridge, MA

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 having some resources being over-utilized. 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 urgent 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 in the system (say, by having operating rooms on standby) to handle random arrivals of emergency cases.

This paper is motivated by a project we did in collaboration with a major electric and gas utility company. We worked on improving scheduling of services for the company's gas utility segment which faces a lot of uncertainty in its daily operations. In 2011, the Gas business segment of the company generated several billion dollars in revenue. The following is a brief description of natural gas transmission and distribution in the United States. Natural gas is either produced (in the US Gulf Coast, midcontinent, and other sources) or imported (from the Middle East or South America). Afterwards, it is delivered to US interstate pipelines to be transmitted across the US. Once it reaches a neighborhood, the gas is delivered by a local gas utility company, which owns and operates a network of gas pipelines used to deliver gas to the end customers. The gas utility involved in the project owns several of these local networks.

A large part of daily operations of the gas utility is the maintenance of the large gas pipeline network. 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 attend to any reports of gas leaks. In the US, more than 60% of the gas transmission pipes are 40 years old or older (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, which 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 8 people and destroyed 38 homes in the San Francisco suburb (Pipeline & Hazardous Materials Safety Administration 2011a). 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 that any member of the public can call to report any suspected gas leaks. It is the company's policy to attend to any such reports within twenty-four hours of receiving them. The key characteristic of emergency gas leak jobs is that they are unpredictable, they need to be attended 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 *service crews* to execute both types of jobs. These service crews often work on shifts of eight hours, but can also work on overtime if there are jobs left to be done. Any hours worked in excess of the crew's shift is billed as overtime, and costs between 1.5 to 2 times the regular hourly wage. The company has experienced significant overtime driven by both uncontrollable factors such as timing uncertainty related to emergency leaks, diverse and unknown site conditions and uncertainty in job complexity as well as controllable factors such as workforce management, scheduling processes and information systems. Service crews historically worked a significant proportion of their hours on overtime. An average crew member may work between 25% to 40% of his or her work hours on overtime pay.

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

1.1. Literature Review and Our Contributions

Our work makes contributions in several key areas. We outline our contributions and contrast them with previous work found in related literature.

Modeling and problem decomposition. The company needs to make decisions about a standard job's schedule (which date it will be worked on) and its crew assignment (which crew is assigned to work on the job) before the number of gas leaks are known. The objective is to minimize the maximum expected work hours of any crew on any day. We model this problem as a mixed integer program (MIP). However, solving this problem for realistic problem sizes is intractable for several reasons. Not only does the presence of stochasticity increase the dimension of the MIP, the presence of integrality constraints further complicates the problem. Therefore, we propose a *two-phase decomposition* which makes the problem more tractable. The first phase is a *job scheduling phase*, where standard jobs are scheduled so as to meet all the deadlines, but without overloading a single day with too much work (Section 4). This scheduling phase solves a mixed integer program, but its size is drastically smaller than the original 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 (Section 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 leak jobs, hence, the assignment problem on each day is solved as a two-stage stochastic MIP. This type of decomposition is similar to what is often done in airline planning problems (see for example Barnhart et al. 2003), which in practice are solved sequentially due to the problem size and complexity. Airlines usually first solve a schedule design problem, which determines the flights flown during different time periods. Then in the next step, they decide which aircraft to assign to each flight depending on the forecasted demand for the flight. Airline planning problems are solved through deterministic models which are intractable due to its problem size. In contrast, the models in our paper are stochastic in nature, adding a layer of modeling and computational difficulties.

LP-based heuristic for scheduling phase. 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, therefore several works in the literature propose heuristics for solving it approximately (see (Leung and Li 2008) for a survey of heuristics). Common heuristics are based on list scheduling, which sorts the jobs and schedules them one-by-one based on the list (Kafura and Shen 1977, Hwang et al. 2004). However, these list scheduling heuristics are for scheduling problems with *parallel* machines. In contrast, for our problem with *unrelated* machines with processing set restrictions, *we propose a heuristic for the job scheduling problem that is based on linear programming (LP) methodology.* Since this heuristic is based on linear programming, in practice it solves very fast for realistic problem.

Performance guarantee for the LP-based heuristic. We prove a relative performance guarantee for the LP-based heuristic of the scheduling problem. This performance guarantee is *data-dependent*, i.e., it depends on the job durations and other parameters. The proof uses Lovász's Local Lemma (Erdős and Lovász 1975, Srinivasan 1996), McDiarmid's inequality (McDiarmid 1989), as well as the technique of randomized rounding (Raghavan and Thompson 1987).

Algorithm for assignment under a stochastic number of urgent jobs. The assignment phase problem is a two-stage stochastic MIP, where in the first stage the assignment of standard jobs to crews is determined, and in the second stage (after the number of gas leaks is known) the assignment of gas leak jobs to crews is decided. Most literature on problems of this type develops 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 non-convex, 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, such solution methods are difficult to implement. Moreover, since they are developed for general two-stage stochastic problems, they do not give insights on how resources should be allocated in anticipation of an uncertain number of urgent jobs. In this paper, *we exploit the structure of the problem and of the optimal assignment and propose a simple algorithm for assigning the standard jobs under a stochastic number of emergencies* (Algorithm 3). This algorithm can be thought of as a generalization of the Longest-Processing-Time First (LPT) algorithm in the scheduling literature (Pinedo 2002). Our algorithm first assigns gas leak jobs for each scenario to mimic properties of the optimal assignment of gas leak jobs (Proposition 2). This assignment is made so that any crew who works on a given number of leaks in a scenario works on at least as many leaks in a scenario with more gas leaks. Then, the algorithm sorts the standard jobs in decreasing duration, and assigns the jobs one-by-one to the crew achieving a minimum index. For LPT this index is the current number of hours assigned. In contrast, our algorithm uses as an index the expected maximum hours (makespan) resulting from assigning to that crew. Computational results show that the algorithm produces assignments with expected makespan close to optimal.

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 are also applicable to other settings where resources need to be allocated in a flexible manner in order 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 jobs, respectively, in our problem.

Business analytics for a large US utility. We collaborated with a large multi-state utility company 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 emergency gas leaks. Due to the size of the problem, the company's need for fast solution methods led us to develop the job scheduling heuristic and the crew assignment heuristics described earlier. 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 with an annual labor cost of $1 billion. Simulations with our model demonstrate that the new business processes can potentially reduce annual overtime hours by 22.3%, resulting in a $84 million reduction in the hypothetical utility's annual labor cost.

1.2. Outline

In Section 3, we present the job scheduling and crew assignment problem, as well as motivate the two-stage decomposition. In Section 4, we introduce the job scheduling phase, introduce an LP-based heuristic, and prove the data-driven performance guarantee of the heuristic. Section 5 introduces the crew assignment phase. In this section, we prove a structural property of the optimal crew assignment, and propose an algorithm to perform assignment built on this property. In Section 6, we discuss how we used simulation and the models we developed for business analytics at the Gas business of a large multi-state utility company.

2. Gas utility operations and organization

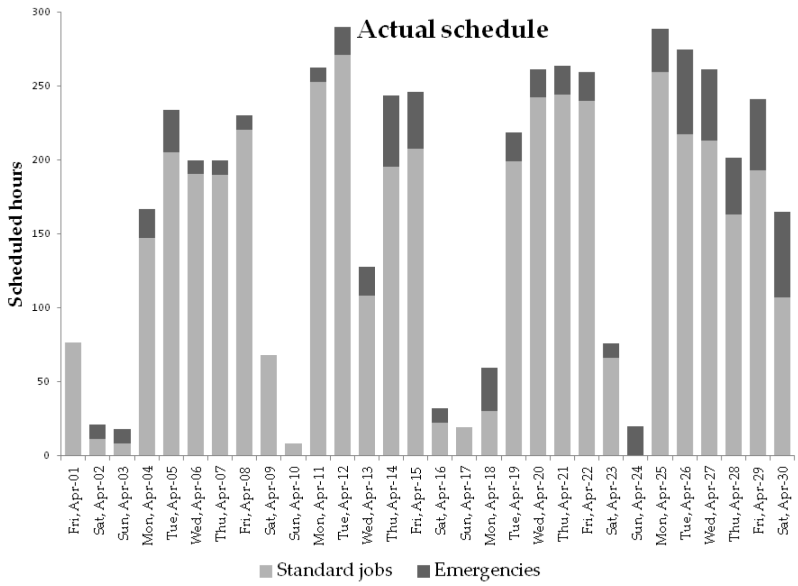
In this section, we give a background of the company operations and organization during which our project was conducted. This discussion serves the purpose of motivating the model, assumptions, and our choice of heuristics.

Gas utility companies in the US operate large networks of gas pipelines. Some of these pipelines are aging and are composed of corrosive material. Some gas pipes still in service in many cities today are composed of corrosive cast iron and have been installed since the 1830's (Pipeline & Hazardous Materials Safety Administration 2011b). Gas utility companies are often part of a government-mandated cast iron main replacement program which aims to replace many of the cast iron pipes into more durable steel or PVC pipes. To meet the requirements of this program, the company 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 city or several neighboring cities) are assigned to a "yard." A yard is the physical company site which houses service 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. Separate yards belonging to the company operate independently. Small yards can have ten crews, while large yards can have up to 30 crews, with each crew composed of two or three crew members. Before work on a standard job can be initiated, yards are required to apply for a permit with the city. These permits expire, therefore standard jobs need to be finished before the expiration of the permit. The company maintains a centralized database of standard jobs which lists each job's deadline, status (e.g. completed, pending or in progress), location, job type, 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.

Each yard has an employee called a resource planner who is charged with making decisions about the yard's daily operations. In particular, at the start of each day, the resource planner looks over the pending standard jobs and upcoming deadlines and decides which jobs should be done by the yard that day, and which crews should execute these jobs. After service crews have been dispatched to their assignments, the yard might receive reports of emergency gas leaks that also need to be handled by service crews. These emergency gas leaks are found by dedicated company crews (separate from the yard's service crews) regularly monitoring leak prone pipes to identify any leaks that need immediate attention. In compliance with government regulations, the company's policy is to dispatch a crew to an emergency gas leak within 24 hours of receiving the report. Leaks that do not pose significant risk to the public are fixed later within regulatory deadlines (usually in 12 months) dictated by the risk involved. These less severe leaks are in the category of standard jobs. Emergency gas leaks are highly unpredictable; a large yard can have between zero to six

Figure 1 **Current system: Actual crew-hours worked in April 2011 in an average-sized yard.**



emergencies per day. They also are long duration jobs since each leak requires around 8 hours to complete.

Service crews have eight hour shifts, but can work beyond their shifts if needed. Any hours worked in excess of the crew's shift is billed as overtime, and costs between 1.5 to 2 times the regular hourly wage. Based on data from the company's yards, crews in each yard have been working a significant proportion of their hours at overtime. An average crew member works 25% to 40% of his or her hours on overtime. Figure 1 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). From Figure 1, we observe that even without the randomness introduced by the emergency gas leaks, the hours spent working on standard jobs are unevenly divided among the workdays. We observed that one of the major causes of overtime is suboptimal job scheduling and planning for the occurrence of gas leaks. Currently, the company has no guidelines or does not use quantitative methods for job scheduling and crew assignment. Instead, resource planners depend on their experience and feedback from supervisors. The company does not currently measure and analyze crew productivity. This results in resource planners relying on subjective input from supervisors on crew assignment decisions. Also, resource planners do not provide slack capacity (i.e., idle crew hours) to attend to any gas leak jobs that might occur later in the day. The variability of emergency leaks 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 built a stochastic optimization model for multiperiod planning of yard operations. Particularly, under a random number of emergencies, the model we develop guides decisions about job scheduling (i.e, determining which date each standard job is done) and crew assignment (once scheduled on a date, determining which crew is assigned to complete the job). Later, we discuss a novel decomposition motivated by the practical limitations encountered during the project.

Before we discuss the model, we first state all the assumptions made in our model, which are motivated from the operations of a yard.

1. There is a deterministic number of crews per day.
2. Standard jobs cannot be preempted. To start a job requires the crew to block off a portion of a street and dig a sizable hole on the ground in order to access the gas pipe. Therefore, standard jobs are generally not paused arbitrarily before completion.
3. The standard job durations are deterministic. Standard job durations are predicted accurately by a few factors: the job type, the size and diameter of the pipe, the age of the pipe, and whether the job is on a main street or not. Based on historical job data, we built a regression model which predicts job durations based on job characteristics. We observe minimal variation between predicted values and actual values of job durations.
4. The number of emergencies per day is unknown (stochastic) at the start of a day. However, all emergencies have the same duration, and is significantly longer than standard job durations.
5. The day can be divided into two parts (pre-emergency and post-emergency). A simplifying assumption we make is that the specific time that an emergency arrives does not matter. This is because, to comply with regulation, a crew needs only to be dispatched to an emergency within 24 hours. Therefore, when crews are already working on standard jobs when an emergency arrives, the emergency does not have to be attended to until a crew finishes its current job.
6. Crew assignment does not take distances (geography) into consideration. This simplifying assumption is made since travel time between jobs is usually less compared to the duration of jobs.

We next derive our model for yard operations. Consider a set of standard jobs needed to be completed within a given time horizon (e.g. one month). Each 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 leaks may be reported. The number of leaks is only realized once the job schedule and crew assignments for that day have been made. In our model, we assume that the number of leaks per day is a random variable with a known distribution. The following is a summary of the notation used in our model.

T	length of planning horizon
K_t	set 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 $\tau_i \leq T$, where $i = 1, \ldots, n$
d_L	duration of each leak job
$L(\omega)$	number of leaks under scenario ω
Ω_t	(finite) set of all scenarios in day t, where $t = 1, \ldots, T$
$P_t(\cdot)$	probability distribution of scenarios on day t, $P_t : \Omega_t \mapsto [0, 1]$

There are two different types of decisions to be made: the *first stage decisions* (i.e., decisions that have to be made before the uncertainties are realized,) as well as the *second stage decisions* (i.e., decisions that can only be made after the uncertainties are realized). At the start of the planning horizon, the job schedule has to be decided. This is because the date in which a job is scheduled to be done must be known in advance for planning purposes. At the beginning of each day, the crew assignments need to be decided before the number of gas leaks is known. This is because the calls for gas leaks typically occur throughout the day, but the crews must be dispatched to their assigned standard jobs before these calls are received. Thus, in the context of the gas distribution company's problem, job schedules and crew assignments are first stage decisions. The second stage decisions are the number of gas leaks each crew has to attend to in each day.

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 scenario ω is realized on day t, let $Z_{tk}(\omega)$ be the second-stage decision variable denoting the number of leak jobs assigned to crew k. It clearly depends on the number of known 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 realization of the number of gas leaks, $L(\omega)$, the objective of the day t recourse problem is to choose an assignment of gas leak jobs, $Z_t(\omega) \triangleq (Z_{tk}(\omega))_k$, so as to minimize the *maximum number of hours worked over all crews.* Thus, the day t recourse problem is:

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

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

We chose a min-max objective instead of an objective of minimizing total overtime (work hours in excess of a crew's shift). Both objectives achieve similar monetary costs (where cost is proportional to the total overtime hours), but a min-max objective distributes overtime evenly over the crews. To illustrate this, consider a recourse problem with two emergencies (of 8 hour durations each), two standard jobs (of 8 hour durations each), and two service crews. Under an objective of minimizing overtime, an optimal solution is to assign one emergency to the first crew (8 hours), and assign the remaining work to the second crew (24 hours). Under a min-max objective, an optimal solution is to assign one emergency and one standard job to each of the crews (16 hours). Under both solutions, the total overtime is 16 hours. However, the overtime is shared by the two crews under a min-max objective.

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

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

The first stage 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 actual scheduling and assignment model we implement is slightly more complicated. For instance, there is a set of job "types", and each job can be one of these types. All jobs of the same type often have the same duration. There might be an additional constraint on meeting a minimum quota of jobs of the same type over the planning horizon. Another possible variation to the model might be having different job durations depending on the assigned crew. However, for ease of exposition and for making notation simpler to follow in this paper we ignore these constraints and focus on problem (2). Nevertheless, the analysis and results of this paper are very similar for the more complicated version of the model as well.

Optimization problem (2) can be rewritten as a mixed integer problem. Proposition 3 found in Appendix A gives the MIP formulation.

3.1. Practical limitations of the joint job scheduling and crew assignment problem

The job scheduling decisions and crew assignment decisions need to be made *jointly*. This is because the objective is a function of the crew assignments, therefore job scheduling decisions alone cannot determine the cost. However, there were several practical issues that prevented the implementation of the joint scheduling and assignment problem in yard operations which we discuss below.

Firstly, the full optimization problem (10) is intractable to solve for actual yard problems within a reasonable amount of time. For actual yard settings, a crew assignments need to be determined within at most several seconds. If there are no emergencies, the problem is known to be NP-hard (Pinedo 2002). Additionally, the presence of stochastic number of emergencies increases the problem dimension of the deterministic equivalent problem (10), since the size is multiplied by the number of scenarios.

Secondly, the yard employees and resource planners required transparency in how decisions are being made. There are general-purpose computational methods that solve stochastic optimization problems efficiently such as the integer L-shaped method (Laporte and Louveaux 1993), scenario decomposition (Carøe and Schultz 1999), or cutting plane approaches with sequential convexification of the second stage problem (Sherali and Fraticelli 2002). However, since yard decisions are traditionally being made by resource planners without guidance from any quantitative models or data, resource planners were naturally suspicious of "black box" decision models that do not give insights as to how decisions are being made.

Finally, due to issues about integration with the company's current databases and other technical issues, the company chose not to invest in a commercial integer programming solver for a implementation of the project throughout the whole company. Therefore, a limitation faced in the project was the fact that our models and heuristics 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 crudely 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 under a *stochastic number of emergencies*. The goal is to minimize the expected maximum hours worked by any crew.

The two-phase decomposition provides transparency in how decisions about job schedules and crew assignments are made. Moreover, the problem is more tractable due to the smaller problem dimensions. Sections 4–5 provide more details on two phases of the decomposition. However, both the job scheduling phase and the crew assignment phase are NP-hard problems. For the job scheduling phase, we propose a heuristic based on solving the LP-relaxation which, based on actual problem sizes, can be implemented in Excel Solver. For the crew assignment phase, we propose an intuitive sorting-based heuristic that is a stochastic variant of the Longest Processing Time First (LPT) rule.

4. Phase I: Job scheduling

In the job scheduling phase, standard jobs of varying durations have to be scheduled on a planning horizon. However, there is a stochastic number of emergencies per day, and the number of crews can change for different days. For instance, yards usually have less crews working during weekends compared to weekdays. Moreover, there are less company crews detecting gas leaks during weekends, so there are usually less 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:

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

The motivation behind scaling the average scheduled hours per day by the number of crews is so that the optimal solution will schedule less 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 gas leaks per day. Due to 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 following mixed integer program.*

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

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 only scheduled 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, solves 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.

In what follows, we introduce a heuristic for approximating the solution for the job scheduling problem (3). Similar to Lenstra et al. (1990), our 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. We are able provide a data-dependent performance guarantee for the heuristic (Theorem 1) which we derive using a novel technique based on *stochastic analysis*. If a binary search is performed before solving the LP relaxation, the resulting schedule has a makespan is no more than a factor of the optimal makespan, with the factor the minimum between 2 and a data-dependent expression (Theorem 2).

4.1. LP-based job scheduling heuristic

The details of the job scheduling algorithm is given in Appendix B (Algorithm 1). The idea behind the algorithm is to solve the linear programming relaxation of the scheduling phase MIP. The LP solution is rounded into a feasible job schedule by solving a smaller scale MIP.

Consider the LP relaxation of the scheduling phase MIP (4) where all constraints of the form $X_{it} \in \{0, 1\}$ are replaced by $X_{it} \geq 0$. Denote the solutions to the LP relaxation by C^{LP} and X^{LP}. The algorithm takes the LP solution and converts it into a feasible job schedule using a rounding procedure. The idea in the rounding step 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 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 (see Lenstra et al. (1990) for proof).

The following theorem states that the schedule resulting from the job scheduling algorithm (Algorithm 1) 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 1, then X^H is feasible for the scheduling phase problem* (4)*, and has an objective cost C^H such that*

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

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

The proof based on stochastic analysis can be found in Appendix C, but we outline its idea here. Introduce the randomized rounding $\tilde{X}$ by interpreting the LP solution X^{LP} as probability distributions such that $E(\tilde{X}_{it}) = X_{it}^{LP}$. Note that all realizations of $\tilde{X}$ are all the possible roundings of X^{LP}. Moreover, the algorithm produces the rounding X^H with the smallest cost (the maximum ratio of scheduled hours to number of crews). Define 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). To prove the theorem, we need to show that there is a *positive probability* that none of the bad events $B_1, B_2, \dots, B_T$ occur. Note that each bad event is mutually dependent on at most δ other bad events. Then if there exists a bound on $\Pr(B_t)$ for all t, we can use Lovász's Local Lemma (Erdős and Lovász 1975) to prove that the event that none of these "bad" occurs is strictly positive. Since B_t is the event that a function of independent random variables deviates from its mean, $\Pr(B_t)$ can be bounded using the large deviations result McDiarmid's inequality (McDiarmid 1989).

Lenstra et al. (1990) propose a job scheduling algorithm for unrelated machines which performs a binary search procedure before solving the LP relaxation of a MIP. In particular, let the makespan C a constant, and consider the LP relaxation of a restricted version of (4) where each job i can only be scheduled on a date t with $d_i/|K_t| \leq C$. Using binary search, they look for the smallest value of C for which the corresponding LP relaxation is feasible. Denote this value as C^B, and the solution to the corresponding LP relaxation as $X^{LP'}$. Using a particular rounding procedure, they round $X^{LP'}$ to a feasible job schedule, and they prove using graph theory that this rounding has a cost no more than $2C^{OPT}$.

In both randomly generated job scheduling problem instances as well as actual yard problems, we observe that the data-dependent bound (5) is typically less than 2. Moreover, if we perform the

Lenstra et al. (1990) binary search procedure and round the LP solution $X^{LP'}$ using our rounding procedure, the resulting job schedule has a cost no more than αC^{OPT}, where α is the minimum of 2 and a data-dependent expression.

THEOREM 2. *Let C^{OPT} be the optimal objective value of the scheduling phase problem* (4)*. If C^B is the result of the binary search in Algorithm 2 and $X^{H'}$ is the schedule. Then $X^{H'}$ is feasible for the scheduling phase problem* (4)*, and has a cost $C^{H'}$ such that*

$$\frac{C^{H'}}{C^{OPT}} \leq \min\left\{2,\ 1+\frac{1}{C^B}\left(\min_{t=1,\ldots,T}|K_t|\right)^{-1}\sqrt{\frac{1}{2}\left(\sum_{i=1}^{n} d_i^2\right)(1+\ln\delta)}\right\}, \tag{6}$$

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

4.2. Computational Experiments

We implemented the LP-based algorithm (without an initializing binary search step) to solve 100 randomly generated problem instances. The size of the problem instance is chosen so that the job scheduling problem solves to optimality within a reasonable amount of time. In each problem instance, there are 70 jobs to be scheduled (with durations randomly drawn between 0 to 8 hours), 7 days in the planning horizon, and 3 crews available each day. For simplicity, we assume all jobs are due on the last day. Table 1 shows results for ten of the 100 problem instances. On average (over all 100 instances), the schedule from Algorithm 1 achieves a maximum ratio of scheduled hours to number of crews that is 5.3% greater than the optimal maximum ratio. However, Algorithm 1 manages to drastically improve computational efficiency. Solving for the optimal schedule in (3) sometimes requires several hours. On the other hand, Algorithm 1 only takes a few seconds to solve.

We also implemented the algorithm on actual yard data for one month. During that month, there were 481 standard jobs with durations ranging between 3 hours to 9 hours. On weekdays, there were 20 crews per day, and emergency gas leaks ranged between 0 to 6 per day. On weekends, there were 5 crews per day, and emergency gas leaks ranged between 0 to 3 per day. Due to the size of the problem, the job scheduling MIP (4) implemented in Gurobi does not solve to optimality within a reasonable amount of time. 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.

5. Phase II: Crew Assignment

Once the job schedule has been fixed from the first phase, the assignment of crews to standard jobs can be done independently for each day. The basic problem of the crew assignment phase is

Table 1 **Maximum ratio of scheduled hours to number of crews under the optimal schedule and the schedule from Algorithm 1.**

Instance no.	Max hours/crew Optimal	Max hours/crew Algorithm 1	% Difference	Instance no.	Max hours/crew Optimal	Max hours/crew Algorithm 1	% Difference
1	9.64	9.99	3.6%	6	8.12	8.32	2.4%
2	8.64	9.15	5.9%	7	9.32	9.86	5.8%
3	9.88	10.24	3.6%	8	9.53	9.91	4.0%
4	8.64	8.99	4.1%	9	8.60	8.95	4.1%
5	8.90	9.39	5.5%	10	9.61	9.82	2.3%

determining how many crews to reserve for emergency gas leaks, and whether these emergency crews should perform any standard jobs, given the stochastic number of emergencies. If more crews are kept idle, then this might mean that there is more overtime resulting from non-emergency crews performing standard jobs. If less idle crews are kept for emergencies, then there is more overtime whenever there are many emergency gas leaks discovered.

In this section, we model the crew assignment phase as a stochastic mixed integer program which captures this fundamental tradeoff. Even without the presence of emergencies, the problem is NP-hard (Pinedo 2002). However, we prove a simple property of the optimal crew assignment. That is, there exists an optimal crew assignment in which the hours of emergencies assigned for a crew is increasing in the number of emergencies. Based on this property, we propose an algorithm which is a stochastic variant of the popular Longest Processing Time First (LPT) rule.

Denote by I_t the set of job indices that are scheduled for day t (as determined by the job scheduling phase). For each day t, the assignment phase assigns all standard jobs in I_t to available crews. However, these standard job assignments must be made before the number of leaks in day t is realized. Once the number of leaks is known, all gas leak jobs must be assigned to the available crews. The objective for each day t is to make the standard job assignments so that the expected maximum hours on day t is minimized.

For each day t, the assignment phase solves a *two-stage stochastic mixed integer program.* The first stage problem is:

$$\begin{aligned} \underset{Y}{\text{minimize}} \quad & E_t\left[F_t\left(Y, L(\omega)\right)\right] \\ \text{subject to} \quad & \sum_{k \in K_t} Y_{ik} = 1, \qquad i \in I_t, \\ & Y_{ik} \in \{0,1\}, \qquad i \in I_t,\ k \in K_t, \end{aligned} \tag{7}$$

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

$$\begin{aligned} F_t(Y, L(\omega)) \;\triangleq\; \underset{Z}{\text{minimize}} \quad & \max_{k \in K_t} \left\{ d_L Z_k + \sum_{i \in I_t} d_i Y_{ik} \right\} \\ \text{subject to} \quad & \sum_{k \in K_t} Z_k = L(\omega) \\ & Z_k \in \mathbb{Z}^+, \qquad k \in K_t. \end{aligned} \tag{8}$$

Note that the term in the brackets of the objective function is the number of hours assigned to crew k during scenario ω and under the standard job assignments Y. The assignment phase problem can also be rewritten as a mixed integer program (see Appendix F).

A limitation encountered in our project was that the two-stage stochastic program cannot be solved using IP solvers or computational techniques aimed for solving general stochastic optimization problems. This is due to the fact that resource planners, whose duties included making daily yard operations decisions, were resistant of "black box" decision models which did not give insight into how crew assignment decisions are being made. Motivated by this, we developed a crew assignment algorithm which exploits the specific structure of the assignment phase problem, which we will introduce later in Section 5.3. This algorithm is simple and intuitive since it can viewed as a stochastic variant of the Longest Processing Time First (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 emergency gas leak jobs.

5.1. Stochastic model compared to using averages

The stochasticity of the number of gas leaks increases the computational complexity of the problem. However, we now demonstrate why solving the two-stage stochastic optimization model (7) results in more robust assignments than a simple heuristic that ignores stochasticity. In particular, consider a heuristic which makes crew assignment decisions against the *average expected number of leaks*. This heuristic solves:

$$\begin{aligned} \underset{Y}{\text{minimize}} \quad & F_t\left(Y, E_t[L(\omega)]\right) \\ \text{subject to} \quad & \sum_{k \in K_t} Y_{ik} = 1, \qquad i \in I_t, \\ & Y_{ik} \in \{0,1\}, \qquad i \in I_t,\ k \in K_t, \end{aligned} \tag{9}$$

where F_t is defined in (8). We will refer to this heuristic as AVG, while the optimal assignment solved in (7) will be referred to as OPT.

In the following example, we compare the maximum hours for each gas leak scenario under crew assignments from the AVG heuristic and the OPT heuristic. Suppose that there are 7 crews available, and 15 standard jobs need to be assigned. The job durations are between 1 hour and 7 hours (see Table 3). The gas leak job duration is 8 hours. The probability of 0 leaks is 40%, the probability of 1 leak is 20%, and the probability of 2 leaks is 40%. Note that the average number of gas leaks is 1. We are interested in comparing the two heuristics with respect to the maximum work hours under the different leak scenarios.

Table 2 summarizes the results. Under the OPT assignment, all the crews work less than 11.83 hours even if there are 2 gas leaks. Under the AVG assignment, all crews work less than 10.66 hours if there are between 0 to 1 gas leaks. However, if there are 2 gas leaks, then the AVG assignment

Table 2 Maximum hours (over crews) of optimal assignment and assignment based on average number of leaks.

Scenario	Probability	OPT max hours	AVG max hours
0 leaks	0.4	11.76	10.66
1 leak	0.2	11.77	10.66
2 leaks	0.4	11.83	18.28
Expected maximum hours		11.79	13.70

Note: OPT is the optimal solution to (7). AVG is the solution to (9) which optimizes against the average number of leaks

Table 3 Durations of standard jobs.

Job no.	Duration	Job no.	Duration	Job no.	Duration
1	6.58	6	5.36	11	3.48
2	6.41	7	4.96	12	2.66
3	5.63	8	4.85	13	2.61
4	5.49	9	4.25	14	2.26
5	5.47	10	3.83	15	1.51

results in at least one crew working 18.28 hours. That is, there is a 40% probability that a crew under the AVG assignment will be working 18.28 hours. Since OPT results in a crew assignment where all crews work less than 11.83 hours on any leak scenario, it is more robust to stochasticity of gas leaks. These results agree with Birge (1997) who demonstrated that in many real-world applications stochastic optimization models are superior to their deterministic counterparts.

5.2. Structure of the optimal crew assignment

In this subsection, we conduct computational experiments on several examples in order to gain insight into the structure of the optimal crew assignment solution to (7). We find that the optimal crew assignment satisfies some structural properties, which we will exploit in order to propose a simple crew assignment heuristic.

Suppose for a given day, there are a total of 7 crews available. There are 15 standard jobs that need to be assigned to these crews, with durations given by Table 3. There is an average of one gas leak per day, where each gas leak job takes 8 hours. However, in our experiments, we will vary the probability distribution of the random number of gas leaks. In particular, we have seven computational experiments, each experiment using a different gas leak distribution in Table 4.

The optimal solutions to the seven numerical examples are given in the appendix (Tables 13–19). The highlighted cells in each table means that the corresponding crew (column) is assigned to work on a gas leak job during the corresponding leak scenario (row). Based on these results, an observation we can make is that in the optimal solution, if a crew is assigned to work on a gas leak in a given leak scenario, that crew should also be assigned to work on a gas leak in a scenario

Table 4 **Probability distributions of number of gas leaks.**

	Leak scenario					
	0 leaks	1 leak	2 leaks	3 leaks	E[no. leaks]	Stdev[no. leaks]
Leak distribution 1	0.0	1.0	0.0	0.0	1.0	0.00
Leak distribution 2	0.1	0.8	0.1	0.0	1.0	0.45
Leak distribution 3	0.2	0.6	0.2	0.0	1.0	0.63
Leak distribution 4	0.4	0.2	0.4	0.0	1.0	0.89
Leak distribution 5	0.3	0.5	0.1	0.1	1.0	0.89
Leak distribution 6	0.4	0.3	0.2	0.1	1.0	1.00
Leak distribution 7	0.5	0.2	0.1	0.2	1.0	1.18

where there are more gas leaks. This is formalized in the following proposition. The proof is in the appendix.

PROPOSITION 2. *There exists an optimal solution* $(Y^*, Z^*(\omega), \omega \in \Omega)$ *to the stochastic assignment problem* (7) *with the property that if* $L(\omega_1) < L(\omega_2)$ *for some* $\omega_1, \omega_2 \in \Omega$*, then for all* $k \in K$*,* $Z_k^*(\omega_1) \leq Z_k^*(\omega_2)$*.*

Another property of the optimal assignment we observe is that the jobs that have short durations are often assigned to the crews that handle gas leak jobs. Jobs with the longest durations are assigned to crews that work exclusively on standard jobs. That is, *in anticipation of a random number of urgent jobs, it is optimal to keep some crews idle or only assign them short duration jobs*. Based on this observation, and the monotonicity property of Proposition 2, we propose a stochastic assignment heuristic.

5.3. Assignment heuristic

Before introducing the assignment phase heuristic, recall that for the scheduling phase, we proposed a heuristic (Algorithm 1) based on solving the linear programming relaxation of the scheduling problem (12). The advantage of this heuristic compared to other sorting-based heuristics such as LG-LPT, is that it is easy to implement in optimization solvers such as Gurobi and CPLEX. In contrast, we now demonstrate through an example why *it is inappropriate to solve the LP relaxation of the assignment problem*, when there is a stochastic number of emergencies arriving in the future.

Suppose that the LP relaxation of the assignment problem (22) is solved for one of the examples in the previous subsection (for leak distribution 4). The LP relaxation assumes that all jobs can be arbitrarily subdivided among several crews. Table 5 summarizes the optimal solution of the LP relaxation. From the table, the optimal LP solution evenly divides the gas leak hours among all the available crews in all leak scenarios. Hence, when there is one gas leak, all crews are assigned 1.14 gas leak hours. When there are two gas leaks, all crews are assigned 2.29 gas leak hours. The stochasticity of the number of gas leaks does not have any adverse effects on the first stage problem, since any gas leak hours can be borne by all crews. Hence, the optimal LP solution then ignores the

Table 5 Solution of LP relaxation of (22) with leak distribution 4.

		Crew 1	Crew 2	Crew 3	Crew 4	Crew 5	Crew 6	Crew 7
Standard jobs	Total hours	9.3	9.3	9.3	9.3	9.3	9.3	9.3
	Total hours *(0 leak, $p = 0.4$)*	0	0	0	0	0	0	0
Gas leak jobs	Total hours *(1 leak, $p = 0.2$)*	1.14	1.14	1.14	1.14	1.14	1.14	1.14
	Total hours *(2 leaks, $p = 0.4$)*	2.29	2.29	2.29	2.29	2.29	2.29	2.29

effects of the second stage uncertainty and assigns the standard jobs to crews as evenly as possible. As discussed in Section 5.2, this solution is suboptimal since, if the jobs can't be subdivided, the optimal crew assignment is to keep some crews dedicated to gas leak emergencies (compare with optimal crew assignment in Table 16 in the appendix).

Let us now describe the proposed assignment heuristic. This heuristic is a modification of the Longest-Processing-Time First (LPT) algorithm, to account for the presence of an uncertain number of gas leak jobs. Recall that LPT is applicable when there is no uncertainty in the number of gas leak jobs, and it assigns standard jobs to crews by first sorting the jobs by decreasing duration, and then one-by-one assigning the job to the crew with the smallest current load. In the proposed algorithm, gas leak jobs are first assigned for each leak scenario. Then, the algorithm sorts the standard jobs in decreasing order of duration. However, *unlike LPT*, the next job in the list is *not* assigned to the crew with the smallest current load (or expected load). Instead, it determines the increase in expected maximum hours by assigning to each crew. The job is assigned to the crew where this increase is the smallest. If there are any ties, the job is assigned to the crew with the smallest expected current load. The complete description of the algorithm is given in the appendix (Algorithm 3). Note that when the number of gas leak jobs is deterministic, then this procedure is equivalent to LPT.

In the algorithm, the procedure for assigning the gas leak jobs for each scenario preserves the monotonicity property described in Proposition 2. It first sorts the gas leak scenarios in decreasing number of gas leaks. Starting with the first scenario, assign the gas leak jobs to crews using LPT. Now for the following scenarios, the gas leak jobs are also assigned to crews by LPT. However, in case of ties (where more than one crew has the minimum current load), choose a crew whose current load is strictly smaller than the load in the preceding scenario. This is done until all leak jobs for all scenarios have been assigned.

Using Algorithm 3, we assign jobs to crews for the seven numerical examples described in the previous section. Table 6 compares expected maximum hours worked for the optimal crew assignment and the crew assignment resulting from Algorithm 3. Note that in the deterministic setting (Leak distribution 1), Algorithm 3 is equivalent to the LPT algorithm. Therefore, the worst-case bound of $\frac{5}{6}$ for the difference applies, although the actual difference is much smaller (7.96%). Moreover,

Table 6 **Expected maximum hours worked under the optimal crew assignment and the assignment from Algorithm 3.**

	Expected maximum hours		
	Optimal	Algorithm 3	% Difference
Leak distribution 1	10.66	11.50	7.96%
Leak distribution 2	11.41	12.06	5.72%
Leak distribution 3	11.78	12.75	8.22%
Leak distribution 4	11.79	12.67	7.50%
Leak distribution 5	12.18	12.19	0.11%
Leak distribution 6	12.50	12.95	3.62%
Leak distribution 7	12.85	13.34	3.79%

under all different gas leak distributions, Algorithm 3 results in expected maximum hours no more than 8.25% of the optimal. Recall that Table 4 summarizes the mean and standard deviation of number of leaks for the 7 distributions. Based on Table 6, it appears that Algorithm 3 results in a crew assignment that is closer to optimal when the distribution of gas leaks has a higher variance.

6. Business analytics for the Gas business of a large multi-state utility

In this section, we describe how the research above applies to the scheduling of operations at the Gas business of a large multi-state utility. This is based on a joint project between the research team and the company that gave rise to the results of this paper. One of the company's primary operations is the maintenance of a large network of gas pipeline. It keeps a roster of service crews who have two types of tasks: to execute standard jobs by their deadlines, and to respond to gas leak emergencies. 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. We will show that the use of the model coupled with key process changes could help the company achieve significantly reduced costs and better ability to meet deadlines.

6.1. Process mapping and data gathering

In order to understand sources of inefficiency of yard operations, we first set out to map in detail the existing yard processes. To do this, we gathered data about the yard operations in multiple ways. 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. We also did extensive job shadowing of crews from multiple yards performing different types of jobs, and documented the range of processes followed. Additionally, we also constructed historical job schedules based on data gathered from the company's job database (see Figure 1 for a historical schedule for a yard's one month operations).

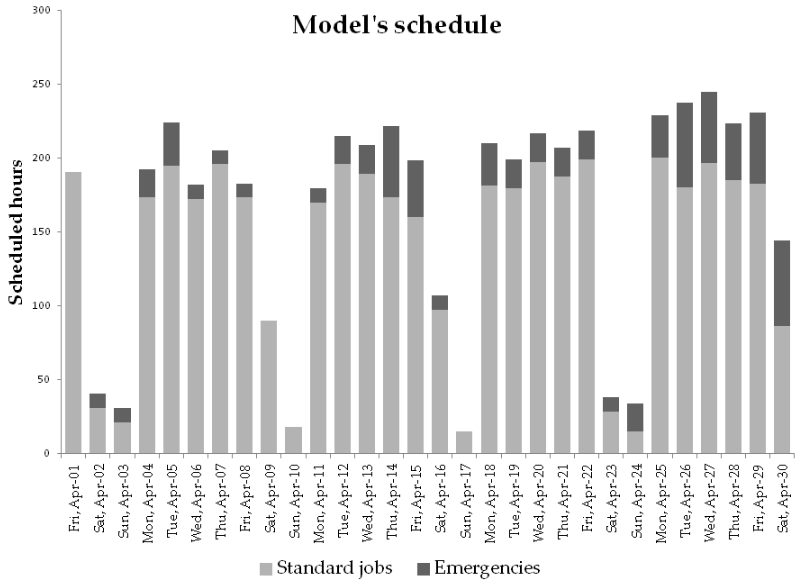
Gathered data suggest that there is insufficient capacity reserved for any emergency gas leak jobs that might occur later in the day. We also observed that availability of standard jobs fluctuates widely based on upstream processes of work order generation and permitting. The combination of the uneven flow of standard jobs and the variability of emergency leaks put resource planners in a reactive mode resulting in a short planning horizon and suboptimal resource utilization. In addition, the company does not currently measure and analyze crew productivity. This results in resource planners relying on subjective input from supervisors on crew assignment decisions. Furthermore, increasing the proportion of administrative work in the supervisors' time has reduced time available for monitoring crew performance in the field. Past studies conducted by the company indicated the actual amount of time spent by supervisors in the field has significant impact on crew productivity, and hence, overtime.

6.2. Improving operations

Our project with the utility company 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 described in this paper, we created a tool – the Resource Allocation and Planning Tool (RAPT) – to optimally schedule jobs and to assign them to crews while providing flexibility for sudden arrival of gas leak emergencies. One of the major components of RAPT is the job scheduling and crew assignment models that we have described in the previous sections. For the tool to be practical, we ensured that it: (i) is simple to use, (ii) is straightforward to maintain, (iii) uses popular software solutions, (iv) interfaces with the company's multiple databases, and (v) is modular such that any changes can be made without too much difficulty. The tool uses a web-based graphical user interface that is easy to use by resource planners. The back-end code was created based on Python and runs on a Windows-Apache-Oracle-Python stack on the company's servers. It accesses the jobs database and the time-sheet database, and uses this historical information to estimate leak distributions and job durations. These estimates and the list of all pending jobs are fed into the optimization models. RAPT outputs onto a webpage the weekly schedule for each crew and detailed plans under different gas leak scenarios.

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 was either missing, inappropriately gathered or not vetted before entry into the system. Having missing or inaccurate data makes it very difficult to apply a data-driven tool such as RAPT and makes it even more difficult to address the right issues. 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. Prior to the project, certain job types were not entered into the database and were instead tracked on paper at each yard. The new processes fix these issues and add this information to the database. Resource planners, supervisors and crew members will be trained to familiarize them with the new process flows prior to implementation.

Figure 2 Hypothetical scenario: Crew-hours worked if optimization model is used to schedule jobs.

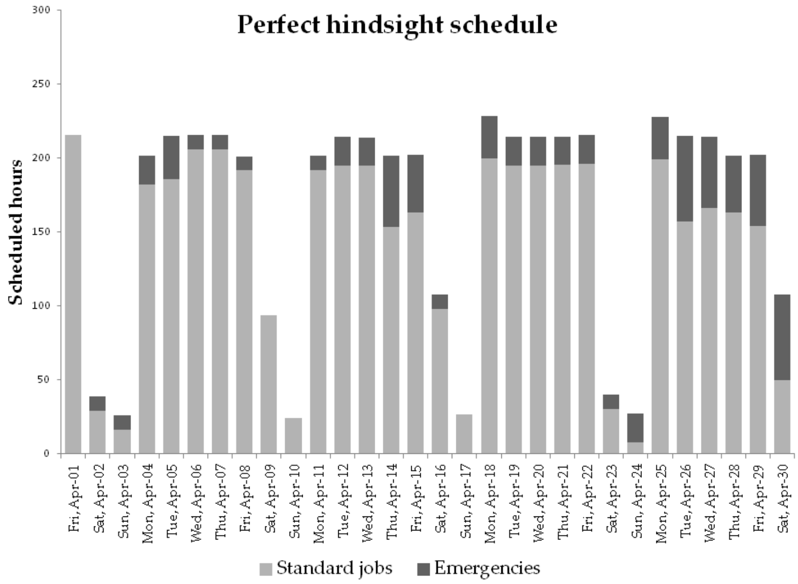


The third objective was to analyze the impact of key process and management drivers (business analytics) on operating costs and the ability to meet deadlines using the optimization model we developed. Results from this analysis will help the company deploy the optimization model with all the necessary process and management changes in order to capture the potential benefits outlined in this paper. The key process and management drivers selected for the study are work queues of available jobs for scheduling, availability of detailed productivity data (down to crew level) and supervisor presence in the field.

6.3. Business Analytics

Recall Figure 1 which shows the actual one-month profile of work hours in an average-sized yard. Figure 2 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 is superior to those produced previously by the resource planner. However, even if compared to the best possible schedule where all uncertainty is removed, the decisions produced by RAPT compare favorably. Figure 3 shows the “perfect hindsight” schedule. That is, it makes scheduling and assignment decisions based on complete knowledge of the realizations of emergencies that occur in the month. Clearly, the “perfect hindsight” decisions result in the maximum possible reduction in overtime since the yard can plan completely for emergencies. However, 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”.

Figure 3 **Hypothetical scenario: Crew-hours worked if optimization model is used to schedule jobs.**



However, this 55% decrease in overtime assumes favorable conditions for the model—for example, that all the jobs are ready to be worked on at any date. In reality, reaping the benefit of the model depends on the business processes currently established in the utility. Therefore, this reduction is most likely not achieved without first changing the company's current processes. Using our models from this paper, we conducted a study to understand the impact of establishing new processes on yard productivity. Based on past studies the company had conducted, the company understands that yard productivity is a complex phenomenon driven by process settings such as the size of work queues (i.e., jobs available for scheduling), effective supervision, incentives, and cultural factors. The research team and the company agreed to analyze three specific drivers of productivity using RAPT: work queue level, use of crew-specific productivity data, and the degree of field supervision.

6.3.1. Optimal Work Queue Level. In reality, jobs need to be in a "workable" state before crews can actually perform them. For example, the company needs to apply for a permit with the city before actually working on the job. Jobs in a "workable jobs queue" are jobs ready to be scheduled by RAPT. The reason why a queue is maintained is because "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). The larger the job queue, the larger the opportunity of the tool to optimize job scheduling and crew assignment over the planning horizon. During the project, we observed some yards kept a low level of jobs in the workable jobs queue. The low workable jobs queue adversely impacted the RAPT output by not fully utilizing the tool's potential. The team decided to run simulations to determine the strategic target level for the workable jobs queue to maximize the impact of RAPT while minimizing the efforts to sustain the workable jobs queue level.

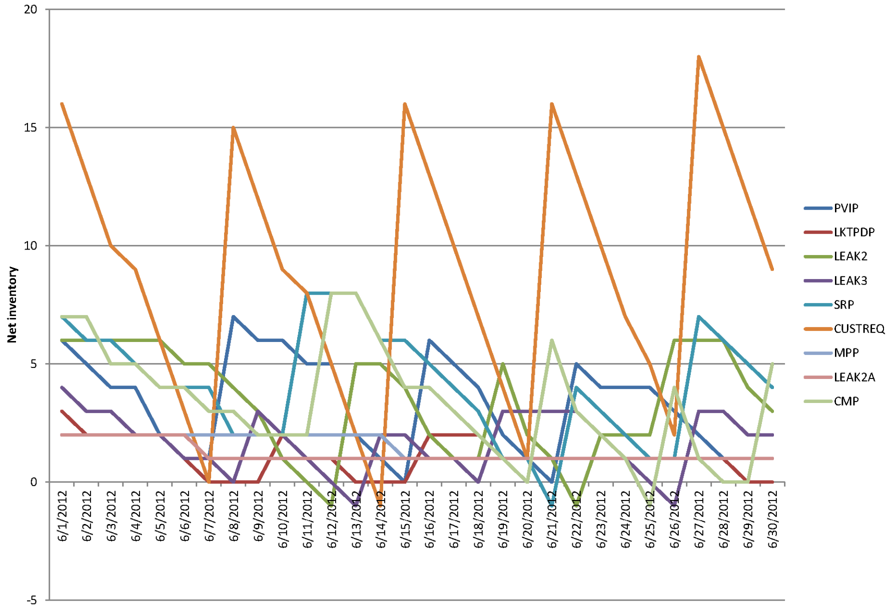
For our simulations we used actual data based on one of the company's yard. Each day, there are five crews available in the simulated yard each with 8 hour shifts. There are ten different job types to be done (refer to Table 7). The job type LKEMER refers to emergency gas leaks that are stochastic in our model. The table also gives the average job duration for each job type. On each day, the Resource Management department announces a minimum quota of the number of jobs required to be done for each 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 7). To be able to meet the daily quota requirements, the yard maintains a workable jobs queue for each job type. As an example, suppose that today the quota for CMP jobs is 10. However, there are only 6 jobs in the workable CMP job queue. Then, today, the yard will work on 6 CMP jobs, and will carry over the remaining 4 CMP jobs as a backlog for the next day. Each time the yard requests for new workable jobs to be added to the queue, there is a lead time of 3 days before the request arrives. For instance, this lead time may include time used for administrative work to apply for a permit.

Table 7 Data for job types used for simulations.

Job Type	Average job duration (hours)	Probability of daily quota				Average daily quota (no. jobs)	Stdev daily quota (no. jobs)
		0	1	2	3		
LEAK2A	6.24	0.93	0.07	0	0	0.07	0.25
CMP	5.85	0.45	0.4	0.12	0.03	0.71	0.79
PVIP	2.50	0.45	0.4	0.12	0.03	0.71	0.79
LKTPDP	5.45	0.73	0.28	0	0	0.28	0.45
LEAK2	6.76	0.45	0.4	0.12	0.03	0.71	0.79
LEAK3	7.80	0.46	0.54	0	0	0.54	0.5
CUSTREQ	6.24	0.05	0.1	0.1	0.75	2.86	0.86
SRP	6.34	0.45	0.4	0.12	0.03	0.71	0.79
MPP	3.87	0.89	0.11	0	0	0.11	0.31
LKEMER	8.79	—	—	—	—	—	—

The yard adapts a continuous review policy for the workable jobs queue 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 3 days. As an example, suppose the yard chooses a reorder point of 2 and an order quantity of 10 for the CMP workable jobs queue. Then, each time the total CMP workable jobs drops below 2, then the yard places an additional request for 10 workable CMP jobs. In our simulations, the

Figure 4 Workable jobs queue over one simulated month with 50% service level.



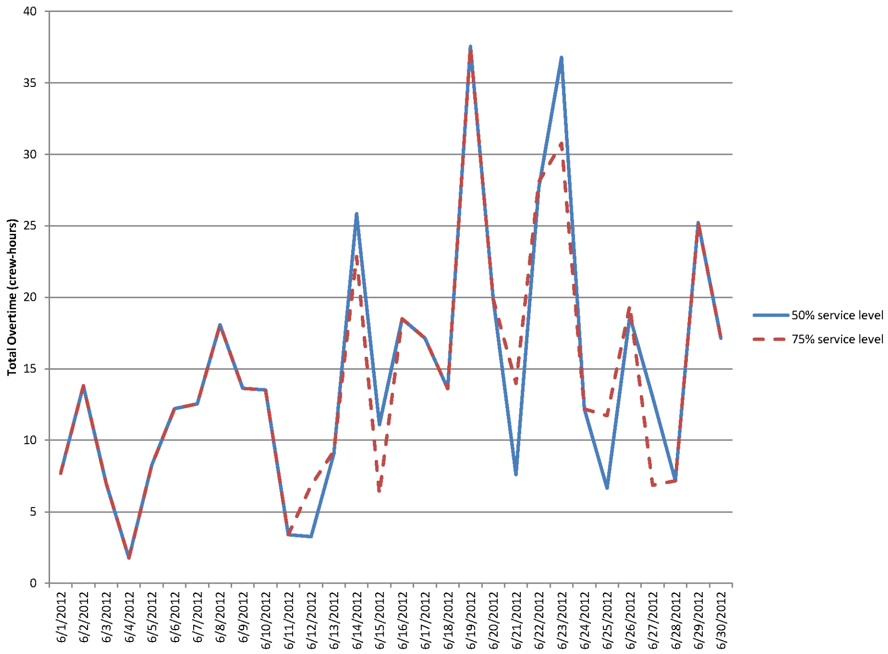
order quantity is set for each job type queue so that, on average, new requests are made every week. The reorder point is determined from a service level the yard chooses, where the service level is the probability that there is enough jobs in the workable jobs queue to meet new quotas during the lead time period (i.e., while waiting for new workable jobs to arrive). For each simulated day, quotas are generated based on Table 7 and met to the maximum extent possible from the workable jobs queue. The jobs are assigned to the 5 crews using the RAPT crew assignment model. In our simulations, we will determine the effect of the service level α on: (i) the average number of workable jobs kept in the queues, (ii) the number of backlogged quotas, and (iii) the day-to-day crew utilization.

Figure 4 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 8 summarizes the results of the simulation for different service levels. Note that increasing the service level increases the average size of the workable jobs queues, resulting in a smaller probability of backlogged jobs. With 50% service level, the average inventory per day in the workable jobs queue is 28.8. However, a total of 7 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 of workable jobs, but with essentially the same effect on backlogged jobs as the smaller service level 75%.

Table 8 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
Standard deviation overtime per day (crew-hours)	8.92	8.26	8.26	8.26

Figure 5 Yard overtime crew-hours over one simulated month.



In fact, having a backlog of quotas also has an effect on the day-to-day crew utilization. For example, in Figure 4 lack of CUSTREQ workable jobs on June 14 meant that crews were under-utilized on that day. However, after workable jobs arrive on June 15, not only do crews have to work on quotas meant for June 15, they also need to be working on the jobs to meet the missed quotas for June 14. In the perfect setting of an infinite supply of workable jobs, crews will be working on average the same amount of hours. Having finite supply of workable jobs means that crew utilization is variable from day-to-day. Figure 5 shows the total expected overtime in one simulated month. Notice that for a 50% service level, the “peaks” and “valleys” of overtime hours are more pronounced. It is important to note that this fluctuation is artificial since it is caused by the inventory policy for the workable jobs queue. This is obvious when comparing it to the overtime hours with a 75% service level, where crews are utilized at a more even rate. The plots for overtime hours under a 90% and 99% service level is the same as under a 75% service since they all result in no backlogged jobs.

6.3.2. Appropriateness of Productivity Data. Presently detailed crew productivity is not available. 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. In consultation with company management, we decided to study the impact of capturing and using job-specific productivity data in crew assignment versus assigning jobs based on average productivity.

Table 9 **Job type and crew expertise.**

	CREW1	CREW2	CREW3	CREW4	CREW5
Expertise	LKEMER	CMP	PVIP	LEAK2	SRP

In the simulation, we assume that each of the five crews are "experts" in one of the job types. Table 9 shows the particular expertise of the five crews. When a crew is an expert in a job type, then it will finish each job of that type in a shorter period of time. We let $\gamma \in [0, 1)$ be an *expertise factor* which denotes by how much the duration is reduced when a job has an expert working on it. 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 from the distribution of quotas given in Table 7. Based on the results of the simulation, we observe that when there are experts, the assignment model assigns most jobs to crews that have expertise in them. For instance, most of the gas leak emergency jobs are assigned to CREW1. Table 10 shows the total expected overtime crew-hours over a one month period. By having expert crews who work 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%.

Table 10 **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%

6.3.3. Impact on cost due to level of supervision. A prior company study had observed that crew productivity was directly related to field supervision. The more time the supervisors spent in the field supervising crews, the more productive the crews were. The team used RAPT to validate and measure the appropriate level of supervision to maximize productivity given that supervision has a cost.

In these simulations, we compare the effect of having an increased supervisor presence in the field on the average expected overtime incurred by crews. Consider the nine work types and their durations given in Table 7. 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. Suppose that there are 5 crews, and the daily quotas are randomly generated based on Table 7. Unlike the previous simulations, 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 5 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 11 summarizes the result of the simulation for the different cases. Clearly, having more productive crews will result in a decrease in overtime. Based on the table, we can infer that each 5% decrease in job durations (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 3 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.

Table 11 **Simulation results for increasing supervisor presence in the field.**

	Base case	5% reduction in job duration	10% reduction in job duration	25% reduction in job duration
Average overtime per day (crew-hours)	15.46	14.04	12.41	7.56
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%

6.4. Projected financial impact

We illustrate the projected financial impact of implementing RAPT and the process changes in a hypothetical utility. The hypothetical utility employs 10,000 field personnel. The straight-time hours per person per year are 2,000, with an additional 500 of overtime hours per person per year. The average wage of a field personnel is $50 per hour. Overtime is paid out at 1.5 times the straight-time wage ($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). Table 12 summarizes the projected financial impact to this hypothetical utility of introducing the business process changes described earlier in this section.

If the utility were to provide crew-specific productivity data as described in Section 6.3.2, using the 5% expertise factor, we would anticipate annual reductions in overtime of 3.23%. This would represent savings of about $12 million.

Suppose the company were to increase supervisor presence in the field as described in Section 6.3.3. Based on previous company studies, increased supervisor presence results in at least a 10% reduction in job duration. This would represent a overtime hour reduction by 19.7%, or $74 million savings per year.

Now, if the company is able to implement both changes, this has a cumulative effect of reducing annual overtime costs by 22.3%. For the hypothetical utility, this would represent savings of about $84 million per year.

Table 12 Projected financial impact of business process changes in a hypothetical utility.

	Base case	Have expert crews	Increase supervisor presence	Have expert crews and increase supervisor presence
Annual overtime hours	5 million	4.84 million	4.02 million	3.89 million
Annual overtime labor cost	$375 million	$363 million	$301 million	$291 million
% Savings in overtime labor cost	—	3.23%	19.7%	22.3%
Savings in overtime labor cost	—	$12 million	$74 million	$84 million

Assumptions: (i) expert crews are 5% more productive, (ii) increased supervisor presence results in a 10% increase in productivity for field personnel.

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. Some examples include bandwidth allocation, hospital scheduling, air traffic management, and shipping. However, sometimes these resources are also used to perform urgent jobs 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 crews in a gas utility company. Service crews have to perform both standard jobs (pipeline construction, pipe replacement, customer service) as well as gas leak repair jobs. The second type of jobs arrive randomly, but have to be worked on as soon as they arrive. When urgent jobs randomly arrive, the problem becomes more complicated since the resources need to be allocated before realizing the number of urgent jobs that have to be performed. Thus, a schedule needs to be flexible in that there must be resources available to perform these future emergencies.

Stochastic optimization models are useful for solving for the optimal schedule under a stochastic number of urgent jobs. However, in most real-life problems, these stochastic models are intractable since they deal with a large number of integer variables and constraints. Therefore, in practice, instead of solving a stochastic problem optimally, decomposition techniques and heuristics have proved valuable in finding a near-optimal solution but with significantly shorter running time.

We use stochastic optimization to model the problem faced by the gas utility company. We decompose the problem into two phases: a job scheduling phase and a crew assignment phase. The optimization problems resulting from each phase are NP-hard, therefore, we provide 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 which replicates the structure of the optimal crew assignment.

We used our models and algorithms to improve job scheduling and crew assignment in the Gas business of a large multi-state utility company which 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. In particular, we studied three different process changes: (i) maintaining an optimal work queue level, (ii) having detailed 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 whose labor costs amount to $ 1 billion per year. Simulations with our model demonstrate that the new business processes can potentially reduce annual overtime hours by 22.3%, resulting in a $84 million savings in annual labor costs.

References

Barnhart, C., A. M. Cohn, E. L. Johnson, D. Klabjan, G. L. Nemhauser, P. H. Vance. 2003. Airline crew scheduling. Randolph W. Hall, ed., *Handbook of Transportation Science*, 2nd ed. Kluwer Academic Publishers, Norwell, MA, 517–560.

Birge, J. R. 1997. Stochastic programming computation and applications. *INFORMS Journal on Computing* **9**(2) 111–133.

Burke, Garance. 2010. Aging gas pipe at risk of explosion nationwide. *Associated Press*. URL `http://www.apnewsarchive.com/2010/Aging-gas-pipe-at-risk-of-explosion-nationwide/id-d3e64ab3509a445285f27f4f81f73e4c`. Online; accessed 17-July-2012.

Carøe, C.C., R. Schultz. 1999. Dual decomposition in stochastic integer programming. *Operations Research Letters* **24** 37–45.

Erdős, P., L. Lovász. 1975. Problems and results on 3-chromatic hypergraphs and some related questions. A. Hajnal, R. Rado, V. T. Sós, eds., *Infinite and Finite Sets (to Paul Erdős on his 60th birthday)*, Volume 2. North Holland, 609–628.

Glass, C. A., H. Kellerer. 2007. Parallel machine scheduling with job assignment restrictions. *Naval Research Logistics* **54**(3) 250–257.

Godfrey, G., W. B. Powell. 2002. An adaptive, dynamic programming algorithm for stochastic resource allocation problems, i: single period travel times. *Transportation Science* **36**(1) 21–39.

Hwang, H.-C., S. Y. Chang, K. Lee. 2004. Parallel machine scheduling under a grade of service provision. *Computers and Operations Research* **31**(12) 2055–2061.

Kafura, D. G., V. Y. Shen. 1977. Task scheduling on a multiprocessor system with independent memories. *SIAM Journal on Computing* **6**(1) 167–187.

Lamiri, M., X. Xie, A. Dolgui, F. Grimaud. 2008. A stochastic model for operating room planning with elective and emergency demand for surgery. *European Journal of Operational Research* **185**(3) 1026–1037.

Laporte, G., F. V. Louveaux. 1993. The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters* **13**(3) 133–142.

Lenstra, J. K., D. B. Shmoys, E. Tardos. 1990. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming* **46** 259–271.

Leung, J. Y.-T., C.-L. Li. 2008. Scheduling with processing set restrictions: A survey. *International Journal of Production Economics* **116** 251–262.

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

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

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

Pipeline & Hazardous Materials Safety Administration. 2011a. Facts & stats – Pacific Gas & Electric pipeline rupture in San Bruno, CA. URL `http://opsweb.phmsa.dot.gov/pipelineforum/facts-and-stats/recent-incidents/sanbruno-ca/`. Online; accessed 16-July-2012.

Pipeline & Hazardous Materials Safety Administration. 2011b. Reports & research – cast iron pipeline R&D projects. URL `http://opsweb.phmsa.dot.gov/pipelineforum/reports-and-research/cast-iron-pipeline/`. Online; accessed 18-December-2012.

Raghavan, R., C. D. Thompson. 1987. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica* **7** 365–374.

Sherali, H. D., B. M. P. Fraticelli. 2002. A modification of Benders' decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse. *Journal of Global Optimization* **22** 319–342.

Srinivasan, A. 1996. An extension of the Lovász Local Lemma and its applications to integer programming. *Proc. of the seventh annual ACM-SIAM symposium on discrete algorithms*. 6–15.

Appendix

A. Deterministic equivalent of the joint job scheduling and crew assignment problem

PROPOSITION 3. *The deterministic equivalent of the optimization problem* (2) *is the following mixed integer program.*

$$\begin{aligned}
\underset{C,V,X,Y,Z}{\textit{minimize}} \quad & C \\
\textit{subject to} \quad & \sum_{\omega\in\Omega_t} P_t(\omega)V_t(\omega) \le C, \qquad t=1,\dots,T, \\
& d_L Z_{tk}(\omega) + \sum_{i=1}^{n} d_i Y_{itk} \le V_t(\omega), \qquad t=1,\dots,T,\ \omega\in\Omega_t,\ k\in K_t, \\
& \sum_{t=1}^{\tau_i} X_{it} = 1, \qquad i=1,\dots,n, \\
& \sum_{k\in K_t} Y_{itk} = X_{it}, \qquad i=1,\dots,n,\ t=1,\dots,T, \\
& \sum_{k\in K_t} Z_{tk}(\omega) = L(\omega), \qquad t=1,\dots,T,\ \omega\in\Omega_t, \\
& X_{it}\in\{0,1\}, \qquad i=1,\dots,n,\ t=1,\dots,T, \\
& Y_{itk}\in\{0,1\}, \qquad i=1,\dots,n,\ t=1,\dots,T,\ k\in K_t, \\
& Z_{tk}(\omega)\in\mathbb{Z}^+, \qquad t=1,\dots,T,\ \omega\in\Omega_t,\ k\in K_t.
\end{aligned} \tag{10}$$

Proof. Let us denote by $\mathcal{F}$ the feasible region of (2). We can write the first-stage problem as:

$$\begin{aligned}
\underset{W,X,Y}{\text{minimize}} \quad & W \\
\text{subject to} \quad & E_t\left[F_t\left(Y_t, L(\omega)\right)\right] \le W, \qquad t=1,\dots,T, \\
& (X,Y)\in\mathcal{F}.
\end{aligned} \tag{11}$$

Using the probability distribution P_t for the gas leak scenarios Ω_t, we can rewrite constraint set (11) for each t as $\sum_{\omega\in\Omega_t} P_t(\omega)\times F_t(Y_t, L(\omega)) \le W$.

Similarly, we can rewrite the second-stage recourse problem $F_t(Y_t, L(\omega))$ as an MIP:

$$\begin{aligned}
\underset{V,Z}{\text{minimize}} \quad & V_t(\omega) \\
\text{subject to} \quad & d_L Z_{tk}(\omega) + \sum_{i=1}^{n} d_i Y_{itk} \le V_t(\omega), \qquad k\in K_t, \\
& \sum_{k\in K_t} Z_{tk}(\omega) = L(\omega) \\
& Z_{tk}(\omega)\in\mathbb{Z}^+, \qquad k\in K_t.
\end{aligned}$$

Combining the above two reformulations results in the optimization problem (10) in the proposition. Since Ω_t is finite, there is a finite number of constraints, and problem (10) is a MIP. $\square$

B. Job scheduling LP-based heuristic

The following describes the job scheduling algorithm (Algorithm 1) in detail. Consider the following linear programming (LP) relaxation of the job scheduling MIP:

$$\begin{aligned}
\underset{C,X}{\text{minimize}} \quad & C \\
\text{subject to} \quad & d_L E_t[L(\omega)] + \sum_{i=1}^{n} d_i X_{it} \le |K_t| \cdot C, \qquad t = 1, \ldots, T, \\
& \sum_{t=1}^{\tau_i} X_{it} = 1, \qquad i = 1, \ldots, n, \\
& X_{it} \ge 0, \qquad i = 1, \ldots, n, \; t = 1, \ldots, T.
\end{aligned} \tag{12}$$

Denote the optimal solution as X^{LP} and the optimal cost as C^{LP}. To round X^{LP} into a feasible job schedule, define the following sets:

$$\begin{aligned}
I^s(t) &= \left\{ i : 0 < X_{it}^{LP} < 1 \right\}, \\
I^f(t) &= \left\{ i : X_{it}^{LP} = 1 \right\}, \\
T_i &= \left\{ t : 0 < X_{it}^{LP} < 1 \right\}.
\end{aligned}$$

The rounding step of the algorithm consists of solving the following MIP:

$$\begin{aligned}
\underset{W,X}{\text{minimize}} \quad & W \\
\text{subject to} \quad & d_L E_t[L(\omega)] + \sum_{i \in I^f(t)} d_i + \sum_{i \in I^s(t)} d_i X_{it} \le |K_t| \cdot W, \qquad t = 1, \ldots, T, \\
& \sum_{t \in T_i} X_{it} = 1, \qquad i \in I^s(1) \cup \cdots \cup I^s(T), \\
& X_{it} \in \{0, 1\}, \qquad t = 1, \ldots, T, \; i \in I^s(t).
\end{aligned} \tag{13}$$

Note that any set of variables $\{X_{it}\}$ satisfying the last two constraints in (13) is a rounding of the fractional variables of the LP solution $\{X_{it}^{LP}\}$. Let us denote by $\{X_{it}^{R}\}$ the solution to the MIP (13). For all $i = 1, \ldots, n$ and $t = 1, \ldots, T$, set the rounded solution X_{it}^H by the following equation:

$$X_{it}^H = \begin{cases} 0, & \text{if } X_{it}^{LP} = 0, \\ 1, & \text{if } X_{it}^{LP} = 1, \\ X_{it}^R, & \text{otherwise.} \end{cases}$$

Note that X^H is a feasible solution to the original job scheduling problem (4).

C. Proof of Theorem 1

Let X^{LP} and C^{LP} be the optimal solution for the LP relaxation (12) of the job scheduling problem. To prove Theorem 1, we first require proving the following proposition.

PROPOSITION 4. *Let X^{LP} be the optimal solution to the LP relaxation* (12)*. Define the randomized rounding $\tilde{X}$ such that for each $i = 1, \ldots, n$, randomly round exactly one of the indices* $\{1, 2, \ldots, T\}$ *to 1, with index t chosen with probability X_{it}^{LP}. Then with* positive probability*,*

$$\max_{t=1,\ldots,T} \frac{1}{|K_t|} \left(d_L E_t[L(\omega)] + \sum_{i=1}^{n} d_i \tilde{X}_{it} \right) \le C^{LP} \left(1 + H\left(C^{LP}, \frac{1}{e\delta} \right) \right),$$

Algorithm 1 LP-based job scheduling algorithm

Require: Planning horizon $\{1,\ldots,T\}$, and standard jobs indexed by $1,\ldots,n$, where job i has deadline $\tau_i \leq T$ and duration d_i

Ensure: Feasible schedule X^H with $\sum_{t=1}^{\tau_i} X_{it}^H = 1$, for all $i = 1,\ldots,n$, and $X_{it}^H \in \{0,1\}$

$X^{LP} \leftarrow$ solution to the linear relaxation (12)
Initialize $X_{it}^H \leftarrow 0$ for all i, t
$I^f(t), I^s(t) \leftarrow \varnothing$ for all t, and $T_i \leftarrow \varnothing$ for all i
for $i = 1$ **to** n **do**
  for $t = 1$ **to** T **do**
    if $X_{it}^{LP} = 1$ **then**
      $X_{it}^H \leftarrow 1$
      $I^f(t) \leftarrow I^f(t) \cup \{i\}$
    else if $X_{it}^{LP} \in (0,1)$ **then**
      $I^s(t) \leftarrow I^s(t) \cup \{i\}$
      $T_i \leftarrow T_i \cup \{t\}$
    end if
  end for
end for
$X^R \leftarrow$ solution deterministic rounding MIP (13)
for $t = 1$ **to** T **do**
  for $i \in I^s(t)$ **do**
    $X_{it}^H \leftarrow X_{it}^R$
  end for
end for

where

$$H(w,p) \triangleq \frac{1}{w}\left(\min_{s=1,\ldots,T} |K_s|\right)^{-1} \sqrt{\frac{1}{2}\left(\sum_{i=1}^n d_i^2\right)\ln\left(\frac{1}{p}\right)}. \tag{14}$$

Proof. For a given t, define $\tilde{X}_t = (\tilde{X}_{1t}, \tilde{X}_{2t}, \ldots, \tilde{X}_{nt})$. Moreover, define the function $f_t : [0,1]^n \mapsto \mathbb{R}$ as

$$f_t(x_1, x_2, \ldots, x_n) \triangleq \frac{1}{|K_t|}\left(d_L E_t[L(\omega)] + \sum_{i=1}^n d_i x_i\right).$$

That is $f_t(\tilde{X}_t)$ is a function of a random variable which represents the ratio of expected hours scheduled on day t to the number of crews on day t under the randomly rounded solution.

Define the "bad" event B_t as the event that the random schedule results in a ratio of scheduled hours to number of crews exceeding the bound in Proposition 4, i.e.,

$$B_t \triangleq \left[f_t(\tilde{X}_t) > C^{LP}\left(1 + H\left(C^{LP}, \frac{1}{e\delta}\right)\right)\right].$$

Therefore, proving Proposition 4 is equivalent to proving

$$0 < \Pr\left(\max_{t=1,\dots,T} f_t(\tilde{X}_t) \le C^{LP}\left(1 + H\left(C^{LP}, \frac{1}{e\delta}\right)\right)\right) = \Pr\left(\bigcap_{t=1}^{T} \overline{B_t}\right). \tag{15}$$

Since there is limited dependency among the "bad events" (i.e., each event B_t is mutually dependent on at most $\delta - 1$ other events), then we can use Lovász's Local Lemma to prove (15).

Lemma 1 **(Lovász's Local Lemma)**. *Let $B_1, \dots, B_m$ be a set of events with $\Pr(B_i) \le p < 1$ and each event B_i is mutually of all but at most s of the other B_j. If $e \cdot p(s+1) \le 1$, then $\Pr\left(\bigcap_{i=1}^{m} \overline{B_i}\right) > 0$.*

Thus, to use Lovász's Local Lemma, we need to find a bound p such that $\Pr(B_t) \le p$ and $e \cdot p(s+1) \le 1$. Since f_t a function of bounded differences, and $\tilde{X}_t$ are independent random variables, we use a large deviations bound (McDiarmid's inequality) to derive a bound on $\Pr(B_t)$.

Lemma 2 **(McDiarmid's inequality)**. *Let $X_1, X_2, \dots, X_m$ be independent random variables all taking values in the set $\mathcal{X}$. Further, let $f : \mathcal{X}^m \mapsto \mathbb{R}$ be a function of $X_1, \dots, X_m$ that satisfies $\forall i, \forall x_1, \dots, x_m, x_i' \in \mathcal{X}$,*

$$|f(x_1, \dots, x_i, \dots, x_m) - f(x_1, \dots, x_{i-1}, \hat{x}_i, x_{i+1}, \dots, x_m)| \le c_i. \tag{16}$$

Then for any $\epsilon > 0$, $\Pr\left(f - E[f] \ge \epsilon\right) \le \exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^m c_i^2}\right)$.

It is easy to verify that f_t satisfies condition (16) in McDiarmid's inequality, with $c_i = d_i/|K_t|$. Note that we can bound $E[f_t(\tilde{X}_t)]$, since

$$E[f_t(\tilde{X}_t)] = \frac{1}{|K_t|}\left(d_L E_t[L(\omega)] + \sum_{i=1}^{n} d_i E[\tilde{X}_{it}]\right) = \frac{1}{|K_t|}\left(d_L E_t[L(\omega)] + \sum_{i=1}^{n} d_i X_{it}^{LP}\right) \le C^{LP},$$

where the last inequality follows since C^{LP} and X^{LP} are feasible for the LP relaxation (12).

Therefore, we have that

$$\Pr\left\{f_t(\tilde{X}_t) \ge C^{LP}\left(1 + H\left(C^{LP}, \frac{1}{e\delta}\right)\right)\right\} \le \Pr\left\{f_t(\tilde{X}_t) - E[f_t(\tilde{X}_t)] \ge C^{LP} H\left(C^{LP}, \frac{1}{e\delta}\right)\right\} \tag{17}$$

$$\le \exp\left(\frac{-2\left(|K_t| C^{LP} H\left(C^{LP}, \frac{1}{e\delta}\right)\right)^2}{\sum_i d_i^2}\right), \tag{18}$$

$$= \exp\left(-\ln(e\delta)\left(\frac{|K_t|}{\min_s |K_s|}\right)^2\right), \tag{19}$$

$$\le \exp(-\ln(e\delta)) = \frac{1}{e\delta} \tag{20}$$

where inequality (17) follows from $E[f_t] \le C^{LP}$, and inequality (18) follows from McDiarmid's inequality with $\epsilon = C^{LP} H\left(C^{LP}, \frac{1}{e\delta}\right)$.

Therefore, since $\Pr(B_t) \le \frac{1}{e\delta}$, and each event B_t is mutually dependent on at most $\delta - 1$ other events, the conditions of Lovász's Local Lemma are met, proving (15) and Proposition 4. $\square$

Note that all realizations of $\tilde{X}$ are all the roundings of the LP solution X^{LP} into a feasible job schedule. Since out of all roundings, X^H produced by Algorithm 1 has the smallest value for the maximum ratio of scheduled hours to number of crews

$$\max_{t=1,\dots,T} \frac{1}{|K_t|}\left(d_L E_t[L(\omega)] + \sum_{i=1}^n d_i X_{it}^H\right),$$

then by Proposition 4, we have found a deterministic rounding X^H for which the maximum threshold for worst-case scheduled hours per crew is at most $C^{LP}\left(1+H\left(C^{LP},\frac{1}{e\cdot\delta}\right)\right) \le C^{OPT}\left(1+H\left(C^{LP},\frac{1}{e\cdot\delta}\right)\right)$.

D. Binary search initialization for the job scheduling LP-based heuristic

Here, we describe the details of the job scheduling algorithm (Algorithm 2) which is initialized by a binary search procedure, then solves an LP relaxation of a MIP, and rounds the LP solution to a feasible schedule. The binary search procedure is adapted from Lenstra et al. (1990). For a fixed parameter C, define the following set of job-date pairs:

$$\Gamma_C \triangleq \left\{(i,t) : t \le \tau_i \text{ and } \frac{d_i}{|K_t|} \le C\right\}.$$

Consider the solving the following linear optimization problem $LP(C)$:

$$\begin{aligned}
\underset{X}{\text{minimize}} \quad & C \\
\text{subject to} \quad & d_L E_t[L(\omega)] + \sum_{i:(i,t)\in\Gamma_C} d_i X_{it} \le |K_t|\cdot C, \qquad t = 1,\dots,T, \\
& \sum_{t:(i,t)\in\Gamma_C} X_{it} = 1, \qquad i = 1,\dots,n, \\
& X_{it} \ge 0, \qquad (i,t)\in\Gamma_C.
\end{aligned} \tag{21}$$

Using binary search, find the smallest value of C for which the LP- problem $LP(C)$ is feasible. Let C^B be this value, and X^B the corresponding optimal solution. We round X^B into a feasible job schedule in the same manner described in Appendix B for rounding X^{LP}.

E. Proof of Theorem 2

Using the binary search procedure in Appendix D, Lenstra et al. (1990) show that the LP solution has a rounding in which the makespan is at most 2 times the optimal makespan C^{OPT}. Since the rounding procedure of Algorithm 2 results in the rounding $X^{H'}$ with the smallest makespan, then the makespan of $X^{H'}$ is at most $2C^{OPT}$. Moroever, with a minor modification of the proof of Theorem 1 (see Appendix C), we can prove that $C^{H'} \le C^B\left(1+H\left(C^B,\frac{1}{e\delta}\right)\right)$. Thus, since $C^B \le C^{OPT}$, we have that

$$\frac{C^{H'}}{C^{OPT}} \le \min\left\{2,\ 1+\frac{1}{C^B}\left(\min_{t=1,\dots,T}|K_t|\right)^{-1}\sqrt{\frac{1}{2}\left(\sum_{i=1}^n d_i^2\right)(1+\ln\delta)}\right\}. \quad \square$$

Algorithm 2 LP-based job scheduling algorithm with binary search initialization procedure

Require: Planning horizon $\{1,\ldots,T\}$, and standard jobs indexed by $1,\ldots,n$, where job i has deadline $\tau_i \le T$ and duration d_i

Ensure: Feasible schedule $X^{H'}$ with $\sum_{t=1}^{\tau_i} X_{it}^{H'} = 1$, for all $i = 1,\ldots,n$, and $X_{it}^{H'} \in \{0,1\}$

Initialize $u \leftarrow$ makespan of arbitrary feasible job schedule
Initialize $l \leftarrow 0$
while $l < u$ **do**
  $C \leftarrow \left\lfloor \frac{1}{2}(l+u) \right\rfloor$
  Solve $LP(C)$ in (21)
  if $LP(C)$ is feasible **then**
    $u \leftarrow C$
  else
    $l \leftarrow C$
  end if
end while
$C^B \leftarrow C$
$X^B \leftarrow$ solution of $LP(C^B)$
Initialize $X_{it}^{H'} \leftarrow 0$ for all i, t
$I^f(t), I^s(t) \leftarrow \varnothing$ for all t, and $T_i \leftarrow \varnothing$ for all i
for $i = 1$ **to** n **do**
  for $t = 1$ **to** T **do**
    if $X_{it}^B = 1$ **then**
      $X_{it}^{H'} \leftarrow 1$
      $I^f(t) \leftarrow I^f(t) \cup \{i\}$
    else if $X_{it}^B \in (0,1)$ **then**
      $I^s(t) \leftarrow I^s(t) \cup \{i\}$
      $T_i \leftarrow T_i \cup \{t\}$
    end if
  end for
end for
$X^{R'} \leftarrow$ solution deterministic rounding MIP (13)
for $t = 1$ **to** T **do**
  for $i \in I^s(t)$ **do**
    $X_{it}^{H'} \leftarrow X_{it}^{R'}$
  end for
end for

F. Deterministic equivalent of the assignment phase problem

PROPOSITION 5. *The deterministic equivalent of the day t two-stage assignment phase problem* (7) *is the following mixed integer program.*

$$\begin{aligned} \underset{V,Y,Z}{\textit{minimize}} \quad & \sum_{\omega\in\Omega_t} P_t(\omega)V(\omega) \\ \textit{subject to} \quad & d_L Z_k(\omega) + \sum_{i\in I_t} d_i Y_{ik} \le V(\omega), \qquad \omega\in\Omega_t,\ k\in K_t, \\ & \sum_{k\in K_t} Y_{ik} = 1, \qquad i\in I_t, \\ & \sum_{k\in K_t} Z_k(\omega) = L(\omega), \qquad \omega\in\Omega_t, \\ & Y_{ik}\in\{0,1\}, \qquad i\in I_t,\ k\in K_t \\ & Z_k(\omega)\in\mathbb{Z}^+, \qquad \omega\in\Omega_t,\ k\in K_t. \end{aligned} \tag{22}$$

Proof. Let us denote by $\mathcal{F}$ the feasible region of (7). Using the probability distribution P_t for the gas leak scenarios Ω_t, we can rewrite the objective function of (7) as $\sum_{\omega\in\Omega_t} P_t(\omega)F_t(Y, L(\omega))$.

Similarly, we can rewrite the second-stage recourse problem $F_t(Y, L(\omega))$ as an MIP:

$$\begin{aligned} \underset{V,Z}{\text{minimize}} \quad & V(\omega) \\ \text{subject to} \quad & d_L Z_k(\omega) + \sum_{i=1}^{n} d_i Y_{ik} \le V(\omega), \qquad k\in K_t, \\ & \sum_{k\in K_t} Z_k(\omega) = L(\omega), \\ & Z_k(\omega)\in\mathbb{Z}^+, \qquad k\in K_t. \end{aligned}$$

Therefore, (7) is equivalent to:

$$\begin{aligned} \underset{V,Y,Z}{\text{minimize}} \quad & \sum_{\omega\in\Omega_t} P_t(\omega)V(\omega) \\ \text{subject to} \quad & d_L Z_k(\omega) + \sum_{i=1}^{n} d_i Y_{ik} \le V(\omega), \qquad \omega\in\Omega_t,\ k\in K_t, \\ & \sum_{k\in K_t} Z_k(\omega) = L(\omega), \qquad \omega\in\Omega_t, \\ & Z_k(\omega)\in\mathbb{Z}^+, \qquad \omega\in\Omega_t,\ k\in K_t, \\ & Y\in\mathcal{F}. \end{aligned}$$

Since Ω_t is finite, there is a finite number of constraints, and this problem is a MIP. □

G. Proof of Proposition 2

Suppose that $Z^*_{k_0}(\omega_1) > Z^*_{k_0}(\omega_2)$ for some $k_0\in K$. We will construct a gas leak assignment $\tilde{Z}(\omega_2)$ for scenario ω_2 which has maximum hours (makespan) no greater than that of $Z^*(\omega_2)$. First, note that since there are less gas leak jobs in scenario ω_1, we have that

$$\max_{k\in K}\left\{ d_L Z^*_k(\omega_1) + \sum_{i\in I} d_i Y^*_{ik} \right\} \le \max_{k\in K}\left\{ d_L Z^*_k(\omega_2) + \sum_{i\in I} d_i Y^*_{ik} \right\}. \tag{23}$$

Define $\tilde{Z}(\omega_2)$, a new gas leak assignment for ω_2, by letting $\tilde{Z}_{k_0}(\omega_2) = Z^*_{k_0}(\omega_2) + 1$, $\tilde{Z}_{k_1}(\omega_2) = Z^*_{k_1}(\omega_2) - 1$ (where k_1 is some crew in K with $Z^*_{k_1}(\omega_2) > 0$), and $\tilde{Z}_k(\omega_2) = Z^*_k(\omega_2)$ for all $k \in K \setminus \{k_0, k_1\}$. Note that the assigned hours (load) of crew k_1 is strictly smaller under this new assignment. Now all that is left to prove is that the load of crew k_0 is smaller than the maximum load in assignment $Z^*(\omega_2)$. Note that since $\tilde{Z}_{k_0}(\omega_2) \le Z^*_{k_0}(\omega_1)$, the load of crew k_0 in assignment $\tilde{Z}(\omega_2)$ under scenario ω_2 is no greater than its load in assignment $Z^*(\omega_1)$ under scenario ω_1. Relationship (23) implies that the load of k_0 under both scenarios is no greater than the maximum load of the assignment $Z^*(\omega_2)$ under scenario ω_2. Therefore, the load of crew k_0 does not increase the maximum load beyond the makespan of assignment $Z^*(\omega_2)$. $\square$

Algorithm 3 Crew assignment algorithm (stochastic variant of LPT)

Require: $\Omega = \{\omega_1, \omega_2, \ldots, \omega_m\}$, where $L(\omega_1) < L(\omega_2) < \cdots < L(\omega_m)$, and standard jobs sorted in decreasing job duration, i.e. $d_1 \ge d_2 \ge \cdots \ge d_n$

Ensure: Assignment of all standard jobs and gas leak jobs to crews under all leak scenarios

$B_k(\omega_{m+1}) \leftarrow 1$, for all $k \in K$
for $s = m$ **to** 1 **do**
  $B_k(\omega_s) \leftarrow 0$, for all $k \in K$
  for $l = 1$ **to** $L(\omega_2)$ **do**
    $\tilde{K} \leftarrow \arg\min_{k \in K} (B_k(\omega_s))$ {set of crews with smallest current load}
    $B_{k_0}(\omega_s) \leftarrow B_{k_0}(\omega_s) + d_L$, where $k_0 \in \tilde{K}$ such that $B_{k_0}(\omega_s) < B_{k_0}(\omega_{s+1})$
  end for
end for
for $i = 1$ **to** n **do**
  for $k \in K$ **do**
    $\tilde{B}_k(\omega) \leftarrow B_k(\omega) + d_i$, for all $\omega \in \Omega$ {Load in scenario ω if job i is assigned to crew k}
    $A_k(\omega) \leftarrow \max\Big(B_1(\omega), \ldots, B_{k-1}(\omega), \tilde{B}_k(\omega), B_{k+1}(\omega), \ldots, B_K(\omega)\Big)$, for all $\omega \in \Omega$ {Makespan in scenario ω if job i is assigned to crew k}
  end for
  $K_i \leftarrow \arg\min_{k \in K} \{\sum_s P(\omega_s) A_k(\omega_s)\}$
  $k_0 \in \arg\min_{k \in K_i} \{\sum_s P(\omega_s) B_k(\omega_s)\}$
  $B_{k_0}(\omega) \leftarrow \tilde{B}_{k_0}(\omega)$, for all $\omega \in \Omega$
end for

H. Optimal crew assignment for examples

Table 13 **Optimal crew assignment with leak distribution 1.**

		Crew 1	Crew 2	Crew 3	Crew 4	Crew 5	Crew 6	Crew 7
Standard jobs	Total # jobs	2	2	2	1	2	3	3
	Total hours	10.3	10.7	10.4	2.6	10.5	10.3	10.6
Gas leak jobs	Total hours *(1 leak, $p=1$)*	0	0	0	8	0	0	0

Table 14 **Optimal crew assignment with leak distribution 2.**

		Crew 1	Crew 2	Crew 3	Crew 4	Crew 5	Crew 6	Crew 7
Standard jobs	Total # jobs	3	2	2	1	3	2	2
	Total hours	10.4	10.4	10.2	2.6	10.6	10.4	10.6
Gas leak jobs	Total hours *(0 leak, $p=0.1$)*	0	0	0	0	0	0	0
	Total hours *(1 leak, $p=0.8$)*	0	0	0	8	0	0	0
	Total hours *(2 leaks, $p=0.1$)*	0	0	8	8	0	0	0

Table 15 **Optimal crew assignment with leak distribution 3.**

		Crew 1	Crew 2	Crew 3	Crew 4	Crew 5	Crew 6	Crew 7
Standard jobs	Total # jobs	2	3	2	1	2	2	3
	Total hours	3.8	11.6	11.8	3.8	11.5	11.1	11.8
Gas leak jobs	Total hours *(0 leak, $p=0.2$)*	0	0	0	0	0	0	0
	Total hours *(1 leak, $p=0.6$)*	8	0	0	0	0	0	0
	Total hours *(2 leaks, $p=0.2$)*	8	0	0	8	0	0	0

Table 16 **Optimal crew assignment with leak distribution 4.**

		Crew 1	Crew 2	Crew 3	Crew 4	Crew 5	Crew 6	Crew 7
Standard jobs	Total # jobs	2	1	2	2	3	2	3
	Total hours	3.8	3.8	11.5	11.1	11.6	11.8	11.7
Gas leak jobs	Total hours *(0 leak, $p=0.4$)*	0	0	0	0	0	0	0
	Total hours *(1 leak, $p=0.2$)*	8	0	0	0	0	0	0
	Total hours *(2 leaks, $p=0.4$)*	8	8	0	0	0	0	0

Table 17 Optimal crew assignment with leak distribution 5.

		Crew 1	Crew 2	Crew 3	Crew 4	Crew 5	Crew 6	Crew 7
Standard jobs	Total # jobs	3	1	2	2	2	3	2
	Total hours	10.6	2.7	10.2	10.4	10.5	10.3	10.7
Gas leak jobs	Total hours *(0 leak, $p = 0.3$)*	0	0	0	0	0	0	0
	Total hours *(1 leak, $p = 0.5$)*	0	8	0	0	0	0	0
	Total hours *(2 leaks, $p = 0.1$)*	0	8	8	0	0	0	0
	Total hours *(3 leaks, $p = 0.1$)*	0	8	8	0	0	8	0

Table 18 Optimal crew assignment with leak distribution 6.

		Crew 1	Crew 2	Crew 3	Crew 4	Crew 5	Crew 6	Crew 7
Standard jobs	Total # jobs	1	2	3	2	2	3	2
	Total hours	3.8	11.0	11.8	3.8	11.5	11.7	11.8
Gas leak jobs	Total hours *(0 leak, $p = 0.4$)*	0	0	0	0	0	0	0
	Total hours *(1 leak, $p = 0.3$)*	0	0	0	8	0	0	0
	Total hours *(2 leaks, $p = 0.2$)*	8	0	0	8	0	0	0
	Total hours *(3 leaks, $p = 0.1$)*	8	8	0	8	0	0	0

Table 19 Optimal crew assignment with leak distribution 7.

		Crew 1	Crew 2	Crew 3	Crew 4	Crew 5	Crew 6	Crew 7
Standard jobs	Total # jobs	3	2	1	2	2	3	2
	Total hours	11.4	8.1	3.5	11.4	8.1	11.4	11.4
Gas leak jobs	Total hours *(0 leak, $p = 0.5$)*	0	0	0	0	0	0	0
	Total hours *(1 leak, $p = 0.2$)*	0	0	8	0	0	0	0
	Total hours *(2 leaks, $p = 0.1$)*	0	0	8	0	8	0	0
	Total hours *(3 leaks, $p = 0.2$)*	0	8	8	0	8	0	0

I. Crew assignment from Algorithm 3 for examples

Table 20 Crew assignment using Algorithm 3 with leak distribution 1.

		Crew 1	Crew 2	Crew 3	Crew 4	Crew 5	Crew 6	Crew 7
Standard jobs	Total # jobs	1	3	3	2	2	2	2
	Total hours	2.6	11.5	11.4	9.5	9.7	10.3	10.3
Gas leak jobs	Total hours *(1 leak, $p = 1$)*	8	0	0	0	0	0	0

Table 21 Crew assignment using Algorithm 3 with leak distribution 2.

		Crew 1	Crew 2	Crew 3	Crew 4	Crew 5	Crew 6	Crew 7
Standard jobs	Total # jobs	1	3	2	2	3	2	2
	Total hours	2.7	10.2	10.1	10.2	11.4	10.3	10.4
Gas leak jobs	Total hours *(0 leak, $p=0.1$)*	0	0	0	0	0	0	0
	Total hours *(1 leak, $p=0.8$)*	8	0	0	0	0	0	0
	Total hours *(2 leaks, $p=0.1$)*	8	8	0	0	0	0	0

Table 22 Crew assignment using Algorithm 3 with leak distribution 3.

		Crew 1	Crew 2	Crew 3	Crew 4	Crew 5	Crew 6	Crew 7
Standard jobs	Total # jobs	1	3	2	2	3	2	2
	Total hours	2.7	10.2	10.1	10.2	11.4	10.3	10.4
Gas leak jobs	Total hours *(0 leak, $p=0.2$)*	0	0	0	0	0	0	0
	Total hours *(1 leak, $p=0.6$)*	8	0	0	0	0	0	0
	Total hours *(2 leaks, $p=0.2$)*	8	8	0	0	0	0	0

Table 23 Crew assignment using Algorithm 3 with leak distribution 4.

		Crew 1	Crew 2	Crew 3	Crew 4	Crew 5	Crew 6	Crew 7
Standard jobs	Total # jobs	1	1	3	2	3	3	2
	Total hours	5.0	5.4	11.9	9.9	11.7	11.2	10.3
Gas leak jobs	Total hours *(0 leak, $p=0.4$)*	0	0	0	0	0	0	0
	Total hours *(1 leak, $p=0.2$)*	8	0	0	0	0	0	0
	Total hours *(2 leaks, $p=0.4$)*	8	8	0	0	0	0	0

Table 24 Crew assignment using Algorithm 3 with leak distribution 5.

		Crew 1	Crew 2	Crew 3	Crew 4	Crew 5	Crew 6	Crew 7
Standard jobs	Total # jobs	1	3	3	2	2	2	2
	Total hours	2.7	10.2	10.4	10.4	10.7	10.5	10.5
Gas leak jobs	Total hours *(0 leak, $p=0.3$)*	0	0	0	0	0	0	0
	Total hours *(1 leak, $p=0.5$)*	8	0	0	0	0	0	0
	Total hours *(2 leaks, $p=0.1$)*	8	8	0	0	0	0	0
	Total hours *(2 leaks, $p=0.1$)*	8	8	8	0	0	0	0

Table 25 Crew assignment using Algorithm 3 with leak distribution 6.

		Crew 1	Crew 2	Crew 3	Crew 4	Crew 5	Crew 6	Crew 7
Standard jobs	Total # jobs	1	2	3	2	2	3	2
	Total hours	5.0	7.6	10.7	10.1	10.2	11.4	10.3
Gas leak jobs	Total hours *(0 leak, $p=0.4$)*	0	0	0	0	0	0	0
	Total hours *(1 leak, $p=0.3$)*	8	0	0	0	0	0	0
	Total hours *(2 leaks, $p=0.2$)*	8	8	0	0	0	0	0
	Total hours *(2 leaks, $p=0.1$)*	8	8	8	0	0	0	0

Table 26 Crew assignment using Algorithm 3 with leak distribution 7.

		Crew 1	Crew 2	Crew 3	Crew 4	Crew 5	Crew 6	Crew 7
Standard jobs	Total # jobs	2	2	3	2	2	2	2
	Total hours	7.2	8.0	9.6	10.1	10.2	9.9	10.3
Gas leak jobs	Total hours *(0 leak, $p = 0.5$)*	0	0	0	0	0	0	0
	Total hours *(1 leak, $p = 0.2$)*	8	0	0	0	0	0	0
	Total hours *(2 leaks, $p = 0.1$)*	8	8	0	0	0	0	0
	Total hours *(2 leaks, $p = 0.2$)*	8	8	8	0	0	0	0