Decision Support System for a Real-Time Field Service Engineer Scheduling Problem with Emergencies and Collaborations

Hanna Grzybowska, Charles Gretton, Philip Kilby, and S. Travis Waller

Repeated replanning with a heuristic for solving a type of vehicle routing problem was used in a dynamic routing and scheduling problem. This problem occurs when field service engineers are assigned a sequence of jobs to attend. The jobs are geographically distributed, and not all jobs to be undertaken are known in advance of planning. This dynamic occurrence of job requests is stochastic. Jobs are assigned an emergency level, which is highest for repair jobs involving a person in danger. In addition, some jobs require two engineers; such jobs are referred to as collaborative. The presented approach reschedules the pending jobs in an event-driven manner (i.e., every time a new repair job is required). The event-driven scheduling process ensures that jobs of high importance, with a high emergency level, are completed promptly. This approach to event-driven replanning will allow companies to plan for real-world scenarios with significantly fewer resources than are used in practice.

The assignment of field service engineers (FSEs) to jobs is a relevant and challenging planning problem for many companies. The complexity of such problems is exacerbated when jobs are spatially distributed, forcing FSEs to commute large distances during and between shifts, and when the factors conditioning the problem change dynamically (e.g., dynamic occurrence of high-priority requests, jobs that require multiple engineers to attend). This paper addresses a specific FSE management problem in which FSEs are designated to attend spatially distributed jobs characterized by emergency level. Some jobs are requested dynamically during scheduled operations. These jobs occur stochastically and must be attended, and therefore they pose a challenging replanning task. Also investigated are scenarios in which one job must be attended by two collaborating engineers. This study problem is called the real-time field service engineer scheduling problem with emergencies and collaborations (real-time FSEEC).

This research is a result of a commercial engagement with a company that manages a group of FSEs in a specific urban area. The FSEs attend electronic machinery requiring regular maintenance and occasional repairs in case of failure. The industrial partner wishes to remain anonymous. A set of synthetic benchmark scenarios was designed that exhibits the core challenges of the problem without revealing any details about that company's business or operations. The synthetic scenarios also represent a general class that occurs in field service engineering. The data sets are available on request.

In the synthetic scenarios, at the beginning of every month, the schedule planner creates an initial routing and scheduling plan that is based on known information, such as engineer availability and pending maintenance requests. The time at which a job is requested is called the call-in time. At the call-in time of a repair job, a new plan must be calculated to accommodate that request. The emergency level of the repair job imposes a particular assignment strategy. Also, the priorities and the time windows at which a job can be scheduled are defined in accordance with the emergency levels. There are three emergency levels: high and medium for repair jobs and low for maintenance jobs. All engineers have the same set of skills, which means that they can attend all machinery. However, in some cases, an FSE might need help. These cases are referred to as collaborations and are taken into consideration in scheduling.

These cases add complexity to the problem because one of the FSEs may be forced to wait for a collaborator to arrive. It can be the case that a collaborative effort is needed for only a part of the job, so that each of the collaborating FSEs might attend the same job for a different amount of time. In the scenarios, both FSEs must finish attending a job at the same time. Finally, the scenarios do not feature a depot, but rather the engineers start their routes from their residences.

In contrast to much of the literature treating similar problems, the objective here is not to provide a new algorithmic approach to solve the addressed problem. Instead the focus is on problem modeling with attention to special constraints defining collaborations and emergency levels. A simulation of realistic scenarios is used to examine whether continually replanning is an adequate strategy for managing operations. This study explores the potential of dynamic scheduling using real-time information. A commercially deployable approach was developed that takes advantage of available data and allocates dynamic requests online.

The remainder of this paper is organized as follows. The next section introduces the relevant literature and contrasts the real-time FSEEC with related problems. Then the real-time FSEEC is described. The proposed decision support system (DSS) is defined, along with the algorithms used. Then the testing scenarios and numerical results are presented, and finally conclusions are presented and potential future research is indicated.

H. Grzybowska and S. T. Waller, School of Civil and Environmental Engineering, University of New South Wales, Sydney, New South Wales 2052, Australia. C. Gretton and P. Kilby, National Information Communications Technology Australia, Canberra and Griffith University, Gold Coast, Canberra 2601, Australia. Corresponding author: H. Grzybowska, h.grzybowska@unsw.edu.au.

Transportation Research Record: Journal of the Transportation Research Board, No. 2497, Transportation Research Board, Washington, D.C., 2015, pp. 117–123.
DOI: 10.3141/2497-12

BACKGROUND

The real-time FSEEC is related to the vehicle routing problem (VRP). It also exhibits properties that are common in staff scheduling and rostering (SSR) problems. Both problems are NP-hard and have been intensely studied (*1–5*). The next section presents both related problems and discusses relevant literature.

Vehicle Routing Problems

The VRP is a challenging combinatorial optimization problem that underpins numerous models of transportation systems. It is the problem of designing optimal delivery routes from a depot to a set of geographically scattered points (e.g., cities, customers) respecting several constraints. Classically, the objective is to calculate the smallest fleet that satisfies all customer demands for traveling the minimal distance, so that (*a*) each route starts and ends at the depot, (*b*) each route is pursued by a single vehicle, and (*c*) each customer is visited exactly once by exactly one vehicle. Richer varieties of the VRP model have been developed to capture more operational concerns. The basic formulation can be supplemented with constraint defining or capacity, time windows, pairing, or priority. Real-time FSEEC considers hard time windows for both jobs and FSE availability. In the latter case, this means that overtime is not permitted.

Of all the variants of VRP, real-time FSEEC most resembles the multidepot VRP (MDVRP) and the open VRP (OVRP), both with time windows. In the MDVRP formulation (*6, 7*), several depots are assumed. A vehicle from any depot can be used to visit a particular customer. In real-time FSEEC, multiple depot locations correspond with locations of the residences of engineers. The difference between MDVRP and real-time FSEEC is that because of contractual arrangements that companies have with their engineers, the trip between the engineer's residence and the first job on the route is not paid. Thus, in real-time FSEEC, the cost of the first trip on a route is not included in the final cost of the solution. Another difference between the two problems is that in the latter, only one vehicle or FSE is present at the depot.

The OVRP is usually used to define operations of a company that does not own a vehicle fleet and subcontracts transportation services (*6, 8–10*). It assumes that each route starts at a depot but does not end there. Instead, the last visited customer on the route indicates the end of the route. Consequently, the cost of the last trip between the last customer and the depot is not included in the total route cost. The real-time FSEEC respects this rule.

Dynamic VRPs

The main feature distinguishing a dynamic VRP from its static variants is how requests are treated. In this work, the phrase "input data" refers to the model of the underlying routing and scheduling problem that is applied at some instance of time, that is, in the input to the planning exercise. Yang et al. distinguish the two cases for when input data become known in time (*11*). In the dynamic case, new requests occur during solution execution, whereas in the static case all requests are known a priori. In the dynamic case, input data become known after the start of plan execution. Such dynamics can be modeled deterministically, with qualitative, possibilistic, or Knightian uncertainty, or stochastically. In the first setting, the events that should occur during plan execution are known, and therefore they can be accommodated completely. In the second two cases, only when request events occur dynamically is it known exactly what the new input data correspond to and therefore what the new underlying transport problem corresponds to. In this context, real-time FSEEC can be defined as dynamic and stochastic.

In dynamic VRPs, the stochasticity in arrival of the requests may concern variables regarding customers (e.g., new customers or change in location, size of the demand, service time, or time window) (*11–13*), travel times (e.g., related to traffic incidents, pick hours) (*14, 15*), and vehicles (e.g., breakdowns) (*16*). Often, more than one of these issues is addressed in the same problem (*17–19*). Larsen provides an extensive literature review (*20*). The most recent literature review is by Pillac et al. (*21*).

In the literature, dynamically appearing customer requests are modeled with random events (*22*) or with events that appear in accordance with a known probability distribution (*23*). Following the latter, this study models the occurrence of dynamic requests for new jobs by using the exponential distribution. The approach is to replan, calculating modified routes and schedules to accommodate new requests as they occur. Therefore, the dispatcher is not provided with any deterministic or probabilistic data about future emergencies and makes decisions only in accordance with currently known information.

Staff Scheduling and Rostering

The main objective of the SSR process is to construct an optimized timetable for every employee in the crew. An optimized schedule addresses three key properties: (*a*) the demand for goods and services is satisfied, (*b*) regulations about shift durations and breaks are observed, and (*c*) the schedule accommodates staff preferences and fairness. The literature reviews indicate that SSR problems are combinatorial, highly constrained, and difficult to solve and have been intensively studied (*24*). In practice, the SSR process consists of three steps. First, the number of qualified employees needed to satisfy the requests is evaluated. Next, the employees are assigned to shifts so that the required staff levels are satisfied at all times, and the overall set of skills required during each shift is complete. Last, within each shift, an employee is assigned a list of tasks to perform. The FSEEC problem treats the last aspect of the typical SSR process, by which in each shift an employee is assigned to a set of requests.

Specific applications require special formulations and problem-solving approaches to create an SSR solution respecting the unique characteristics of the profession or industry. Transportation is an area for which there are many publications about SSR (*4, 25*). The common feature of the problem in this application area, and real-time FSEEC, is that both time and space are considered, that is, tasks have durations and time windows, and their locations are geographically distributed. The areas of emergency management (*26, 27*) and related call centers (*28*) make up the application areas in which the SSR problem definition resembles real-time FSEEC. In both areas, the demand for emergency service is not known in advance.

The most recent literature review on staff scheduling notes that incorporation of uncertainty in the definition of the SSR problem has not been examined in any of the treatments given in classical review papers (*25*). In real-time FSEEC, it is assumed that the repair jobs arrive at random, making the arrival pattern of the workload unpredictable.

FSE Scheduling

There is commercial software supporting the management of FSEs (*29, 30*). Only one publication is closely related to the management of FSEs (*31*). The authors of that work studied the field service scheduling problem with priorities (FSSP) in the mobile phone industry. The FSSP is similar to real-time FSEEC. However, unlike real-time FSEEC, FSSP considers a set of skills assigned to each FSE and does not consider collaborations between the FSEs. The consideration of collaborations is a key contribution of this paper.

Petrakis et al. first evaluated an offline planning system (*31*). They also developed an entirely online system and considered a hybrid of online and offline planning. The conclusions of the study showed the clear advantage in the online approach.

PROBLEM DEFINITION

Given a set of known requests, the underlying task in real-time FSEEC is to determine, for the smallest possible number of FSEs, a set of routes with a corresponding schedule, to attend a collection of spatially distributed jobs in such a way that the total cost is minimal. In other words, the task consists of determining a set of routes with assigned schedules so that

- Each nonattended job location is visited exactly once;
- Each route starts and ends at an engineer's residence location;
- An FSE attends one and only one job at a time;
- Each job is attended within a specified hard time window;
- Each job is attended within FSEs work hours;
- A collaboration requirement, which includes exactly two FSEs, defined for some jobs is respected (collaboration constraint);
- Each of the collaborating FSEs may spend different amounts of time attending the job, but they must always finish together;
- The cost of the first and the last trip on a route is not included in the entire routing cost; and
- The entire routing cost is minimized.

The problem definition includes a bounded scheduling horizon, $[E_H, L_H]$, consisting of T workdays. The availability of each FSE, $k \in K$, at a particular workday, $t \in T$, is defined as $[E_{kt}, L_{kt}]$. Each job, i, is associated with a hard time window, $[e_i, l_i]$, and a service time duration, s_i. The value a_i defines the moment of arrival of an FSE at the location of job i. When an FSE arrives before the job's time window is opened, he or she is allowed to wait. In this case the value of the waiting time is $w_i > 0$ and $w_i = \max\{e_i - a_i, 0\}$. An FSE starts executing job i at time z_i, where $z_i = \max\{a_i + w_i, 0\}$.

The problem calls for a definition of collaborations, when certain jobs must be attended by two FSEs. In those cases, there are two relative jobs, i and $p(i)$, to be attended in parallel by two FSEs. The service time durations of the jobs s_i and $s_{p(i)}$ do not have to be the same. However, both collaborative jobs must be finished at the same time. This assumption is in line with the situation observed in real life when an FSE who was assigned to a job first, after arriving at the location and evaluating the severity of the damage, can request the help of another FSE.

The problem is a mixed integer program. A complete graph $G = (N, A)$ is defined, where the set $N = \{0, 1, \ldots, n\}$ contains nodes representing jobs to attend, and the set $A = \{(i, j): i, j \in N, i \neq j\}$ comprises all feasible arcs between the nodes. The arcs are associated with a nonnegative cost, c_{ij}, assuming that $c_{ij} \neq c_{ji}\ \forall i, j \in N$. Thus, the problem formulation takes the following form:

$$\min \sum_{t\in T}\sum_{k\in K}\sum_{(i,j)\in A} c_{ij}x_{ijkt} + \sum_{t\in T}\sum_{k\in K}\sum_{i\in N} w_i \tag{1}$$

subject to

$$\sum_{t\in T}\sum_{k\in K}\sum_{j\in N} x_{ijkt} = 1 \qquad \forall i \in N \tag{2}$$

$$\sum_{i\in N} x_{ijkt} - \sum_{i\in N} x_{jikt} = 0 \qquad \forall j \in N;\ k \in K;\ t \in T \tag{3}$$

$$\sum_{i}\sum_{j\neq i} x_{jikt} \leq |S| - 1 \qquad \forall S \in N: |S| \geq 2;\ k \in K;\ t \in T \tag{4}$$

$$x_{ijkt}\left(z_i + s_i + c_{ij} - z_j\right) \leq 0 \qquad \forall i \in N;\ j \in N;\ k \in K;\ t \in T \tag{5}$$

$$E_H \leq E_{kt} \leq e_i \leq z_i \leq l_i \qquad \forall i \in N;\ k \in K;\ t \in T \tag{6}$$

$$z_i + s_i \leq l_i \leq L_{kt} \leq L_H \qquad \forall i \in N;\ k \in K;\ t \in T \tag{7}$$

$$w_i = \max\{0, e_i - a_i\} \qquad \forall i \in N \tag{8}$$

$$z_i + s_i = z_{p(i)} + s_{p(i)} \qquad \forall i \in N;\ p(i) \in N \tag{9}$$

$$x_{ijkt} \in \{0, 1\} \qquad \forall i \in N;\ j \in N;\ k \in K;\ t \in T \tag{10}$$

where S is a nonempty proper subset of N; $S \neq N$.

The nonlinear formulation of the objective function (Equation 1) minimizes the total travel time and waiting time of the solution that ensures its feasibility for the specified constraints. Equation 2 assigns each job to exactly one FSE. The degree constraint in Equation 3 specifies that an FSE may perform each job only once. Equation 4 eliminates the construction of potential subtours. The schedule consistency is maintained by Equations 5, 6, and 7, according to which an FSE can start attending job i within its hard time window $[e_i, l_i]$, within the hard time window of the FSE's availability $[E_{kt}, L_{kt}]$, and within the total scheduling horizon $[E_H, L_H]$. Equations 5, 6, and 7 are also true for collaboration jobs, $p(i)$. The value of the waiting time is defined by Equation 8. According to Equation 9, the collaborative jobs i and $p(i)$ must be finished at the same time. The last formulation, Equation 10, expresses the binary and nonnegative nature of the problem involved variable.

The optimization model plans and schedules the engineers for a known set of jobs. The dynamism of the real-time FSEEC problem relates to the repair jobs that are called in real time. Every time a new repair job is called in, the problem must be reformulated and resolved with updated input data. This need requires a policy defined outside the formulated model that reformulates the optimization problem at hand and specifies how and when to replan the jobs.

The policy defines additional constraints regarding a job's priority that reflect the emergency level. In real-time FSEEC, two emergency levels are defined for the repair jobs: (*a*) high, in which machinery must be fixed immediately, and (*b*) medium, in which machinery must be fixed the same day it failed. Maintenance jobs have a low emergency level because their time windows are much bigger than the considered time horizon. Consequently, repair jobs have priority over maintenance jobs. Thus an FSE who is attending a maintenance job can be asked to suspend it and go directly to attend a repair job. Later that day, the FSE must go back to complete the suspended job for the remaining amount of service time. If on that same day there is not enough time in the FSE's availability time window for completing the suspended job, the remaining part must be assigned as the first job to be attended to the next day. The suspended job always has to be finished by the FSE who started it.

PROBLEM-SOLVING APPROACH

This section presents the proposed DSS for real-time FSEEC. A depiction of the complete solution framework is presented in Figure 1.

The proposed DSS consists of two modules. The first is a solver and contains a collection of algorithms for solving the addressed routing and scheduling problem. The algorithms are presented in detail in a subsection. The second module is the simulation engine. Since the main objective of simulation engine is to prepare input data for the solver, it is explained first.

Simulation Engine

The simulation engine undertakes two tasks. First, it records all the events that occur over the simulated period, producing a report. Second, it simulates planned activities and synthesizes call-in events according to the synthetic scenario being simulated.

To define the activity performed by the engineer, a status variable is used. An engineer can be (*a*) in the process of attending a job (status = SERV), (*b*) on the move to the next job (status = MOVE), (*c*) waiting at a job's location (status = WAIT), or (*d*) not working because of unavailability (status = IDLE).

Because the simulation process runs in a loop, the simulator recuperates the output provided by the solver, runs supporting algorithms, and prepares the new input for the solver to be used for replanning. The inputs to the simulation engine constitute (*a*) a current routing and scheduling plan (i.e., sequences of jobs), (*b*) a description of the underlying road network in the form of a graph, and (*c*) an origin–destination travel time matrix in accordance with which the routing plan is being created and performed. The produced output includes (*a*) current FSEs' defining variables (i.e., locations and statuses), (*b*) a list of attended jobs, (*c*) a list of jobs that are currently being attended, (*d*) the amount of time remaining to finish the currently attended jobs, and (*e*) a list of jobs to attend.

The call-in for a new repair job j at time τ_j is an event generated by the simulator, which triggers replanning. At this time the simulation clock is stopped, and the simulator requests a new plan that accommodates the simulated call-in. First, the scheduling horizon $[E_H, L_H]$ is updated (i.e., $E_H = \tau_j$). The end of the scheduling horizon, L_H, remains the same. τ_j also defines the earliest possible moment when the repair job, j, can be started (i.e., $z_j \geq \tau_j$) since $e_j = \tau_j$. The upper bound of the time window of job j is equal to the highest value of the upper bound of the FSEs' availability for that day (i.e., $l_j = \max\{L_{kt}\}\ \forall k \in K$). Next, the simulation engine reviews the current routes and identifies the current locations and statuses of all FSEs. If the status of an FSE is WAIT or SERV, then the FSE's location corresponds with the location of the currently visited job. The remaining waiting time or service time of the job is calculated. If an FSE's status is MOVE, the current location is assumed to be the location of the last attended job. Next, for each route, the attended jobs are identified and kept in a set-apart record. The jobs that have not been attended and job j are put in another set-apart list. The next step depends on the emergency level of the newly called-in repair job, j.

If the emergency level of job j is medium, its time window $[e_j, l_j]$ is updated as follows: $e_j = \tau_j$ and $l_j = \max\{L_{k,t+1}\}\ \forall k \in K$, where $t + 1$ represents the next workday. Job j is kept in the list of the jobs to be attended. This set of jobs will be passed as input to the solver to be reassigned.

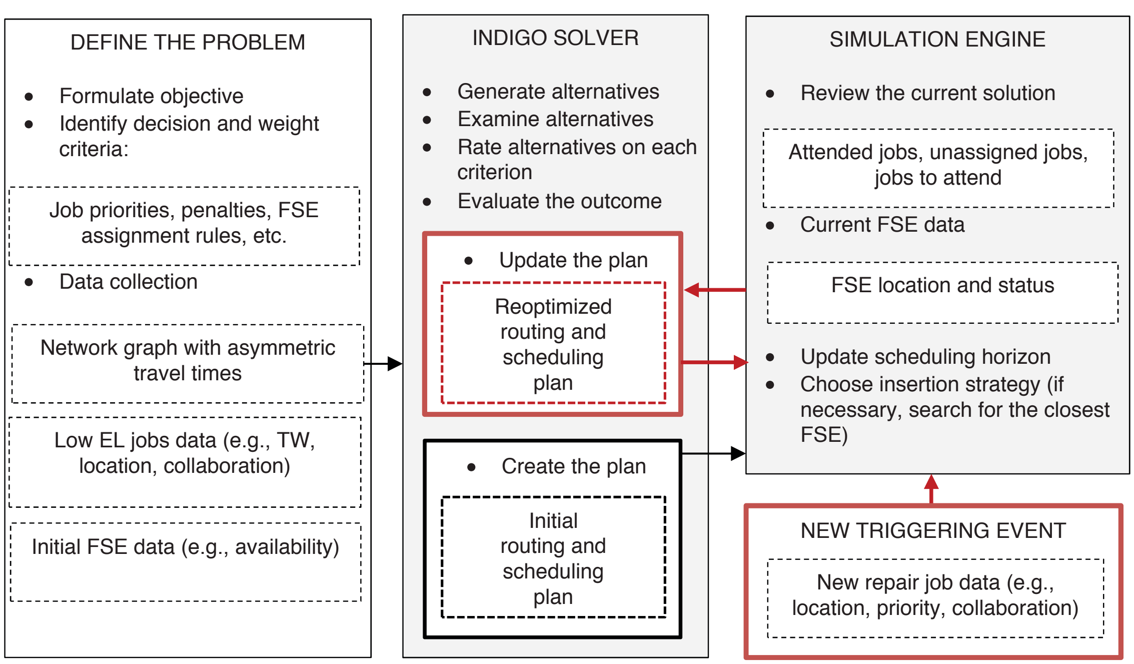

FIGURE 1 Framework of proposed DSS.

In the case in which the emergency level of job j is high, the simulation engine searches for an FSE who at time τ_j is closest to job j. The closest FSE k^* is always selected from the FSEs who are currently active (i.e., status ≠ IDLE) and not in the process of attending another repair job (i.e., with high or medium emergency level). At this stage, the simulation engine can follow one of the two procedures, depending on the current status of the closest FSE.

If the closest FSE k^* is on the move between jobs i and $i + 1$, he or she can be diverted and sent to job j. This means that job j will be the first one to be attended by the closest FSE once the simulation clock is running again. The value of the performed travel time (i.e., $\tau_j - t_i$, where t_i is the moment of leaving the location of job i) is referred to as the cost of redirection and is included in the final cost of the solution.

If the closest FSE k^* is in the process of executing job i with low emergency level, this job is suspended, and the FSE is sent to attend the new repair job, j, directly. For the nonperformed part of the suspended job, the simulation engine creates a dummy job i'. Its location corresponds with the location of the suspended job i, and its service time, $s_{i'}$, is equal to the remaining nonperformed service time of the suspended job, s_i (i.e., $s_{i'} = z_i + s_i - \tau_j$). The time window of the dummy job $[e_{i'}, l_{i'}]$ is set up as hard and tight. If the availability of the closest FSE k^* allows for it, the dummy job has to be attended the same day it was suspended, and its time window is set up as follows: $e_{i'} = \tau_j$ and $l_{i'} = L_{k^*,t}$. If the dummy job cannot be fitted within the availability time window of the closest FSE k^* that day, it must be attended the next workday. Then, the dummy job's time window is set up as follows: $e_{i'} = \tau_j$ and $l_{i'} = L_{k^*t+1}$. Such a tight time window definition is forcing the solver to allocate the dummy job early in the new sequence.

In the last step, the simulation engine sums up times of travel, waiting, and service, which were performed on each route up until the moment when the simulation clock was stopped.

Once all the input data are ready, it is sent to the solver, which reschedules the jobs. When a new repair job $j + 1$ is called in, the clock is moved forward to the time instant τ_{j+1}, and the replanning is repeated.

In real life, real-time information on repair jobs could be gathered by a call center. In this research, that process is emulated. The information about a repair job is revealed dynamically and in accordance with the synthetic models in this case. The details of the scenarios are provided in the testing scenarios section.

Indigo Vehicle Routing Heuristic

It is proposed that all static route planning and scheduling is done with a flexible heuristic based on an adaptive large neighborhood search (ALNS). The basic structure of ALNS was described by Ropke and Pisinger (*32*). Here, the vehicle routing heuristic is implemented in the Indigo system (*33*). The search proceeds iteratively. During each iteration, visits from an incumbent solution are removed (become unassigned) and are then reinserted. Several removal-selection heuristics and insertion heuristics were described by Ropke and Pisinger (*32*). The Indigo heuristic used in the present work applies those authors' delete and insert methods. Routes are constructed by inserting visits one at a time. Insertion is parallel in that all routes are considered when the search is deciding which visit to insert next and in what position to insert it. For example, the minimum insert cost feature, used to choose the next visit to insert, will select the visit that increases the current solution cost by the smallest amount. Similarly, the minimum insert cost feature, when used to select the insert position, will insert the target visit in the position that will result in the smallest increase in cost.

COMPUTATIONAL EXPERIMENTS

The simulation engine was developed with Python 2.7.3 [(default version Apr 10 2012, 23:24:47) MSC version 1500, 64 bit (AMD64), 32-bit Windows]. The Indigo closer is implemented in C++.

Eight new benchmark scenarios were designed for testing the approach. The intention was to provide a generic representation of a real-time FSEEC problem that would closely emulate the scenarios encountered in real life.

Testing Scenarios

Three main experimental design levels are considered:

- Number of repair jobs,
- Emergency levels of repair jobs, and
- Collaborations.

The combinations of factors resulting from eight computational experiments are presented in Table 1.

TABLE 1 Test Scenarios

	Maintenance Jobs Statistics[a]			Repair Jobs Statistics by EL: Medium			Repair Jobs Statistics by EL: High		
Scenario	Number	Collaboration (%)	Duration	Number	Collaboration (%)	Duration	Number	Collaboration (%)	Duration
1	420	0	exp(109.5)	0	0	na	0	0	na
2	620	0	exp(109.5) exp(38.1)	0	0	na	0	0	na
3	420	0	exp(109.5)	200	0	exp(38.1)	0	0	na
4	420	0	exp(109.5)	180	0	exp(38.5)	20	0	exp(41.9)
5	420	5	exp(109.5)	0	0	na	0	0	na
6	620	5	exp(109.5) exp(38.1)	0	0	na	0	0	na
7	420	5	exp(109.5)	200	5	exp(38.1)	0	0	na
8	420	5	exp(109.5)	180	5	exp(38.5)	20	5	exp(41.9)

NOTE: exp = exponential distribution; na = not applicable.
[a]Low emergency level (EL).

In every scenario, the number of engineers is constant and equal to 10. The scheduling horizon corresponds to 1 month of work consisting of 20 workdays. Each workday is 8 h long. Half the FSEs start work at 7 a.m., and the other half start at 9 a.m. Thus, the total availability of the engineers is 96,000 min.

Scenarios 1, 2, 5, and 6 do not consider repair jobs, unlike Scenarios 3, 4, 7, and 8, which include dynamically called-in repair jobs. The number of jobs with low emergency level is equal to 420, except in Scenarios 2 and 6, where the number of maintenance jobs is 620. Scenarios 3 and 7 consider 200 dynamic jobs with medium emergency level, and Scenarios 4 and 8 consider 180 jobs with medium emergency level and 20 jobs with high emergency level.

The duration of a job depends on its type. For modeling duration of jobs, exponential distribution with $\lambda = 0.00913$ was used for jobs with a low emergency level and with $\lambda = 0.026$ for jobs with medium and high emergency levels, except in Scenarios 2 and 6, where both distributions were used: $\lambda = 0.00913$ for 420 jobs and $\lambda = 0.026$ for 200 jobs.

In reality, time intervals between call-in instants of repair jobs are not uniform. They may vary according to, for example, time of day (i.e., peak hours). Since an exponential distribution provides a good approximate model for the time until a default takes place, it is used to model the length of time intervals between call-in times of repair jobs. In the scenarios containing only repair jobs with medium emergency level (i.e., Scenarios 3 and 7), the exponential distribution uses parameter $\lambda = 0.00547$. In scenarios containing both medium and high emergency level repair jobs (i.e., Scenarios 4 and 8), two exponential distributions are used, one with $\lambda = 5.4532\text{E-}4$ to model time intervals between call-in times of jobs with a high emergency level, and another with $\lambda = 0.00492$ for jobs with a medium emergency level.

In cases in which collaborations are considered (i.e., Scenarios 5, 6, 7, and 8), their number roughly corresponds to 5% of the total number of jobs. Hence, there are 20 jobs with a low emergency level and 10 jobs with a medium or high emergency level requiring collaboration. In Scenario 6 there are 30 collaborative jobs.

The graph representing the underlying road network and values of travel time and distance corresponds with real-life values, but to maintain anonymity of the original data, random noise was introduced.

Results

The following table tabulates key statistics gathered while running simulator with the Indigo routing heuristic (Table 2).

In most of the scenarios, the average number of the attended jobs is equal to the total considered number of the jobs. In the scenarios including repair jobs with high emergency level (i.e., Scenarios 4 and 8), the number of unattended jobs is highest but is still very small in comparison with the total. It is equal to or smaller than 0.2%. In all the scenarios, all the high- and medium-emergency-level jobs were attended.

The values of travel time are the highest in the scenarios considering dynamic requests (Scenarios 3, 4, 7, and 8). A comparison of the scenarios with the same total number of jobs, 2 and 3, 2 and 4, 6 and 7, and 6 and 8, shows differences of around 93%, 97%, 90%, and 93%, accordingly, which shows how big an impact the dynamic requests have on the travel time. The difference in travel time between Scenarios 3 and 4 and between Scenarios 7 and 8 is equal to about 51% and 25%, respectively. The increment is a result of Scenarios 4 and 8 considering jobs with a high emergency level. These jobs have the highest priority of all and might result in suspension of maintenance jobs, causing additional travel time.

The waiting time appears only in the scenarios considering dynamic requests (Scenarios 3, 4, 7, and 8) and constitutes about 20% of the total solution cost. This result reflects that the jobs with medium or high emergency levels are associated with tight time windows, and engineers must wait for the time window to be opened.

That there are defined collaborative jobs (Scenarios 7 and 8) means that the increment of waiting time is no bigger than 6%, in comparison with the similar scenarios without collaborative jobs (Scenarios 3 and 4). Similar comparisons regarding the total solution cost show differences no bigger than 3%. Notwithstanding, the consideration of collaborative jobs has a big impact on the travel time. The difference in the total solution cost reaches 65% and 46% between Scenarios 3 and 7 and between Scenarios 4 and 8, respectively.

The total solution cost represents about 56%, 61%, 56%, and 61% of the total available work time in Scenarios 1, 2, 5, and 6, accordingly. These scenarios do not include dynamic requests, which explains why the level of occupancy of the engineers is quite low. In the scenarios that include dynamic requests (Scenarios 3, 4, 7, and 8), the average occupancy of the engineers is about 30% higher and corresponds to about 81%, 83%, 83%, and 85% of the total available work time, accordingly. The impact of the dynamic requests is significant and highlights the challenge resulting from planning and scheduling of the dynamic jobs.

Naturally the total value of CPU grows when dynamic requests are considered. However, the average CPU is lower in the scenarios considering dynamic requests in comparison with similar scenarios without dynamic jobs, because in the simulation process, calls are made

TABLE 2 Results

Scenario	Number of Routes	Number of Attended Jobs by Type		Total Solution Evaluation by Variable					
		Maintenance[a]	Repair[b]	Travel Time (min)	Waiting Time (min)	Service Time (min)	Solution Cost (min)	Total CPU Time (s)	Average CPU Time (s)
1	200	420.00	0	17.30	0.00	53,383.00	53,400.30	32.90	32.90
2	200	620.00	0	51.50	0.00	58,802.00	58,853.50	47.46	47.46
3	200	420.00	200	785.10	16,158.70	60,919.00	77,862.80	1,052.74	5.24
4	200	419.59	200	1,608.00	17,295.90	60,762.20	79,666.10	954.41	4.32
5	200	420.00	0	107.10	20.40	53,383.00	53,510.50	168.46	168.46
6	200	620.00	0	216.80	8.90	58,618.80	58,844.50	249.04	249.04
7	200	418.90	200	2,223.50	16,891.20	60,734.30	79,849.00	3,555.83	17.69
8	200	417.24	200	2,959.10	18,362.00	60,659.10	81,980.20	3,767.87	17.05

[a]Low EL.
[b]High–medium EL.

to the Indigo solver to determine the closest engineer. These calls are very short and affect the calculation of the average, lowering its value.

CONCLUSIONS AND FUTURE RESEARCH

This paper defined and computationally explored, using simulation, a methodological proposal for a DSS to assist in decision making in the management of FSEs when real-time information is available. The work was inspired by a commercial project. A model for the real-time FSEEC problem, used to carefully emulate the real-life case, was presented. Collaborative jobs were considered, and emergency levels were defined. The approach used an Indigo solver to rebuild the current routing plan, and the policy treating the repair jobs was defined in accordance with their emergency levels.

The big impact that dynamic requests have on the solution was shown, along with the benefits that might be provided by the event-driven replanning and rescheduling. In the synthetic scenarios, as in real-world examples, the reactive replanning approach can schedule engineers for all emergencies and satisfies the vast majority of periodic maintenance tasks. In the tested synthetic scenarios, all the dynamic requests were attended over all runs. Moreover, the approach can plan for real-world scenarios by using significantly fewer resources than are used in practice.

Future research will focus on the aspects of robustness in the real-time FSEEC problem, with a goal of treating robustness more preemptively by anticipating expected call-in jobs and their locality when planning and scheduling at the outset.

ACKNOWLEDGMENTS

National Information Communications Technology Australia is funded by the Australian Department of Communications and the Australian Research Council through the Information Communications Technology Centre of Excellence program.

REFERENCES

1. Golden, B.L., S. Raghavan, and E.A. Wasil (eds.). *The Vehicle Routing Problem: Latest Advances and New Challenges.* Springer, New York, 2010.
2. Goel, A., and V. Gruhn. A General Vehicle Routing Problem. *European Journal of Operational Research,* Vol. 191, No. 3, 2008, pp. 650–660.
3. Ernst, A.T., H. Jiang, M. Krishnamoorthy, and D. Sier. Staff Scheduling and Rostering: A Review of Applications, Methods and Models. *European Journal of Operational Research,* Vol. 153, 2004, pp. 3–27.
4. Ernst, A.T., H. Jiang, M. Krishnamoorthy, B. Owens, and D. Sier. An Annotated Bibliography of Personnel Scheduling and Rostering. *Annals of Operations Research,* Vol. 127, No. 1–4, 2004, pp. 21–144.
5. Bodin, L.D. Towards a General Model for Manpower Scheduling: Parts 1 and 2. *Journal of Urban Analysis,* Vol. 1, 1973, pp. 191–245.
6. Liu, T., Z. Jiang, R. Liu, and S. Liu. A Review of the Multi-Depot Vehicle Routing Problem. *Energy Procedia,* Vol. 13, 2011, pp. 3381–3389.
7. Geetha, S., P.T. Vanathi, and G. Poonthalir. Metaheuristic Approach for the Multi-Depot Vehicle Routing Problem. *Applied Artificial Intelligence,* Vol. 26, 2012, pp. 878–901.
8. Fleszar, K., I.H. Osman, and K.S. Hindi. A Variable Neighbourhood Search Algorithm for the Open Vehicle Routing Problem. *European Journal of Operational Research,* Vol. 195, 2009, pp. 803–809.
9. Salari, M., P. Toth, and A. Tramontani. An ILP Improvement Procedure for the Open Vehicle Routing Problem. *Computers and Operations Research,* Vol. 37, No. 12, 2010, pp. 2106–2120.
10. MirHassani, S.A., and N. Abolghasemi. A Particle Swarm Optimisation Algorithm for Open Vehicle Routing Problem. *Expert Systems with Applications,* Vol. 38, 2011, pp. 11547–11551.
11. Yang, J., P. Jaillet, and H. Mahmassani. Real-Time Multivehicle Truckload Pickup and Delivery Problems. *Transportation Science,* Vol. 38, No. 2, 2004, pp. 135–148.
12. Mitrovic-Minic, S., R. Krishnamurti, and G. Laporte. Double-Horizon Based Heuristics for the Dynamic Pickup and Delivery Problem with Time Windows. *Transportation Research Part B,* Vol. 38, No. 8, 2004, pp. 669–685.
13. Ferrucci, F., S. Bock, and M. Gendreau. A Pro-Active Real-Time Control Approach for Dynamic Vehicle Routing Problems Dealing with the Delivery of Urgent Goods. *European Journal of Operational Research,* Vol. 225, 2013, pp. 130–141.
14. Taniguchi, E., and H. Shimamoto. Intelligent Transportation System Based Dynamic Vehicle Routing and Scheduling with Variable Travel Times. *Transportation Research Part C,* Vol. 12, 2004, pp. 235–250.
15. Potvin, J.-Y., Y. Xu, and I. Benyahia. Vehicle Routing and Scheduling with Dynamic Travel Times. *Computers and Operations Research,* Vol. 33, 2006, pp. 1129–1137.
16. Li, J.-Q., P.B. Mirchandani, and D. Borenstein. Real-Time Vehicle Rerouting Problems with Time Windows. *European Journal of Operational Research,* Vol. 194, 2009, pp. 711–727.
17. Fleischmann, B., S. Gnutzmann, and E. Sandvoss. Dynamic Vehicle Routing Based on Online Traffic Information. *Transportation Science,* Vol. 38, No. 4, 2004, pp. 420–433.
18. Gendreau, M., F. Guertin, J.-Y. Potvin, and R. Séguin. Neighborhood Search Heuristics for a Dynamic Vehicle Dispatching Problem with Pickups and Deliveries. *Transportation Research Part C,* Vol. 14, 2006, pp. 157–174.
19. Grzybowska, H. *Combination of Vehicle Routing Models and Dynamic Traffic Simulation for City Logistics Applications.* PhD dissertation. Universitat Politècnica de Catalunya, Spain, 2012.
20. Larsen, A. *The Dynamic Vehicle Routing Problem.* PhD dissertation. Technical University of Denmark, Lyngby, 2000.
21. Pillac, V., M. Gendeau, C. Gueret, and A.L. Medaglia. A Review of Dynamic Vehicle Routing Problems. *European Journal of Operations Research,* Vol. 225, 2013, pp. 1–11.
22. Chen, Z., and H. Xu. Dynamic Column Generation for Dynamic Vehicle Routing with Time Windows. *Transportation Science,* Vol. 40, No. 1, 2006, pp. 74–88.
23. Ichoua, S., M. Gendreau, and J.-Y. Potvin. Exploiting Knowledge About Future Demands for Real-Time Vehicle Dispatching. *Transportation Science,* Vol. 40, 2006, pp. 211–215.
24. Brucker, P., R. Qu, and E. Burke. Personnel Scheduling: Models and Complexity. *European Journal of Operational Research,* Vol. 210, 2011, pp. 467–473.
25. Van den Bergh, J., J. Belien, P. De Bruecker, E. Demeulemeester, and L. De Boeck. Personnel Scheduling: A Literature Review. *European Journal of Operational Research,* Vol. 226, 2012, pp. 367–385.
26. Erdogan, G., E. Erkut, A. Ingolfsson, and G. Laporte. Scheduling Ambulance Crews for Maximum Coverage. *Journal of the Operational Research Society,* Vol. 61, 2010, pp. 543–550.
27. Sadjadi, S.J., R. Soltani, M. Izdakah, F. Saberian, and M. Darayi. A New Nonlinear Stochastic Staff Scheduling Model. *Scientia Iranica* E, Vol. 18, No. 3, 2011, pp. 699–710.
28. Saccani, N. Forecasting for Capacity Management in Call Centres: Combining Methods, Organization, People and Technology. *IMA Journal of Management Mathematics,* Vol. 24, 2013, pp. 189–207.
29. ClickSoftware. *ClickSoftware Positioned as a Leader for Third Consecutive Year in the Gartner Magic Quadrant for Field Service Management.* Oct. 16, 2013. http://ir.clicksoftware.com.
30. Maoz, M. *Magic Quadrant for Field Service Management.* Gartner Research, Stamford, Conn., 2008.
31. Petrakis, I., C. Hass, and M. Bichler. On the Impact of Real-Time Information on Field Service Scheduling. *Decision Support Systems,* Vol. 53, 2012, pp. 282–293.
32. Ropke, S., and D. Pisinger. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science,* Vol. 40, No. 4, 2006, pp. 455–472.
33. Kilby, P., and A. Verden. Flexible Routing Combing Constraint Programming, Large Neighbourhood Search, and Feature-Based Insertion. Presented at Artificial Intelligence and Logistics Workshop, IJCAI'11, Barcelona, Spain, 2011, pp. 43–48.

The Standing Committee on Transportation Network Modeling peer-reviewed this paper.