

Multicommodity network flow approach to the railroad crew-scheduling problem

B. Vaidyanathan
K. C. Jha
R. K. Ahuja

We present our solution to the crew-scheduling problem for North American railroads. (Crew scheduling in North America is very different from scheduling in Europe, where it has been well studied.) The crew-scheduling problem is to assign operators to scheduled trains over a time horizon at minimal cost while honoring operational and contractual requirements. Currently, decisions related to crew are made manually. We present our work developing a network-flow-based crew-optimization model that can be applied at the tactical, planning, and strategic levels of crew scheduling. Our network flow model maps the assignment of crews to trains as the flow of crews on an underlying network, where different crew types are modeled as different commodities in this network. We formulate the problem as an integer programming problem on this network, which allows it to be solved to optimality. We also develop several highly efficient algorithms using problem decomposition and relaxation techniques, in which we use the special structure of the underlying network model to obtain significant increases in speed. We present very promising computational results of our algorithms on the data provided by a major North American railroad. Our network flow model is likely to form a backbone for a decision-support system for crew scheduling.

Introduction

This paper concerns the development of new algorithms for railroad crew scheduling, which is one of the most important decision problems faced by railroad management. The crew-scheduling problem (CSP) consists of assigning crews to trains and creating rosters for each crew while satisfying a variety of Federal Railway Administration (FRA) regulations and trade-union work rules. The objectives are to minimize the cost of operating trains and to improve the quality of life for crew. An improved quality of crew life can lead to more productive employees, less employee turnover, and safer operations. North American railroads desire a software product that can help them make dramatic strides in crew management, but there is no methodology or software product that meets their specific needs. Although airline CSPs have been well studied and well solved, and railroad CSPs for European

and Asian railroads have also been addressed to some extent, CSPs for North American railroads, because of various union and regulatory complexities, are unique and remain unsolved, as we describe. This paper focuses on developing efficient network flow-optimization models that can form a backbone for all important aspects of crew scheduling for North American railroads: tactical, planning, and strategic. Henceforth in this paper, unless otherwise specified, we are referring to the context of North American railroads when we refer to the CSP.

U.S. freight tonnage is expected to double in volume over the next 20 years [1]. Railroad executives are extremely concerned about their ability to attract, train, and retain sufficient semiskilled labor to staff the increased number of train starts that will be needed to support this growth. Railroad companies pay train crew employees very high salaries (around \$70,000 per year

©Copyright 2007 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

0018-8646/07/\$5.00 © 2007 IBM

plus benefits¹) and still have difficulty attracting a high-quality work force. Operating a train as an engineer or managing a train as a conductor is not an easy job. It is further complicated by the fact that crews are seldom assigned to trains on the basis of a fixed schedule.

Generally the company calls the next available crew on the telephone and gives them their assignment two hours before a train departs. The crew takes the train to an away location, where they rest at a hotel waiting for an assignment, and then, when their turn comes up, they return on the assigned train. Consequently, train crews do not know from one day to the next, let alone a week or a month ahead, when they will be working. Train crews spend inordinate amounts of time on call, waiting for assignment, and then spend a great deal of time away from their homes and families. The irregular work style of railroad crews makes attracting potential employees to this career more difficult.

Also, railroads are not very profitable, typically earning less than ten percent return on capital, and thus are constrained from raising already high wages to attract more employees. To close the supply-and-demand gap for train crews, railroads must raise the productivity of their existing crews and change the historical pattern of operations to improve employees' quality of life. Success on both fronts will be required to ensure that railroads can continue to grow profitably. Labor cost, the largest component of a railroad operating expense, accounts for a large percentage of total revenue. Depending on the size of its network, each Class I railroad (a Class I railroad, as defined by the Association of American Railroads, has an operating revenue exceeding \$319.3 million) employs in the neighborhood of 15,000 to 25,000 locomotive engineers, conductors, and brakemen [3]. Consequently, improving the efficiency and effectiveness of train crews has the potential to dramatically reduce the cost of transportation. In this paper, we propose a network flow model and algorithms for assigning crews to trains that will make a significant impact on on-time performance and crew utilization and productivity, while also improving both the quality of life for crew and railroad safety.

In a large Class I railroad, various divisions have the task of analyzing train crews. Each group is interested in a different aspect of crew planning and scheduling. These perspectives can generally be characterized on the basis of the planning horizon of the issue at hand. Crew issues faced by railroads can be broadly classified into three categories. The models and algorithms proposed in this paper have applications in all of these areas of decision making:

- *Tactical*—Decisions that must be made immediately to support real-time train operations. Tactical problems have a planning horizon of 24 to 48 hours.
- *Planning*—Decisions that must be made as a part of the crew-schedule design process. Typically, railroads make adjustments to their network operating plan every month, with significant changes two or three times a year to account for both long-run and seasonal changes in traffic patterns.
- *Strategic*—Decisions that must be made considerably (i.e., more than a year) in advance of implementation to ensure that sufficient lead time is available to properly prepare and implement a new business practice.

Crew scheduling is one of the important mathematical problems in the rich set of planning and scheduling problems that can be modeled and solved using mathematical optimization techniques [4–6]. Crew scheduling is a well-known problem in operations research and has been historically associated with airlines and mass transit companies. Several papers on crew-scheduling management have appeared in the past literature; most notable among these are Wren [7], Bodin et al. [8], Carraresi and Gallo [9], Wise [10], and Desrosiers et al. [11]. These papers explore a set-covering-based approach to solve the CSP. Crew scheduling is conventionally divided into two stages: crew pairing and crew rostering. A *crew pairing* is a sequence of connected segments that start and end at the same crew base and satisfy all legal constraints. The objective is to find the minimum cost set of crew pairings so that each flight or train segment is covered. The objective of *crew rostering* is to assign individual crew members to trips or sequences of crew pairings. This pairing-and-rostering approach uses a set-covering formulation and is solved using a branch-and-bound framework in which the linear programming relaxations at each node in the branch-and-bound tree are solved by using column generation. This type of algorithm is also popularly called *branch-and-price*.

The pairing-and-rostering approach has gained wide acceptance and application in the airline industry. In a recent survey paper, Gopalakrishnan and Johnson [12] discuss the state of the art in solution methodologies for the airline crew pairing-and-rostering problem. There have also been some applications of this approach in the railroad industry. Caprara et al. [13], Ernst et al. [14], and Freling et al. [15] describe the application of this approach to railroad crew management. Caprara et al. describe the solution techniques adopted at an Italian railroad company. They consider several business rules that are specific to European railroads and develop a heuristic algorithm to generate rosters. Ernst et al.

¹The crew salary information is based on data obtained from [2]. In 2004, the average annual salary was \$63,000, and we extrapolated our figure based on the assumption of five percent annual raises.

consider a set of constraints called *workload constraints*, which imply that, on average, at each depot, each crew should work within acceptable time limits. While solving the problem, they relax the lower bound constraints. These constraints do not apply to the North American crew-scheduling problem. Freling et al. develop a decision-support system for airline and railroad crew planning using a branch-and-price-solution approach to solve the integrated problem of pairing and rostering. They show that the integrated approach provides significant benefits over the sequential approach of solving the pairing problem and then the rostering problem. Barnhart et al. [16, 17] also describe applications of the pairing-and-rostering approach.

Other research in the area of railroad crew scheduling that uses a different approach is that of Chu and Chan [18] and Walker et al. [19]. Chu and Chan consider the problem of crew scheduling for Light Rail Transit in Hong Kong. They decompose the problem into two stages: The first one partitions driving blocks into pieces, and the second one combines the pieces into runs. With their added localized optimization heuristics, they were able to solve the problem in less than half an hour of computation time. However, their approach could not model the problem completely, and the solution could only be used as a guideline to generate crew schedules. Walker et al. develop an integer-programming-based method for simultaneous disruption recovery of train timetable and crew roster in real time in the context of New Zealand railroads. The crew rules that they consider are relatively simplistic and can be expressed in the form of integer programming constraints, and they solve the problem using a column-and-constraint-generation algorithm.

While there have been several papers devoted to the study of railroad CSPs in Europe, Asia, and Australia, North American railroad problems are yet to be addressed satisfactorily. The only application of optimization methods to North American railroad crew scheduling is that of Gorman and Sarrafzadeh [20]. They studied crew balancing in the context of a major North American railroad, the Burlington Northern Santa Fe Railway, and developed a dynamic programming approach to solve the problem. The major shortcoming of their research is that they did not consider the possibility of different crew types, each governed by a different set of rules. Another drawback is that their approach could handle only a particular crew district configuration (*single-ended* crew district). While most crew districts in North America are *single-ended*, there are several that are *double-ended* or even more complex. The multicommodity network flow approach described in this paper models all of the rules considered by Gorman and Sarrafzadeh and also handles the case in which different crew pools have

different sets of rules. It is also applicable to all of the crew district configurations encountered in North America. These configurations are described in the next section.

From our review of the literature, we see that crew pairing-and-rostering approaches that use column generation have been the predominantly successful method for solving CSPs. However, this approach cannot be used for North American railroads for the following two reasons:

1. The rail networks of all Class I North American railroads are each divided into several crew districts. As a train follows its route, it goes from one crew district to another, picking up and dropping off crew at crew change terminals. Almost all crew districts consist of two or three terminals. Hence, a pairing-and-rostering approach is needlessly complex and is not required, as most pairings would consist of two trains, an outbound train from home to away and an inbound train from away to home. Also, rail networks typically consist of 200 to 300 crew districts, and the emphasis is on an approach that is simple and fast; column-generation techniques, which are computationally very intensive, are not appropriate.
2. The FRA regulations governing North American railroads are extremely complex. The most complicated of these rules is the first-in-first-out (FIFO) requirement. The FIFO constraint requires that crews should be called on duty in the order in which they become qualified for assignment at a location. None of the previously published solutions can handle a constraint of this kind. While this constraint is easy to state, explicitly modeling it makes the problem computationally intractable. The success of all approaches using column-generation or branch-and-price algorithms depends on the ease of solving the subproblem. The addition of the FIFO-side constraint to the problem would spoil the special structure of the subproblem and greatly increase the computation time. Because our model must be fast enough to be used in a real-time environment, this approach is once again not suitable.

To summarize, while there has been significant work in the area of crew scheduling for European, Asian, and Australian railroads as well as in the area of airline crew scheduling, there is no modeling approach that is flexible enough to tackle crew-scheduling problems faced by North American railroads. Our approach is a novel contribution to the application of innovative

optimization techniques to solve real-world business problems.

Network flow models have found successful application in a large number of diverse fields, including applied mathematics, computer science, engineering, management, and operations research [21]. In this paper, we model the CSP as a multicommodity network flow problem on an underlying space-time network. In this model, *crew pools* (sets of crews governed by the same business rules in a crew district) represent commodities, and the flow of individual crew members represents their assignments. The space-time network is constructed in such a way that the flow of crew members automatically satisfies all FRA regulations and trade-union rules other than the FIFO requirement. We formulate the CSP as an integer programming formulation (IPF) on a space-time network in which FIFO constraints are modeled as side constraints to the multicommodity flow problem. We show that solving the IPF using the standard branch-and-bound methodology is computationally intractable. On the other hand, the same problem with relaxed FIFO constraints can be solved very efficiently. We call the CSP with relaxed FIFO constraints the *relaxed problem*, and a solution to this problem provides a lower bound to the optimal solution of the CSP. We develop the *successive constraint generation* (SCG) algorithm, which starts with the solution of the relaxed problem and then iteratively adds constraints to remove FIFO violations. We also develop the *quadratic cost perturbation* (QCP) algorithm, which perturbs arc costs in the space-time network to penalize FIFO violations, and we prove that this approach guarantees FIFO compliance. We also show that the QCP approach produces optimal solutions in most cases and a gap of less than 0.2 percent for a few cases, with running times of the order of minutes.

Our major research contributions in this paper are the following:

- We develop a space-time network construction algorithm so that the flow of crews on this network automatically satisfies all FRA regulations and trade-union rules other than the FIFO requirement. The network-construction procedure is flexible enough to handle several combinations of rules and regulations and also various different configurations for different crew districts. It is also flexible enough to handle costs that are nonlinear functions of arc durations.
- We formulate the CSP as an integer programming problem on the space-time network, enforcing the FIFO requirements by adding side constraints. We prove the one-to-one correspondence between solutions to this integer program and solutions to the CSP.

- We show that the FIFO requirement, if handled by the integer programming approach, complicates the structure of the problem and renders it computationally intractable.
- We develop SCG, an exact algorithm that first solves the relaxed version of the integer program (without FIFO constraints) and then iteratively adds constraints in order to eliminate FIFO infeasibilities.
- We develop an approach based on a QCP that perturbs the cost of arcs in the space-time network in such a way as to penalize violations of the FIFO constraints.
- We prove that this method guarantees FIFO compliance for the problem that we study, and we also show that it produces the optimal solution in most cases.
- We present extensive computational results and case studies of our algorithms on real-world data.

The outline for the rest of the paper is as follows. In the next section, we give a complete description of the problem, focusing on the terminology used, governing rules and regulations, inputs, and the nature of constraints and the objective function. We then describe the mathematical modeling approach, which includes construction of the space-time network and the IPF, followed by a description of the solution approaches we have developed to efficiently solve the problem. We then present some of the practical applications of the model as well as computational results comparing the performance of all of our algorithms. We also present the results of a case study done on a representative scenario. The final section presents our concluding remarks.

Problem description

In this section, we provide an overview of CSPs faced by North American railroads. We first describe some of the essential terminology needed to understand the problem. We then review some of the typical regulations that govern crew management. Next, we list the set of inputs required to properly define and formulate the CSP, and finally we briefly describe the nature of constraints and the objective function.

Terminology

Crew district—The rail network is divided into crew districts that constitute a subset of terminals (nodes). Each crew district is typically a geographic corridor over which trains can travel with one crew. A typical railroad network for a major railroad in the U.S. may be divided into as many as 200 to 300 crew districts. As a train follows its route, it goes from one crew district to another, picking up and dropping off crew at crew change

terminals. In contrast to the airline industry, in which certain crews have the flexibility to operate over a large territorial domain, crews in the North American railroad industry are qualified to operate only in certain specific geographic territories. The physics of operating a train depends on the track geometry, which is defined by the hills and curves in the route and by signaling and interlocking systems that control the movement of trains. A crew must be intimately familiar with all aspects of a route in order to operate a train safely on that route. Consequently, most crews are qualified to operate on only a limited number of routes.

Crew pools—Within a crew district, there are several types of crews, called crew pools, which may be governed by different trade-union rules and regulations. For example, a crew pool may have preference over the trains operated in a prespecified time window. Similarly, a crew pool consisting of senior crew personnel is assigned only to predesignated trains so that crews in that pool know their working hours ahead of time. The multiple crew pools within each district, with different constraints, make CSPs complex and difficult to model mathematically.

Home and away terminals—The terminals where crews from a crew pool change trains are designated either as home terminals or away terminals. The railroad incurs no lodging cost when a crew is at its home terminal. However, the railroad has to make arrangements for crew accommodation at their away terminals. Different crew districts have different combinations of home and away terminals. A crew district with one home terminal and one away terminal is called a *single-ended crew district*. In such crew districts, a crew typically operates a train from its home location to an away location, rests in a hotel for at least eight hours, operates another train back to its home terminal, rests for ten to twelve hours, and repeats this cycle. The other type of crew district is a *double-ended crew district*, in which more than one terminal is a home terminal for different crew pools. Some of the other crew-district configurations are crew districts with one home terminal and several away terminals and crew districts with several home terminals and corresponding sets of away terminals.

Crew detention—Once a crew reaches its away terminal and rests for the prescribed number of hours, the crew is ready to head back to its home terminal. However, if there is no train, the crew may have to wait longer. According to the trade-union rules, once a crew is at the away terminal for more than a prespecified number of hours (generally 16 hours), the crew earns wages (called *detention costs*) without being on duty. For example, if a crew is waiting for assignment at the away terminal for 18 hours, it is paid detention charges for two hours.

Crew deadheading—This term refers to the repositioning of crew between terminals. At the away terminal, there is sometimes no return train projected for some time, or there is a shortage of crews at another terminal. Thus, instead of waiting for train assignment at its current terminal, the crew can take a taxicab or a train (as passengers) and deadhead to the home terminal. Similarly, the crew may also deadhead from a home terminal to an away terminal in order to rebalance and better match the train demand patterns and avoid train delays. Crew deadheading is expensive; the crew is considered to be on duty while deadheading and thus earns wages, and the railroad may also incur taxi expenses. Each year, a major freight railroad may spend tens of millions of dollars on crew deadheading.

On-duty and tie-up time—Whenever a crew is assigned to a train, it performs some tasks to prepare the train for departure; hence, crews are called on duty before train departure time. The time at which the crew must report for duty is called the *on-duty time*. Similarly, a crew performs some tasks after the arrival of the train at its destination, and thus crews are released from duty after the train arrival. The time at which the crew is released from duty is called *tie-up time*. We refer to the duty duration before train departure as *duty-before-departure* and the duty duration after train arrival as *duty-after-arrival*. Hence, the total duty period of a crew assigned to a train is the sum of the duty-before-departure, the duty-after-arrival, and the travel time of the train.

Duty period—In most cases, the duty period of a crew assigned to a train is the total duration between the *on-duty time* and the *tie-up time*. In some cases, when a crew rests for a very short time at an away location before being assigned to a train, the rest time and the duration of the second train may also be included in the duty period of the crew. (The calculation of the duty period is described in more detail in the next section.)

Dead crews—By federal law, a train crew can be on duty for a maximum of 12 consecutive hours, at which time the crew must cease all work, a state we refer to as *dead*, or *dog-lawed*. Dead crews are a frequent consequence of such events as delayed trains, congestion, and mechanical breakdowns. In such cases, crew dispatchers must send a relief crew by taxi or another train so that the dead crew can be relieved. The dead crew must then get sufficient rest before becoming available to operate another train.

Train delays—When a train reaches a crew change location and there is no available crew qualified to operate this train, the train must be delayed. Each train delay disrupts the operating plan and causes further delays due to the propagating network effect. Train delays due to crew unavailability are quite common among railroads. These delays are very expensive (some

estimate \$1,000 per hour) and can be reduced significantly through better crew scheduling and train scheduling.

Regulatory and contractual requirements

Assignment of crews to trains is governed by a variety of FRA regulations and trade-union rules. These regulations range from the simple to the complex. The regulations also vary from district to district and from crew pool to crew pool. We list below some examples of these kinds of constraints and their typical parameter values:

- The duty period of a crew cannot exceed 12 hours. The duty period of a crew on a train is usually calculated as the time interval between the on-duty time and tie-up time of the train.
- Whenever a crew is released from duty at the home terminal or has been deadheaded to the home terminal, they can resume duty only after 12 hours (ten hours' rest followed by a two-hour call period) if the duty period is greater than ten hours, and after ten hours (eight hours' rest followed by a two-hour call period) if the duty period is less than or equal to ten hours.
- Whenever a crew is released from duty at the away terminal, they must go for a minimum eight hours' rest, except under the following circumstances:
 - If the total time period corresponding to the last travel time from the home terminal followed by a rest time of less than four hours plus travel time of the next assignment back home is shorter than 12 hours (in this case, $duty\ period = travel\ time\ on\ inbound\ train + rest\ time\ at\ away\ location + travel\ time\ on\ outbound\ train$).
 - If the total time corresponding to the last travel time from the home terminal plus travel time of the next assignment back home is less than 12 hours when the rest time between the assignments is more than four hours (in this case, $duty\ period = travel\ time\ on\ inbound\ train + travel\ time\ on\ outbound\ train$).
- Crews belonging to certain pools must be assigned to trains in a FIFO order.
- A train can be operated only by crews belonging to prespecified pools.
- Every train must be operated by a single crew.
- Crews are guaranteed a certain minimum pay per month regardless of whether or not they work.

Figure 1 gives an example of the kind of decision process that must be followed by crew planners. Since the regulations for crew assignment can vary from district to district and crew pool to crew pool, it is a mathematical

challenge to build a unified model to formulate and solve this problem. This partly explains why these problems remain unsolved and no commercial optimization product has yet been deployed at railroads. Another reason why there has been limited operations research analysis of complex rail problems could be that the rail industry in the U.S. has been consolidated into only four major players, which means that there are not many customers for such a solution. Also, because of low margins in the railroad industry, investment in research funding is viewed as a luxury, despite a potentially high return on investment in automated decision-support systems.

Problem inputs

Here we describe the inputs to the mathematical formulation of the CSP.

- *Train schedule*—This schedule provides information about the departure time, arrival time, on-duty time, tie-up time, departure location, and arrival location for every train in each crew district through which it passes. We do not consider stochasticity in the train schedule, and we assume that train delays are due only to the unavailability of crew and not to train cancellations or other disruptions.
- *Crew pool attributes*—This includes attributes of various crew types, including their home locations, away locations, minimum rest time, and train preferences.
- *Crew initial position*—This provides the position of crew at the beginning of the planning horizon. It includes information about the terminal at which a crew is released from duty, the time of release, the number of hours of duty done in the previous assignment, and the crew pool to which the crew belongs.
- *Train-pool preferences*—These preferences, if any, are information about the set of trains that can be operated by a crew pool.
- *Away-terminal attributes*—This information includes the rest rules and detention rules for each crew pool at each away terminal.
- *Deadhead attributes*—This is the time taken to travel by taxi between two terminals in a crew district.
- *Cost parameters*—These parameters are used to set up the objective function for the CSP. They consist of crew wage per hour, deadhead cost per hour, detention cost per hour, and train delay cost per hour.

Constraints and objective function

The CSP involves making decisions regarding the assignment of crews to trains, deadheading of crews by

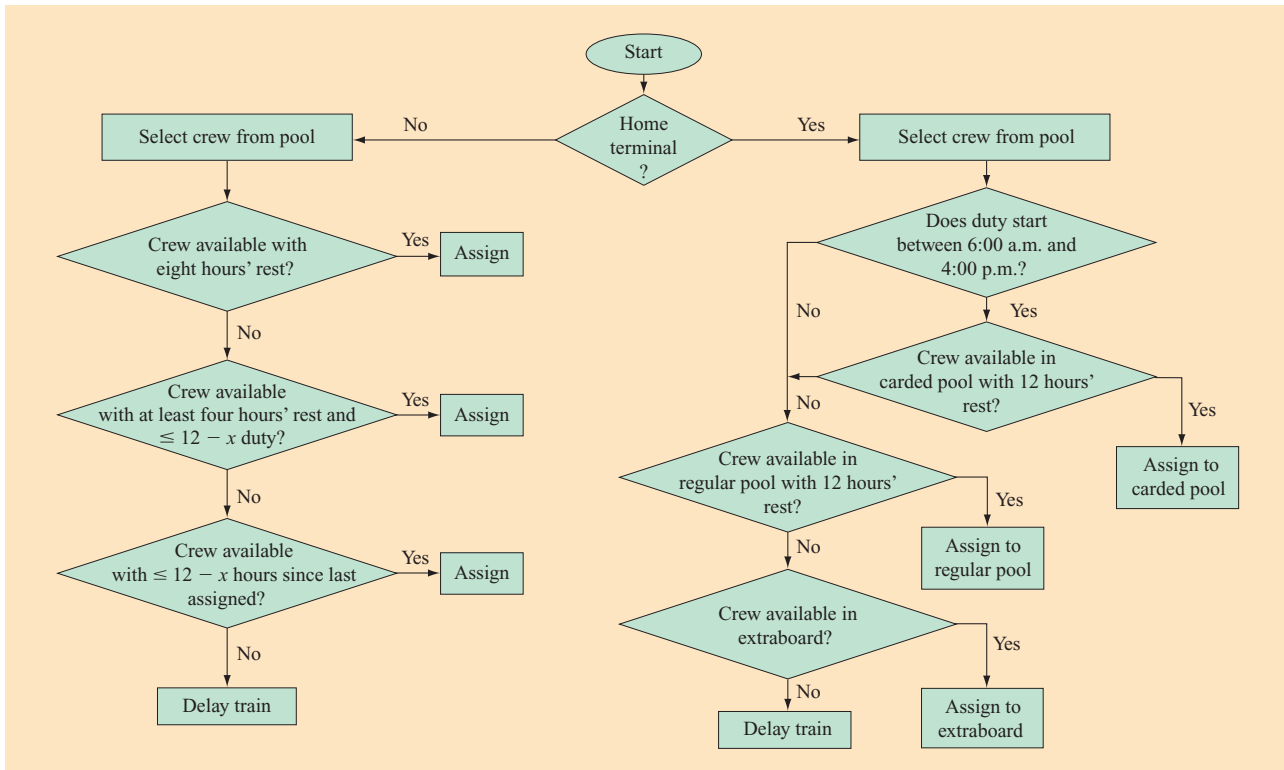


Figure 1

Example of a crew-assignment decision tree. (Extraboard: an emergency supply of crew that can be assigned if required.)

taxi, and train delays. The constraints can be categorized into two groups: operational constraints and contractual requirements. The operational constraints ensure that every train gets a qualified crew to operate it and prevents a crew from being assigned to more than one train at the same time. These constraints also include the assignment of certain crew pools to prespecified trains. The assignment of crews to trains must, in addition, satisfy the contractual requirements described above in the section on regulatory and contractual requirements. In our mathematical model, the operational constraints of the model are handled by the integer multicommodity flow formulation, and the contractual restrictions are honored in the network construction phase, as described in the next section. The objective function of the CSP is to minimize the total cost of crew wages, deadheading, crew detentions, and train delays.

Mathematical modeling

In this section, we present our mathematical modeling approach to solve the CSP. We first describe the construction of the space-time network, which is central to all of our solution methodologies. In the second part of this section, we formulate the CSP as an integer

multicommodity flow problem on this network, establish correspondence between the mathematical formulation and the CSP, and discuss the size of the problem and inherent computational complexities.

Space-time network

The CSP is formulated as an integer multicommodity flow problem with side constraints on a space-time network. We decompose the CSP for each crew district and construct the space-time network for a crew district. In the network, each node corresponds to a crew event and has two defining attributes: location and time. The events that we model while we construct the space-time network for the CSP are departure of trains, arrival of trains, departure of deadheads, arrival of deadheads, supply of crew, and termination of crew duty to mark the end of the planning horizon. All of the arcs in the network facilitate the flow of crews over time and space. **Figure 2** presents an example of the space-time network in a crew district. For the sake of clarity, this network represents only a subset of all of the arcs.

For each crew, we create a supply node whose time corresponds to the time at which this crew is available for assignment and whose location corresponds to

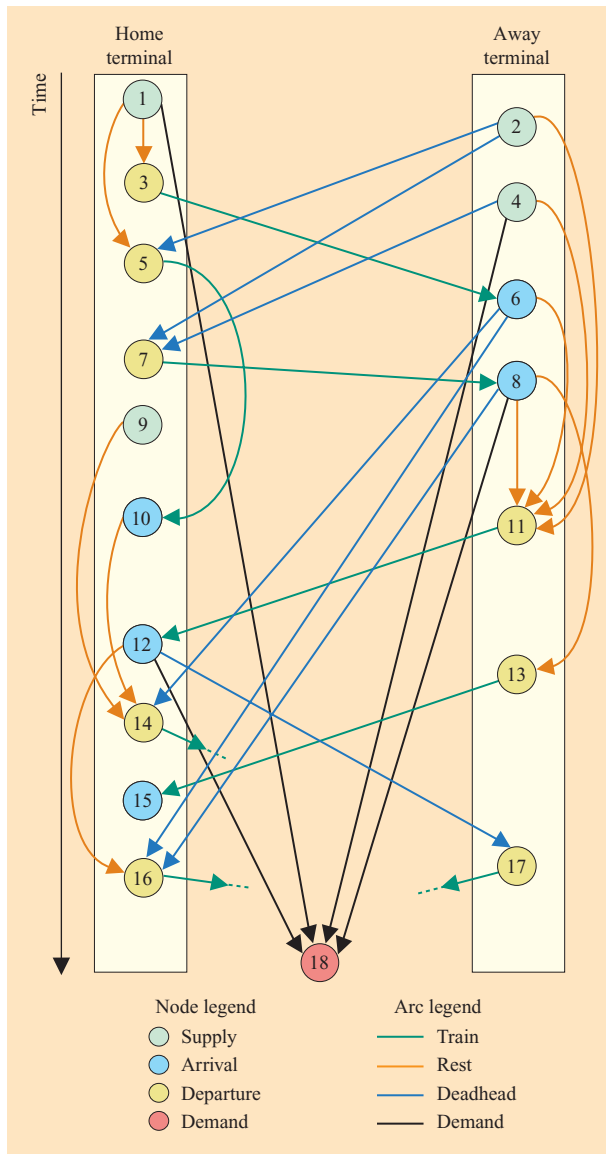


Figure 2

Space-time network for a single-ended district with a single crew type.

the terminal from which the crew is released for duty. Each supply node is assigned a supply of one unit and corresponds to a crew member. We also create a common sink node for all crews at the end of the planning horizon. This sink has no location attribute and has the time attribute equal to the end of the planning horizon. The sink node has a demand equal to the total number of crew supplied. The supply and sink nodes ensure that all of the crews that flow into the system at the beginning of the planning horizon are accounted for and flow out of the system at the end of the planning horizon.

For each train l passing through a crew district, we create a departure node l' at the first departing station of the train in the crew district and an arrival node l'' at the last arriving station of the train in the crew district. Each arrival or departure node has two attributes: place and time—for example, $place(l') = departure\ station(l)$ and $time(l') = on-duty\ time(l)$; and similarly, $place(l'') = arrival\ station(l)$ and $time(l'') = tie-up\ time(l)$.

In the network, for each train l we create a *train arc* (l', l'') connecting the departure node and arrival node. We create *deadhead arcs* to model the travel of crew by taxi. A deadhead arc is constructed between a train-arrival or crew-supply node at a location and a train-departure node at another location. All of the deadhead arcs that satisfy the contractual rules and regulations are created. We construct *rest arcs* to model the resting of a crew at a location. A rest arc is constructed between a train-arrival node or a crew-supply node at a location and a train-departure node at the same location. Rest arcs are created in conformance with the contractual rules and regulations. All rest arcs that satisfy the contractual rules and regulations are constructed. Since the contractual regulations are often crew-pool-specific, deadhead arcs and rest arcs are created specific to a crew pool. This implies that only crew belonging to a particular crew pool can flow on a particular rest arc or a deadhead arc. For example, suppose that a supply node corresponds to a crew belonging to crew pool A; then all of the arcs that emanate from this node can carry only crew belonging to crew pool A.

Finally, we create *demand arcs* from all train-arrival nodes and crew-supply nodes to the sink node. Each arc has an associated cost equivalent to the crew wages, deadhead costs, or detention costs, as the case might be. Also in the network, the time at the tail of an arc is always less than the time at the head of an arc, which ensures the forward flow of commodities on the time scale. It can be noted that all contractual requirements other than the FIFO constraint are easily handled in the network construction.

The space-time network models the flow of crews while honoring all of the contractual constraints except for the FIFO rule. However, it does not model the case in which qualified crews are not available for assignment to a train, thus causing a train delay. Therefore, we next construct additional arcs incorporating train delays. At a location, we create rest arcs and deadhead arcs that do not honor the rest regulations, and we penalize them to ensure that flows on these arcs occur only when qualified crews are not available for assignment. The flows on these arcs denote that trains will be delayed until crews become qualified for train operation. However, because the delay of a train may have a propagating effect on the availability of crews in subsequent assignments,

we assume here that the crew assigned to a delayed train has sufficient slack in the rest time at the train-arrival node to qualify it for subsequent assignments. Thus, the additional rest arcs and deadhead arcs model the train delays, with the assumption that the effect of train delays is only local.

To summarize, this section describes the construction of the space-time network for the CSP. Honoring contractual regulations while constructing the network significantly reduces the number of constraints in the integer program. We next present the multicommodity IPF of the CSP.

Integer programming formulation (IPF)

We formulate the CSP as an integer multicommodity flow problem on the space-time network described in the previous section. In our formulation, each crew pool represents a commodity. Crews enter the system at crew-supply nodes, and every supply node corresponds to a supply of one crew. The crew takes a sequence of connected train, rest, and deadhead arcs before finally reaching the sink. While flow of more than one crew type can take place on a train arc, rest and deadhead arcs can have flow of only one type, because the business rules for rest and deadhead are crew-pool-specific. We next present the IPF of the problem; the notation is shown in **Table 1**.

The decision variables are as follows:

x_l^c : flow of crew pool $c \in C$ on each train arc $l \in L$,

x_d : flow on deadhead arc $d \in D$,

and

x_r : flow on rest arc $r \in R$.

The objective function is

$$\min \sum_{l \in L} \sum_{c \in C} c_l^c x_l^c + \sum_{d \in D} c_d x_d + \sum_{r \in R} c_r x_r.$$

The constraints are the following:

$$\sum_{c \in C} x_l^c = 1, \quad \text{for all } l \in L, \quad (1)$$

$$\sum_{a \in i^+} x_a = 1, \quad \text{for all } i \in N_s, \quad (2)$$

$$\sum_{a \in N_d^-} x_a = f, \quad (3)$$

$$x_l^c = \sum_{a \in \text{tail}(l)_c^-} x_a, \quad \text{for all } l \in L, c \in C, \quad (4)$$

$$x_l^c = \sum_{a \in \text{head}(l)_c^+} x_a, \quad \text{for all } l \in L, c \in C, \quad (5)$$

Table 1 Notation for integer programming formulation.

N	Set of nodes in the space-time network
L	Set of train arcs in the network, indexed by l
D	Set of deadhead arcs in the network, indexed by d
R	Set of rest arcs in the network, indexed by r
A	Set of arcs in the space-time network, indexed by a
$G(N, A)$	Space-time network
N_s	Set of crew-supply nodes
N_d	Sink node
C	Set of crew pools in the system, indexed by c
i^+	Set of outgoing arcs at node i
i^-	Set of incoming arcs at node i
i_c^+	Set of outgoing arcs specific to crew pool c at node i
i_c^-	Set of incoming arcs specific to crew pool c at node i
A_r	Set of arcs on which flow will violate FIFO constraint if there is flow on rest arc r
f	Total number of available crew
M	A very large number
c_l^c	Cost of crew wages for crew pool $c \in C$ on train arc $l \in L$
c_d	Cost of deadhead arc $d \in D$
c_r	Cost of rest arc $r \in R$
$\text{tail}(l)$	Node from which arc l originates
$\text{head}(l)$	Node at which arc l terminates

$$\sum_{r \in A_r} x_r - M(1 - x_r) \leq 0, \quad \text{for all } r \in R, \quad (6)$$

$$x_l^c \in \{0, 1\} \text{ and integer, for all } l \in L, c \in C, \quad (7)$$

$$x_d \in \{0, 1\} \text{ and integer, for all } d \in D, \quad (8)$$

$$x_r \in \{0, 1\} \text{ and integer, for all } r \in R. \quad (9)$$

Constraint (1), the train cover constraint, ensures that every train is assigned a qualified crew to operate it.

Constraint (2) ensures flow balance at a crew-supply node, while Constraint (3) ensures flow balance at the sink node.

Constraints (4) and (5) ensure flow balance at train-departure and train-arrival nodes, respectively. The flow balance constraints at a train-arrival node ensure that the crew assigned to a train is subsequently assigned to a rest arc, a deadhead arc, or a sink arc that emanates from the arrival node of the train. Flow balance constraints at a train-departure node ensure that the crew assigned to the

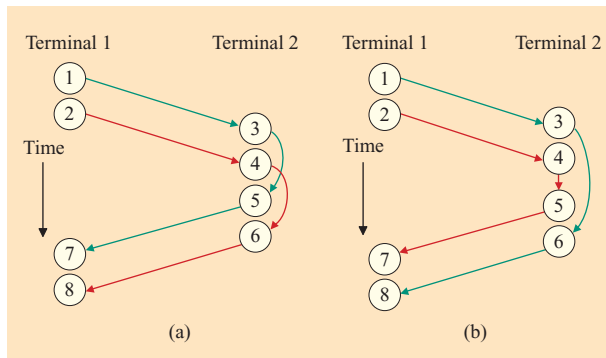


Figure 3

The FIFO rule: (a) Valid assignment; (b) invalid assignment.

train has been assigned to a rest arc, a deadhead arc, or a supply arc that terminates at the departure node of the train. Constraint (6) ensures that the crew assignment honors the FIFO constraint. Constraints (7), (8), and (9) specify that all of the decision variables in the model are binary. The objective function is constructed to minimize the total cost of crew wages, deadheading, detentions, and train delays. Note that the detention and delay costs are taken into account while calculating the cost of rest arcs.

Now we show how Constraint (6) enforces FIFO requirements. **Figure 3** illustrates crew assignments in two situations: one in which FIFO is satisfied and the other in which FIFO is violated. In case (a), the crew on the train traveling on arc (1–3) arrives at terminal 2 first and also leaves first, and hence FIFO is satisfied. In case (b), FIFO is violated because the crew on the train traveling on arc (1–3) enters terminal 2 first but leaves after the other crew. Therefore, in the solution, if there is flow on arc (4, 5), there should not be any flow on arc (3, 6).

Let us consider the following cases for Constraint (6) with respect to flow on arc (4, 5):

- Case 1: $x_{(4,5)} = 1$: The constraint becomes

$$\sum_{r' \in A_{(4,5)}} x_{r'} \leq 0 \Rightarrow x_{r'} = 0 \quad \forall r' \in A_{(4,5)}.$$

This ensures that if there is flow on rest arc (4, 5), there cannot be flow on any arc belonging to the prohibited set $A_{(4,5)}$, and hence there will not be any flow on arc (3, 6).

- Case 2: $x_{(4,5)} = 0$: The constraint becomes

$$\sum_{r' \in A_{(4,5)}} x_{r'} \leq M \quad \forall r' \in A_{(4,5)},$$

which essentially means that the constraint is relaxed.

Let us now estimate the size of a typical instance of the CSP in a crew district. Most crew districts have two terminals, and a typical train schedule has approximately 500 trains running over the course of several weeks in a crew district. Each crew district could have two to four crew types and approximately 50 crews. Therefore, the space-time network could have approximately $50 + 2 \times 500 = 1,050$ nodes. The number of arcs in the network could be very large if we construct all feasible rest arcs and deadhead arcs. To restrict the number of arcs constructed, we place a limit on the maximum duration of rest arcs. For example, if the train schedule stretches over a period of ten days, it is unrealistic for a crew to rest for more than three days. In this case, we can restrict the maximum rest arc duration to three days. After the space-time network of a typical problem is pruned on the basis of this rule, the number of deadhead arcs is typically approximately 25,000, and the number of rest arcs is approximately 100,000.

Because the number of rest arcs for a typical problem is of the order of 100,000 and each rest arc has one FIFO constraint, the number of FIFO constraints in the model would be 100,000. With the number of FIFO constraints that large, we would be losing one of the main advantages of the network flow formulation—that by honoring all business rules in the network construction phase, we keep the number of constraints small. Our computational results also confirm that handling FIFO constraints explicitly in this manner makes the problem computationally intractable.

Let us now consider the IPF in which we relax Constraint (6), the FIFO constraint; we call this problem the *relaxed problem*. This problem typically has more than 100,000 variables and several thousand constraints, which makes it a large optimization problem in itself. Integer programs of this size are usually very difficult to solve to optimality or near-optimality in a reasonable amount of time, but we were able to solve this problem to optimality in a matter of minutes using the branch-and-bound-based MIP solver ILOG CPLEX** 9.0. We believe that this is due to the special structure of the relaxed problem, which helps shorten the solution time significantly. All variables in the formulation are binary variables, and this leads to the MIP engine exploring fewer branches on the branch-and-bound tree compared with the case in which variables are integer variables. Whenever the engine branches on a noninteger variable, the value is set to 0 on one branch and to 1 on the other. Hence, at each level of the tree, the value of one variable is pre-fixed and can be eliminated from the model. Consequently, it is very likely that a feasible integral solution is obtained early in the process, and nodes in the branch-and-bound tree are fathomed much earlier than while solving a general integer program.

Another benefit of the network-flow-based approach is that even though we do not explicitly model each crew, the space-time network and the constraints are such that we can easily extract from the final solution of the model the set of trains a crew takes over the entire planning horizon. To do this, we start at the supply node of a particular crew and identify a path from this supply node to the sink node that has positive flow on it. Note that because of the commodity-specific flow balance constraints at each node, every crew has a unique path with positive flow from its supply node to the sink node.

Theorem 1

There is a one-to-one correspondence between a feasible flow on the space-time network satisfying Constraints (1)–(9) and a feasible solution to the CSP.

Proof

Consider a feasible flow on the space-time network. We have seen how the path of each crew can be extracted from the solution using a simple run-through procedure. Because of the network construction methodology, the extracted path of each crew has to satisfy all business and contractual rules. Hence, we see that every feasible solution on the space-time network corresponds to a feasible crew schedule. We can also show that the reverse transformation from a feasible crew schedule to a feasible flow on the space-time network is possible, thus establishing the result.

Thus, we have shown the one-to-one correspondence between feasible solutions to the IPF and feasible solutions to the CSP, and have therefore established the validity of our integer programming approach. In the next section, we describe various algorithms to solve the CSP which are centered on handling FIFO constraints in a computationally efficient manner.

Solution approaches

In this section we present our approaches to solve the CSP. Since the FIFO constraints are the ones that complicate the nature of the IPF, our solution approaches are centered around effective ways to handle this constraint. We develop a constraint-generation-based exact approach and a cost-perturbation-based heuristic approach to solve the problem. While the constraint-generation-based approach performs significantly better than the direct approach to solve the IPF, its application in a real-time environment may be restricted by long running times. On the other hand, the cost-perturbation scheme produces good-quality FIFO-compliant solutions very efficiently and hence is better suited for the real-time environment.

Successive constraint generation (SCG) algorithm

The SCG algorithm works by iteratively pruning out crew assignments that violate the FIFO constraints from the current solution of a more relaxed problem. We considered two methods for implementing constraint generation: 1) a branch-and-bound algorithm in which constraints are added to the linear programming (LP) relaxation that is solved at each node of the branch-and-bound tree until FIFO violations are eliminated (branch-and-cut); and 2) an iterative method in which we run a branch-and-bound algorithm on the relaxed problem, solve it to optimality, and then add constraints to remove infeasibilities. This is followed by another run of branch-and-bound on the more constrained problem, and so on.

As a result of our deliberation, we chose to implement the second method over the first, for the following reasons:

- Because the LP relaxation of the relaxed problem can have fractional flows on the rest arcs, the number of rest arcs with positive flow in the LP relaxation will be greater than the number of rest arcs with positive flow in an integral version. Also, the greater the number of rest arcs with positive flow, the greater the possibility of FIFO violations. Hence, more FIFO constraints are likely to be added to a non-integral solution. SCG that is implemented using the second method allows us to stop at any point when we feel that the level of FIFO infeasibility is reasonably small. We are able to do that because, after the addition of a set of constraints, we obtain an integral solution at regular intervals of less than a minute. On the other hand, in the branch-and-cut method, the addition of constraints at a node on the branch-and-bound tree would only guarantee FIFO compliance of the LP relaxation, which is not an integral solution in general. Hence, we do not have the option to prematurely terminate until we reach a point at which we obtain at least one integral solution to the LP relaxation. Consequently, we do not have control over the quality of intermediate solutions in terms of the number of FIFO violations.
- We show in our computational results that QCP does an excellent job of enforcing FIFO constraints for the current set of business rules, but we also mention that QCP does not guarantee FIFO compliance when there is priority in assigning crews to trains. We believe that the real benefit of SCG could come from being used in conjunction with QCP. In this approach, we would first apply QCP to obtain a solution with very few FIFO violations. SCG would

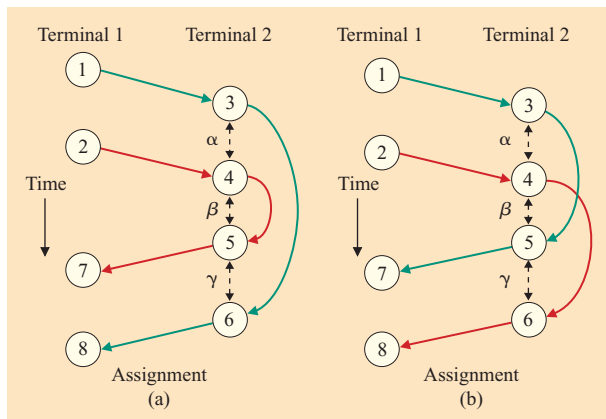


Figure 4

FIFO assignments: Crew assignments are made (a) in a non-FIFO manner; (b) in a FIFO manner.

then be applied on this solution to prune out the small number of remaining infeasibilities. A branch-and-cut approach in this context would be unnecessarily complicated because, when a small number of constraints are added, the problem can be reoptimized within a few seconds using SCG.

The SCG algorithm starts with the optimal solution of the relaxed problem, which may have several violations of the FIFO rule. In each iteration, the algorithm scans the rest arcs in the current solution that have positive flow, and for each such rest arc assignment that violates FIFO constraints, it adds the corresponding FIFO constraints. We then re-solve the problem and recheck for FIFO infeasibilities. This process is repeated until all FIFO infeasibilities are removed.

The SCG algorithm is as follows:

- Step 1: Solve the relaxed problem. If a feasible solution exists, proceed to Step 2. Otherwise stop, because the problem is infeasible.
- Step 2: Examine all of the rest arcs with positive flow in the solution of Step 1. Add FIFO constraints to the integer program on those rest arc assignments that violate FIFO requirements.
- Step 3: If FIFO constraints are added in Step 2, reoptimize the modified integer program and go to Step 2. Otherwise stop, because an optimal solution has been produced.

Note that the final solution of SCG satisfies all of the constraints of the IPF, and the constraints of SCG are a subset of the constraints of IPF. Hence, the SCG

algorithm is an exact algorithm guaranteeing an optimal solution to the original problem. However, in the worst case, SCG could add all of the FIFO constraints to the integer program and would thus become an intractable approach. Fortunately, this seldom happens in practice. Our computational results show that the number of constraints added is usually much smaller than the total number of rest arcs in the network.

While the SCG is an exact algorithm and produces provably optimal solutions, the running time of this algorithm can be quite high. In some instances during our computational experiments, SCG had a running time of the order of minutes, while in others, it had a running time of the order of hours. While such running times are acceptable in the planning environment, they would restrict the applicability of this algorithm in the real-time environment. In the next section, we describe a cost-perturbation-based algorithm that produces very good-quality FIFO-compliant solutions with running times comparable to those of the relaxed problem.

Quadratic cost perturbation (QCP) algorithm

In the previous section, we describe an SCG-based approach to remove the FIFO violations iteratively. In this section, we present an algorithm that penalizes the FIFO violations in a solution. We show that this method guarantees zero FIFO violations in the case in which there is no priority in assigning crews to trains, and it serves as a heuristic method for the case in which there are priority restrictions. Cost perturbation not only enforces FIFO constraints but also retains the special network flow structure of the problem, leading to fast computational times. The basic intuition behind this approach is that we perturb the costs of arcs while solving the relaxed problem in such a way as to guarantee FIFO compliance.

Figure 4 presents our cost-perturbation strategy for the case in which there is only one crew pool type. Consider the case in which crews are detained at Terminal 2. Then, due to the nature of detention costs, the cost of the assignment made in the FIFO manner [Assignment (b)] would definitely be less than or equal to the cost of Assignment (a), and hence the solution to the relaxed problem would honor FIFO constraints. On the other hand, suppose that all of the rest arcs had a cost of zero; then both assignments would have the same cost, and the relaxed problem would have no cost incentive to choose Assignment (b) over Assignment (a). Thus, a solution to the relaxed problem may violate the FIFO constraints. In order to provide an incentive to the relaxed problem to choose case (b) over case (a), we perturb the cost assignments on rest arcs so that the solution of the relaxed problem has assignments of type (b) instead of type (a).

The cost perturbation scheme that we use is a function of the duration of rest arcs. Suppose that the time duration between events corresponding to nodes 3 and 4, 4 and 5, and 5 and 6 are α , β , and γ , respectively. Consider a cost assignment that is proportional to the square of the duration of rest arcs. The constant of proportionality is represented by k . Then,

$$\begin{aligned} \text{Cost of Assignment (a)} &= k[\text{duration of arc (3, 6)}]^2 \\ &\quad + k[\text{duration of arc (4, 5)}]^2 \\ &= k(\alpha + \beta + \gamma)^2 + k\beta^2 \\ &= k(\alpha^2 + 2\beta^2 + \gamma^2 + 2\alpha\beta \\ &\quad + 2\beta\gamma + 2\gamma\alpha) \end{aligned}$$

and

$$\begin{aligned} \text{Cost of Assignment (b)} &= k[\text{duration of arc (3, 5)}]^2 \\ &\quad + k[\text{duration of arc (4, 6)}]^2 \\ &= k(\alpha + \beta)^2 + k(\beta + \gamma)^2 \\ &= k(\alpha^2 + 2\beta^2 + \gamma^2 + 2\alpha\beta + 2\beta\gamma). \end{aligned}$$

It can be observed that the cost of Assignment (b) is less than that of Assignment (a). Hence, when the rest arcs have zero costs, the quadratic cost perturbation scheme in the relaxed problem will give FIFO-compliant assignments when there is only one crew pool type. The observation made here can also be generalized for multiple crew pools unless there is a priority of crew pools in assignments to trains. If there is a priority assigned to crews in train assignments, a crew can have a FIFO-violated assignment to gain the priority assignments. We state our observations here as the following theorem.

Theorem 2

QCP applied to the relaxed problem guarantees FIFO-compliant crew assignments if there is no priority in assigning crews to the trains.

Proof

In the space-time network, rest arcs may have one of three costs assigned to them: zero cost, detention cost, or train delay cost. If, for example, all of the rest arcs in Figure 4 have zero cost, as shown above, the relaxed problem chooses the FIFO-compliant assignment because it is less expensive. If the rest arcs in Figure 4(a) have detention costs on them, the FIFO assignment shown in Figure 4(b) has either the same or a lesser level of detention. Hence, the perturbation scheme will work in this case as well. A similar argument would also work for train delay costs because FIFO assignments will always

have train delays equal to or less than those of non-FIFO assignments.

Since we do not want to change the cost structure of the original problem to a large extent, we set the value of k to a very small value and perturb the cost of each rest arc by a value that is computed as described above. Our computational tests (presented below in the section on comparison of algorithms) show that this method works very well, and that the solutions produced by QCP are indeed FIFO-compliant in the case in which there are no priorities. The solution time of this method is very short and is comparable to that of the relaxed problem. Note that in the case in which there are priorities, this approach can be used to obtain a solution with a small number of violations, and then the SCG algorithm can be used to prune out these violations. Another interesting observation is that for most of the instances tested, this method produces solutions with objective function values that are the same as those for the relaxed problems. This implies that FIFO constraints can be satisfied with little or no impact on the solution cost. Hence, using this approach, we are able to obtain an excellent quality of solutions using much less computational time. Because of its attractive running times and high solution quality, this method has the potential to be used in both planning and real-time environments.

Significance and uses of the model

The crew-scheduling model has applications in the tactical, planning, and strategic environments. In this section, we elaborate and provide specific examples of ways in which the model can be used as an effective tool for decision making.

Tactical crew scheduling

The defining problem in tactical crew scheduling is determining which crew should be assigned to operate each train. However, a number of subproblems and issues must be considered before crews are assigned to trains. Railroads have around-the-clock crew-calling centers that are responsible for monitoring the status of each crew and the status of each train and anticipating when a particular crew should be called to operate a particular train. A typical crew-calling center employs 200 to 300 crew callers who call crews and answer inbound telephone queries from management and the crews. First, a crew caller looks at the projected lineup (crew assignment) of outbound trains at a particular crew-change location. With a projection of train departure times, say 13:30, 15:00, and 16:00, the crew caller then goes through a number of checks before assigning a crew to a train; the checks determine such factors as whether the train is covered by a designated assigned pool or by FIFO assignment from the general pool and when the

next qualified crew will be rested and available to operate this train. The actual rules are very complex, and the combinations of solutions that must be considered can overwhelm a person.

Our model has several applications in the tactical scheduling environment. Some of these applications are given below:

- *Assignment of crews to trains*—The output of our model tells us how to assign crews to trains.
- *Recommending which crews to place in hotels and which crews to deadhead home*—When a crew arrives at an away terminal, the crew callers must decide whether the crew should deadhead back home or go to a hotel for rest. The model can be used to mathematically look ahead and systematically make the tradeoffs among different cost categories of crew wages, deadheads, detentions, and rest violations.
- *Minimizing the number of trains delayed because of shortage of crew*—Train delays are potentially very costly because they may lead to the unavailability of crew to operate another train in the future and may have a negative domino effect on network-wide operations. By creating several deadhead arcs while constructing the space-time network, we ensure that such a situation is avoided.
- *Disruption management*—The crew-scheduling model can be used as a tool to return disrupted operations to normalcy. Suppose that at some point in time the operations are disrupted. The current state or snapshot of the system gives us the location of each crew and the hours of duty already served. Using this information and the information about the future train schedule, the crew-scheduling models can be used to optimally reassign crew to trains.

Crew planning

The essence of the crew-planning problem for operations or planning is to determine the number of crews that should be in each crew pool. It can be noted that since each position is guaranteed a minimum number of work hours per month, it is quite costly to overestimate the number of positions required to staff a pool. Currently, railroads solve the pool-sizing problem on the basis of historical precedent and rules-of-thumb, through negotiation with the union, and by trial and error. The network flow model can satisfy the need for a structured approach that captures all of the considerations, quantifies the various costs, and recommends the best way to define and staff crew pools. Some of the applications of the model in the planning environment are described in the following paragraphs.

Developing and evaluating crew schedules—The crew-scheduling model can be used to compare the current crew schedule with the model-generated schedule on the basis of several criteria, such as average rest time at the home location, average rest time at the away location, and average deadhead time. By suitably changing the model cost parameters, we can obtain schedules with different characteristics. For example, if we want to minimize detention, we can set the detention cost to a very large value and run the model.

Varying the size of crew pools—Using the crew-scheduling model, we can study the impact of varying the crew pool size on the quality of the solution. For example, suppose our objective is to minimize the number of crew used. While formulating the problem, we give large cost incentives to flow on the sink arcs from crew-supply nodes to the sink node. This would lead to the model producing a crew schedule that uses the minimum number of crew.

Crew strategic analysis

Strategic management involves the development of policies and plans and the allocation of resources to implement these plans. The timeframe of strategic management extends over several months or years. Strategic crew problems include forecasting future headcount needs and evaluating major policy changes, such as negotiating changes to trade-union rules or changing the number and location of crew change points on a network. The railroad industry is now experiencing unprecedented traffic growth. Therefore, it is very important to be able to quantify the expected impact on manpower needs as traffic grows, because it takes 18 to 24 months to hire, train, and qualify train crew personnel. Recently, corporate strategists have been struggling with the tradeoff between crew costs and train delays. Our model can be used to quickly calibrate efficient frontiers for each crew district and determine the number of crews that minimizes the sum of train delay costs and crew costs. If railroad management is dissatisfied with that level of train performance, the cost of train delay can simply be increased, and the model will request additional crews so that a new cost-minimizing solution is obtained.

Some of the applications of the network flow model in the strategic environment are given in the following paragraphs.

Determining the number and territory of crew districts—We can use the crew-scheduling model to reoptimize and test different crew district configurations. For example, suppose crew district 1 operates trains between locations A and B, and crew district 2 operates trains between locations B and C. Merging all three stations into a single

Table 2 Comparison of algorithmic performance.

Weeks	Crew pools	Relaxed problem			Exact IPF				SCG				QCP		
		Cost (\$)	Time (s)	FIFO constraints violated	Cost (\$)	Time (s)	FIFO constraints		Cost (\$)	Time (s)	FIFO constraints		Cost (\$)	Time (s)	FIFO constraints violated
							Added	Violated			Added	Violated			
1	1	130,952	10.8	73	-	3,600	11,062	N/A	132,022	2,015	958	25	130,952	10.9	0
2	1	265,284	30.3	148	-	3,600	23,527	N/A	267,067	1,981	1,492	95	265,284	31.4	0
3	1	399,816	57.2	225	-	3,600	35,976	N/A	399,816	1,908	1,657	151	399,816	60.0	0
4	1	531,378	91.8	274	-	3,600	48,797	N/A	532,091	2,326	1,805	226	531,378	97.4	0
1	2	132,495	17.7	64	-	3,600	17,999	N/A	132,495	347.5	478	0	132,495	17.7	0
2	2	267,130	55.6	118	-	3,600	40,623	N/A	267,316	2,423	1,068	0	267,221	60.9	0
3	2	402,045	112.0	173	-	3,600	63,215	N/A	405,227	4,858	1,321	25	402,678	125.0	0
4	2	533,694	187.3	226	-	3,600	86,477	N/A	538,039	3,928	1,745	25	534,327	210.7	0

crew district could give us better opportunity to optimize costs.

Evaluating the effects of changing crew trade-union rules—The CSP is a complex optimization problem because of strict trade-union rules related to crew operation. Changes to any of these rules will likely face considerable resistance from the labor union. At the same time, such changes have the potential to affect crew costs substantially. Using the crew-scheduling model, we can evaluate the impact of changing the trade-union rules on the crew cost. For example, suppose we want to know the impact of changing the mandatory rest time at home from twelve hours to ten hours. We can run the model with the parameter setting of ten hours and measure the change in crew cost.

Forecasting crew requirement—On the basis of the forecasted train schedule, we can use the model to help us forecast crew requirements. We first run the model assuming that a very large number of crews are available. Because the crew supply is much greater than required, many crews will flow directly from the crew supply to the sink node. The total crew supply minus the number of unused crews will give an idea of the number of crews required, based on the forecasted train schedule.

In this section, we have seen that the crew-scheduling model has several real-world applications in the tactical, planning, and strategic environments. If put into production, the model has the potential to enable railroad professionals to improve their day-to-day operations and to plan effectively to achieve their long-term organizational goals.

Computational results

In this section, we present computational results of our algorithms on several problem instances and a case

study of a representative instance. We implemented our algorithms in Microsoft Visual Basic** programming language and tested them on the data provided by a major Class I railroad. We modeled our integer programs using ILOG Concert Technology** 2.0 modeling language and solved them using the CPLEX 9.0 solver. We conducted all computational tests on a 2.4-GHz Intel Pentium** 4 processor with 512 MB RAM.

Comparison of algorithms

In this section, we compare the performances of the relaxed problem, the exact IPF, the SCG algorithm, and the QCP algorithm in several real-world instances. Our problem instances consist of train schedules over a period of one to four weeks. In one instance, the number of crew pools is 1, making the problem a single-commodity flow problem. In the other set of instances, the number of crew pools is 2, and the problem is formulated as a multicommodity flow problem. For each instance, we measure the solution cost, the solution time, the number of FIFO constraints added to the formulation, and the number of FIFO constraints violated in the solution. It can be noted that no FIFO constraints are added while solving the relaxed problem and the QCP. The results of our computational tests are presented in **Table 2**.

We have reached the following conclusions from the results:

- The solutions to the relaxed problem have the highest number of FIFO violations, but the solution times are the fastest.
- The IPF has several thousand FIFO constraints. These constraints make the problem computationally intractable, and we could not obtain a feasible

Table 3 Effect of varying crew pool.

<i>Crew available</i>	<i>Crew used</i>	<i>Deadheads (hr)</i>	<i>Detention (hr)</i>	<i>Train delay (hr)</i>	<i>Solution cost (\$)</i>	<i>Increase in cost (\$)</i>
42	31	38	37.00	8.77	262,838	-
40	30	38	37.00	8.77	262,838	0
38	29	38	37.00	8.77	262,838	0
36	29	40	37.00	7.85	263,340	502
34	29	40	37.00	7.85	263,340	0
32	28	40	37.00	7.85	263,340	0
30	28	41	37.00	7.85	263,697	357
28	28	41	37.00	7.85	263,697	0
26	26	43	30.65	30.38	268,704	5,007
24	24	43	17.50	154.83	295,486	26,782
22	22	44	6.37	417.12	354,610	59,115
20	-	-	-	-	Infeasible	-

solution for any of the instances in one hour of computational time.

- The SCG algorithm starts with the solution to the relaxed problem as the initial solution and progressively reduces the number of infeasibilities. However, the amount of computational time taken by this algorithm is still quite large. We were able to obtain a FIFO-compliant solution for two instances; for all other instances, we terminated the algorithm when the iteration that was running during minute 30 of computational time was complete.
- The QCP algorithm produces FIFO-compliant crew schedules for all instances. Also, in six instances out of eight, the objective function values are equal to that of the relaxed problem. As the relaxed problem provides a lower bound to the optimal solution, the QCP algorithm produces the optimal solution in six instances out of eight, and the optimality gap is less than 0.2 percent for the other two instances. This algorithm also has very fast solution times, which are comparable to that of the relaxed problem.

Thus, we conclude that the QCP algorithm outperforms the other algorithms in both solution quality and solution time. It produces optimal or near-optimal solutions in a few minutes of running time and therefore has the potential to be used in both the planning and real-time environments.

Case study

In this section, we conduct a case study to illustrate how the crew-scheduling model can be used to derive useful

information and support decision making at a railroad. We perform the case study on a representative two-week dataset that has 326 trains, two crew pools, and 48 crews, and we run the computational tests using the QCP algorithm. The various aspects of the problem that we observe in this case study are discussed in the following sections.

Effect of varying the number of available crews

In this study, we quantify the effect of varying the number of available crews on the overall solution quality. We start with a set of 42 available crews and reduce the number of crews available until the problem becomes infeasible. **Table 3** presents the computational results, and **Figure 5(a)** shows the relationship between the number of available crews and solution cost.

We can make the following observations from this study:

1. As the number of available crews decreases, the model attempts to compensate for the lack of crews by increasing the level of deadheading and train delays.
2. Initially, reducing the number of available crews has no adverse effect on the solution cost, but as more crews are removed, the solution cost rises steeply. For example, reducing the number of crews available from 42 to 26 (a reduction of 38 percent) has an insignificant impact on the solution cost, but reducing the number of crews from 24 to 22 leads to the solution cost increasing by more than \$59,000 (an increase of 20 percent).

Table 4 Effect of varying deadhead cost.

Deadhead cost/hr (\$/hr)	Deadheads	Detention (hr)	Train delay (hr)	Solution cost (\$)
0	42	34.55	5.45	253,079
100	38	37.00	8.77	260,051
200	38	37.00	8.82	266,396
300	37	40.33	9.48	272,733
400	37	40.33	9.57	278,918
500	36	40.33	13.13	284,955
600	36	40.33	13.13	290,955
700	36	40.33	13.13	296,955
800	36	40.33	13.13	302,955
900	36	40.33	13.18	308,967
1,000	35	36.80	22.95	314,935
10,000	33	36.80	55.13	813,771

The objective function in this case study is not a function of the number of crews used; it is only a function of total deadhead, detention, and delay. This is why solutions using different numbers of crews have identical costs provided that their total deadhead, detention, and delay are the same.

Effect of varying deadhead cost

In this study, we quantify the effect of varying deadhead cost on the number of deadheads, total detention hours, total train delay hours, and overall solution cost. The default cost of deadheading used by the railroad is \$144 per hour. We start with a deadhead cost of \$0 per hour and then progressively increase deadhead cost while measuring the impact on the solution, as shown in **Table 4**.

We can make the following observations from this study:

1. As the deadhead cost increases, the number of deadheads in the solution decreases. However, after a certain point, there is no significant decrease in the number of deadheads. For example, even for a very high deadhead cost of \$10,000, the solution has 33 deadheads. From this observation we can conclude that there is an inherent imbalance in the system that necessitates deadheading.
2. As the deadhead cost increases, the solution of the model has fewer deadheads and more train delays. This behavior of the model provides the insight that if the deadhead cost increases at some point in time, the railroad must adapt by allowing far more flexibility in terms of train delays. Alternatively, the

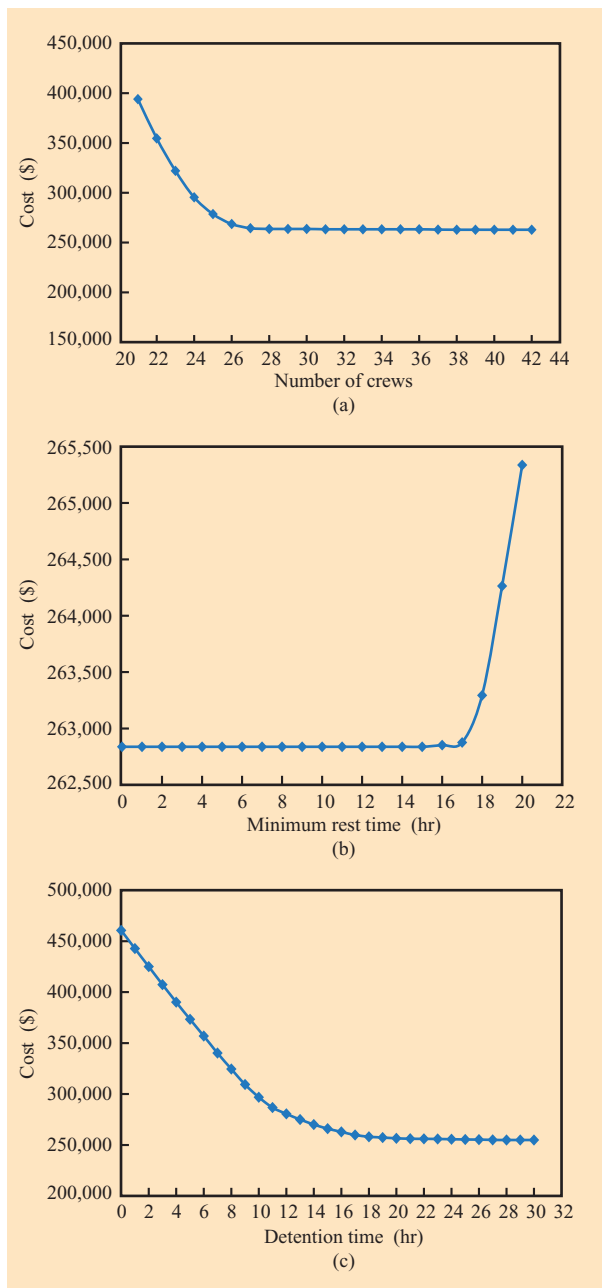


Figure 5

Case study results: Solution cost as a function of (a) number of crews; (b) minimum rest time at home; (c) detention time.

management can also negotiate with crew unions and reduce the minimum rest time requirements.

Effect of varying minimum rest time at the home location

In this study, we quantify the effect of varying the minimum rest time at the home location on the average rest time at the home location, train delays at the home

Table 5 Effect of varying minimum rest time at the home location.

Minimum rest (hr)	Average rest at home (hr)	Train delays at home (hr)	Average rest away (hr)	Train delays away (hr)	Solution cost (\$)
0	10.02	0.00	12.83	8.77	262,838
2	11.60	0.00	12.99	8.77	262,838
4	12.86	0.00	13.12	8.77	262,838
6	14.37	0.00	13.12	8.77	262,838
8	15.16	0.00	13.15	8.77	262,838
10	16.81	0.00	13.31	8.77	262,838
12	18.51	0.00	13.33	8.77	262,838
14	20.48	0.00	13.25	8.77	262,838
16	21.53	0.07	13.09	8.77	262,853
18	23.98	1.23	13.18	9.52	263,294
20	28.33	3.78	13.09	17.40	265,337

Table 6 Effect of varying detention cost.

Deadhead cost/hr (\$/hr)	Detention (hr)	Deadheads (hr)	Train delay (hr)	Solution cost (\$)
0	305.45	35	1.17	254,840
40	64.57	37	3.32	258,630
80	40.33	37	8.77	260,528
120	37.00	38	8.77	262,098
160	34.55	39	8.77	263,542
200	30.77	39	11.97	264,904
240	23.32	39	18.50	265,849
280	23.32	39	18.50	266,782
320	18.67	39	24.18	267,534
360	11.10	39	35.53	268,167
400	1.93	40	49.23	268,452
500	1.93	40	49.23	268,645
600	1.93	40	49.23	268,838
700	1.93	40	49.23	269,032

location, average rest time at the away location, train delays at the away location, and the overall solution cost. The default value of minimum rest time used by the railroad is 12 hours. We start with a minimum rest requirement of zero hours and progressively increase the

Table 7 Effect of varying detention time.

Detention time (hr)	Average rest at away location (hr)	Detention (hr)	Solution cost (\$)
0	8.95	1,136.60	460,558
4	8.95	633.18	390,079
8	10.41	331.60	324,478
12	11.12	83.17	280,491
16	13.33	37.00	262,838
20	14.20	3.67	256,561
24	14.78	1.52	255,620
28	15.82	0.00	254,840

value of this parameter while measuring the impact on the solution, as shown in **Table 5** and **Figure 5(b)**.

From this study we observe that the minimum rest time at home can be increased to 16 hours without a significant increase in the solution cost. However, any increase beyond 16 hours leads to a steep increase in the solution cost. The railroad management can use these inputs to effectively negotiate rest times with the union. For example, if the union wants the minimum rest time to be increased from 12 hours to 14 hours, the management can use the model to quantify the impact of this change and negotiate appropriately.

Effect of varying detention cost

The railroad pays detention charges for each hour of crew rest beyond 16 hours at an away location. In this section, we quantify the effect of varying the detention cost on the total detention hours, number of deadheads, total train delay hours, and overall solution cost, as presented in **Table 6**. The default value of detention cost used by the railroad is \$140 per hour.

We make the following observations from this study:

1. As the detention cost per hour increases, the number of detention hours in the solution decreases.
2. As the detention cost per hour increases, the solution has a greater number of deadheads and train delays. This behavior of the model provides the insight that if the detention cost increases at some point in time, the railroad must adapt by allowing more flexibility in terms of train delays and crew deadheading.

Effect of varying detention time

In this study, we quantify the effect of varying the detention time (the minimum rest time at the away location after which a crew becomes eligible for a detention

allowance) on the average rest time at the away location, detention hours, and overall solution cost. The results are shown in **Table 7** and **Figure 5(c)**. The default value of detention time provided by the railroad is 16 hours.

This study shows that increasing the detention time has an impact on the solution cost, but it diminishes as the detention time increases. We observe that increasing the detention time from 0 to 20 hours reduces the solution cost, but increasing it beyond 20 hours has almost no impact on the solution quality.

Summary and conclusions

We have described a network-flow-based approach to solve the railroad CSP in the context of North American railroads. The CSP for North American railroads is governed by several FRA regulations and trade-union work rules. To develop a good crew schedule, in addition to satisfying these regulations, the total wage costs, train delay costs, deadhead costs, and detention costs must all be minimized. The railroad divides the network into a number of crew districts, and each crew district has several crew pools. Each crew pool at a district can have a different set of operating rules. These factors make this a complex problem to model and solve.

The network flow formulation for the CSP developed in this paper is both flexible and robust, and it can easily be manipulated to represent each of the possibilities encountered in real life. We formulate the CSP as an integer program on a space-time network. The network is constructed in such a way that all FRA regulations and trade-union work rules other than FIFO constraints are enforced during the network construction phase itself. The operational constraints are handled in the IPF. We develop two approaches to handle FIFO constraints. The first is an SCG approach in which constraints are generated iteratively to cut out FIFO violations. The second approach, QCP, relies on perturbing the objective function to generate FIFO-compliant solutions. We provide computational results comparing the performance of various approaches and show that the perturbation approach outperforms the other approaches in terms of both solution time and solution quality.

The crew-scheduling model has applications in a wide range of settings. We describe several applications of the model in the tactical, planning, and strategic environments. The broad spectrum of applications varies from the short-term problem of assigning crews to trains over the next few days to the long-term problem of forecasting crew requirements on the basis of future demand patterns. The model provides railroad executives with a method to calibrate and quantify the impact of current decisions on future operations by running several “what-if” scenarios.

We believe that this research will eventually lead to the deployment of crew-planning models and algorithms at

North American railroads, replacing the current manual process and that, in doing so, it will have a significant impact on railroad on-time performance, crew utilization, and productivity, while also improving the quality of life for crew and improving railroad safety.

Acknowledgments

We thank Larry A. Shughart from Innovative Scheduling, who has several years of railroad experience, for helping us understand the real-world significance of crew scheduling and for providing us with valuable business insights. We also thank the anonymous referees for their insightful comments and suggestions, which led to a significantly improved paper.

**Trademark, service mark, or registered trademark of ILOG, Inc., Microsoft Corporation, or Intel Corporation in the United States, other countries, or both.

References

1. “2004 Freight Rail Component of the Florida Rail Plan,” Florida Department of Transportation Report, 2005; see <http://www.dot.state.fl.us/rail/Publications/2004Plan/FreightComponentSmry.pdf>.
2. America’s Railroads—Employee Compensation, National Railway Labor Conference; see <http://www.raillaborfacts.org/issues-at-stake/wages-and-benefits.aspx>.
3. Surface Transportation Board, U.S. Department of Transportation, Bureau of Accounts, Employment Data, 2006; see <http://www.stb.dot.gov/econdata.nsf/322683bcf67f4143852566210062ac90?OpenView&Start=1&Count=300&Expand=1#1>.
4. A. A. Assad, “Analytical Models in Rail Transportation: An Annotated Bibliography,” *INFOR* **19**, No. 1, 59–80 (1981).
5. A. A. Assad, “Analysis of Rail Classification Policies,” *INFOR* **21**, No. 4, 293–314 (1983).
6. J.-F. Cordeau, P. Toth, and D. Vigo, “A Survey of Optimization Models for Train Routing and Scheduling,” *Transport. Sci.* **32**, No. 4, 380–404 (1998).
7. A. Wren, Editor, *Computer Scheduling of Public Transport*, North-Holland Publishing Company, Amsterdam, The Netherlands, 1981.
8. L. Bodin, B. Golden, A. A. Assad, and M. Ball, “Routing and Scheduling of Vehicles and Crews: The State of the Art,” *Computer & Oper. Res.* **10**, No. 2, 63–211 (1983).
9. P. Carraraesi and G. Gallo, “Network Models for Vehicle and Crew Scheduling,” *Euro. J. Oper. Res.* **16**, No. 2, 139–151 (1984).
10. T. H. Wise, “Column Generation and Polyhedral Combinatorics for Airline Crew Scheduling,” Ph.D. dissertation, Cornell University, Ithaca, NY, 1995.
11. J. Desrosiers, Y. Dumas, M. M. Solomon, and F. Soumis, “Time-Constrained Routing and Scheduling,” *Handbooks in Operations Research and Management Sciences: Network Routing*, Vol. 8, M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, Editors, Elsevier Science, Amsterdam, The Netherlands, 1995, pp. 35–139.
12. B. Gopalakrishnan and E. L. Johnson, “Airline Crew Scheduling: State-of-the-Art,” *Ann. Oper. Res.* **140**, No. 1, 305–337 (2005).
13. A. Caprara, M. Fischetti, P. Toth, D. Vigo, and P. L. Guida, “Algorithms for Railway Crew Management,” *Math. Program.* **79**, 124–141 (1997).
14. A. T. Ernst, H. Jiang, M. Krishnamoorthy, H. Nott, and D. Sier, “An Integrated Optimization Model for Train Crew Management,” *Ann. Oper. Res.* **108**, No. 1/4, 211–224 (2001).

15. R. Freling, R. M. Lentink, and A. P. M. Wagelmans, "A Decision Support System for Crew Planning in Passenger Transportation Using a Flexible Branch-and-Price Algorithm," *Ann. Oper. Res.* **127**, No. 1/4, 203–222 (2004).
16. C. Barnhart, E. L. Johnson, R. Anbil, and L. Hatay, "A Column Generation Technique for the Long-Haul Crew Assignment Problem," *Optimization in Industry: Volume II*, T. Ciriano and R. Leachman, Editors, John Wiley and Son, Chichester, England, 1994, pp. 7–22.
17. C. Barnhart, E. L. Johnson, G. L. Nemhauser, and P. Vance, "Crew Scheduling," *Handbook of Transportation Science*, R. W. Hall, Editor, Springer, New York, 2003, pp. 493–521.
18. S. C. K. Chu and E. C. H. Chan, "Crew Scheduling of Light Rail Transit in Hong Kong: From Modeling to Implementation," *Computer & Oper. Res.* **25**, No. 11, 887–894 (1998).
19. C. G. Walker, J. N. Snowdon, and D. M. Ryan, "Simultaneous Disruption Recovery of a Train Timetable and Crew Roster in Real Time," *Computer & Oper. Res.* **32**, No. 8, 2077–2094 (2005).
20. M. F. Gorman and M. Sarrafzadeh, "An Application of Dynamic Programming to Crew Balancing at Burlington Northern Santa Fe Railway," *Int. J. Services Technol. & Management* **1**, No. 2/3, 174–187 (2000).
21. R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall, Upper Saddle River, NJ, 1993.

Received October 2, 2006; accepted for publication December 15, 2006; Internet publication May 17, 2007

Balachandran Vaidyanathan *Department of Industrial and Systems Engineering, University of Florida, 303 Weil Hall, Gainesville, Florida 32611 (vbala@ufl.edu)*. Mr. Vaidyanathan is a doctoral candidate. He received his B.S. degree from the Indian Institute of Technology, Madras, in 2002. His interests include mathematical modeling and the design of algorithms to solve real-world problems. He has been awarded Honorable Mentions in the INFORMS 2004 and 2006 paper competition on Management Science in Railroad Applications sponsored by The Institute for Operations Research and the Management Sciences. In addition to application-oriented research, Mr. Vaidyanathan is also conducting research on developing algorithms for a special case of assignment and transportation problems, which has wide application to transportation networks.

Krishna C. Jha *Innovative Scheduling, Inc., 2153 S.E. Hawthorne Road, Suite 128, Gainesville, Florida 32641*. Dr. Jha is a director of research and development at Innovative Scheduling, Inc., specializing in mathematical modeling, state-of-the-art optimization techniques, algorithmic design and development, and software engineering. During his doctoral studies, he developed innovative very-large-scale neighborhood search algorithms for several combinatorial optimization problems. Dr. Jha is the recipient of the first prize in the INFORMS 2003 paper competition on Management Science in Railroad Applications, and the INFORMS 2006 Daniel H. Wagner Prize for excellence in operations research practice.

Ravindra K. Ahuja *Department of Industrial and Systems Engineering, University of Florida, 303 Weil Hall, Gainesville, Florida 32611 (ahuja@ufl.edu)*. Professor Ahuja is co-director of the Supply Chain and Logistics Engineering Center at the University of Florida, and president and CEO of Innovative Scheduling, Inc. He received his Ph.D. degree in industrial and management engineering from the Indian Institute of Technology, Kanpur. He visited the Sloan School of Management at the Massachusetts Institute of Technology to conduct research on the development of faster algorithms for network flow problems, a collaboration that resulted in the fastest available algorithms for most fundamental network flow problems and led to his coauthoring *Network Flows: Theory, Algorithms, and Applications*. In 1993 this book won the prestigious Lanchester Prize, awarded annually to the best publication in operations research. Dr. Ahuja is also a coauthor of two books on developing decision-support systems; he publishes widely and is an associate editor of *Transportation Science* and *Networks*. He won the INFORMS 2003 Pierskalla Award for best contribution of operations research in health applications, and the INFORMS 2006 Daniel H. Wagner Award for excellence in operations research practice.