Multi-Commodity Network Flow Based Approaches for the Railroad Crew Scheduling Problem

Balachandran Vaidyanathan
Department of Industrial & Systems Engineering,
University of Florida,
Gainesville, FL 32611
Phone: (352) 665-2821
vbala@ufl.edu

Abstract

In this paper, we study one of the most important railroad optimization problems, the *crew scheduling problem*, in the context of North American railroads. Crew scheduling for North American railroads is very different from that of European railroads, which has been well studied. The crew scheduling problem is to assign *crew* (train operators) to scheduled trains over a time horizon (generally a week) at minimal cost while honoring several operational and contractual requirements. Each North American Class I railroad spends at least a billion dollars in crew costs annually and does not have any decision support system available that can assist it in all levels of decision making: tactical, planning, and strategy. Indeed, all decisions related to crew are made manually, thereby leaving sufficient room for improvement. We have developed a network-flow based crew-optimization model that has applications in all levels of decision making in crew scheduling: tactical, planning, and strategy. Our network-flow model maps the assignment of crew to trains as the flow of crew on an underlying network where different crew types are modeled as different commodities in this network. We formulate the crew assignment problem as an integer-programming problem on this network, which allows this problem to be solved to optimality. We also develop several highly efficient algorithms using problem decomposition and relaxation techniques, where we use the special structure of the underlying network model to obtain significant speed-ups. 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.

Multi-Commodity Network Flow Based Approaches for the Railroad Crew Scheduling Problem

Abstract

In this paper, we study one of the most important railroad optimization problems, the *crew scheduling problem*, in the context of North American railroads. Crew scheduling for North American railroads is very different from that of European railroads, which has been well studied. The crew scheduling problem is to assign *crew* (train operators) to scheduled trains over a time horizon (generally a week) at minimal cost while honoring several operational and contractual requirements. Each North American Class I railroad spends at least a billion dollars in crew costs annually and does not have any decision support system available that can assist it in all levels of decision making: tactical, planning, and strategy. Indeed, all decisions related to crew are made manually, thereby leaving sufficient room for improvement. We have developed a network-flow based crew-optimization model that has applications in all levels of decision making in crew scheduling: tactical, planning, and strategy. Our network-flow model maps the assignment of crew to trains as the flow of crew on an underlying network where different crew types are modeled as different commodities in this network. We formulate the crew assignment problem as an integer-programming problem on this network, which allows this problem to be solved to optimality. We also develop several highly efficient algorithms using problem decomposition and relaxation techniques, where we use the special structure of the underlying network model to obtain significant speed-ups. 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.

1. 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. Crew scheduling problems consist 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 on one hand and to improve the quality of life for crew on the other hand. Improved quality of crew life leads to more productive employees, less employee turnover and much 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 comes close to meeting their needs. Although airline crew scheduling problems have been well studied and well solved, and railroad crew scheduling problems for European and Asian railroads have also been addressed to some extent, crew scheduling problems for North American railroads, due to various union and regulatory complexities, are unique and remain unsolved. 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 analysis. Henceforth in this paper, unless otherwise specified, the crew scheduling problem is referred to in the context of North American railroads.

U.S. freight tonnage is expected to double in volume over the next 20 years (Florida Department of Transportation Report [2005]). Railroad executives are very concerned about their ability to attract, train, and retain sufficient semi-skilled labor necessary 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 plus benefits) and yet 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. This is further complicated by the fact that crews are seldom assigned to trains based on a fixed schedule. Generally the company telephones the next available crew 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 and then return on another train as their turn is reached. 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 away from their homes and families. The irregular work-style of railroad crews makes attracting potential employees to this career harder.

Also, railroads are not very profitable, typically earning less than 10% 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 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 profitably grow their businesses. Labor costs, the largest component of a railroad's operating expense, require 36% of total revenue [The Labor Bureau, Inc., May 1996 Report to the Presidential Emergency Board]. Depending on the size of their 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 around 15,000 to 25,000 locomotive engineers, conductors, and brakemen [US DOT Surface Transportation Board, Bureau of Accounts, ICC Wage Form B]. 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 a railroad's on-time performance, crew utilization and productivity, while also improving both quality-of-life for crew and railroad safety.

In a large Class I railroad, various divisions are tasked with analyzing train crews. Each group is interested in a different aspect of crew planning and scheduling. These perspectives can generally be characterized based on the planning horizon of the issue at hand. Crew issues faced by railroads can be broadly classified into three categories: 1) *Tactical* – Decisions that must be made immediately to support real-time train operations. Tactical problems have a planning horizon of 24 to 48 hours. 2) *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. 3) *Strategic* – Decisions that must be made well in advance (more than a year) of implementation to ensure sufficient lead time is available to properly prepare and implement a new business practice. The models and algorithms proposed in this paper have applications in all these areas of decision making.

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 (Assad [1981, 1983], and Cordeau et al. [1998]). 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 due to Wren [1981], Bodin et al. [1983], Carraresi and Gallo [1984], Wise [1995] and Desrosiers et al. [1995]. All these articles explore a set covering based approach to solve the crew scheduling problem. Crew scheduling is conventionally divided into two stages: (1) *Crew pairing*: A crew pairing is a sequence of connected segments that start and end at the same crew base and satisfy all legality constraints. The objective is to find the minimum-cost set of crew pairings such that each flight or train segment is covered. (2) *Crew rostering*: The objective here 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 usually solved using column generation embedded in a branch-and-bound framework (also called branch-and-price).

The pairing and rostering approach has gained wide acceptance and application in the airline industry. Gopalakrishnan and Johnson [2005] in a recent survey paper 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. [1997], Ernst et al. [2001] and Freling et al. [2004] describe the application of this approach to railroad crew management. Caprara et al. [1997] 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. [2001] consider the crew scheduling problem faced by Australian railroads and develop an optimization model which constructs crew parings and rosters. While they consider several business rules, such as rest requirements, they still solve a relaxed version of the problem and mention the necessity for an exact method in their conclusions. Freling et al. [2004] 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. Other articles which describe applications of the pairing and rostering approach include Barnhart et al. [1994, 2003].

Other research in the area of railroad crew scheduling which use a different approach is due to Chu and Chan [1998] and Walker et al. [2005]. Chu and Chan [1998] consider the problem of crew scheduling for Light Rail transit in Hong Kong. They decompose the problem into two stages, the first one involving partitioning of driving blocks into pieces, and the second one involving the combination of pieces into runs. With their added localized optimization heuristics, they were able to solve the problem in less than half an hour of computational time. However, their approach could not model the problem completely and the solution could only be used as a guideline for crew schedule generation. Walker et al. [2005] 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 crew scheduling problems 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 due to Gorman and Sarrafzadeh [2000]. They studied crew balancing in the context of a major North American Railroad, Burlington Northern Santa Fe (BNSF) Railway and developed a dynamic programming approach to solve the problem. The major short-coming 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 which are *double-ended* or even more complex. The multi-commodity network flow approach described in this paper models all the rules considered by Gorman and Sarrafzadeh [2000] and also handles the case where different crew pools have different sets of rules. It is also applicable to all the crew district configurations encountered in North America (in Section 2, we describe these configurations).

From our extensive review of the literature, we see that crew pairing and rostering approaches which use column generation have been the predominantly successful method to solve crew scheduling problems. However, this approach cannot be used for North American railroads due to the following reasons:

1. The rail network of North American railroad is 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 not required since 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 200-300 crew districts and the emphasis is on an approach which is simple and fast and column generation techniques which are computationally very intensive are not appropriate.

2. The FRA regulations governing North American railroads are extremely complex. The most complicating of these rules is *First-In-First-Out* (FIFO) requirement. FIFO constraints require that crews should be called on duty in the order in which they become qualified for assignment at a location. The reader may note that none of the past research handles constraints of this kind. While this constraint is extremely easy to state, explicitly modeling these constraints make the problem computationally intractable. The success of all approaches using column generation or branch-and-price algorithms is dependant on the ease of solving the sub-problem. Addition of the FIFO side constraints to the problem would spoil the special structure of the sub-problem and blow up the computational times. Since our model needs to 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 hence the first of its kind and is therefore a novel contribution to the application of innovative optimization techniques to solve real-world business problems.

In this paper, we model the crew scheduling problem as a multi-commodity network flow problem on an underlying space-time network. In this model, *crew pools* (set of crews governed by same business rules in a crew district) represent commodities, and the flow of individual crew represents their assignments. The space-time network is constructed in such a way that flow of crew automatically satisfies all FRA regulations and trade-union rules other than the *First-In-First-Out* (FIFO) requirement. We formulate the crew scheduling problem as an Integer Programming Formulation (IP) on a space-time network where FIFO constraints are modeled as side constraints to the multi-commodity flow problem. We show that solving the IP formulation 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 crew scheduling problem with relaxed FIFO constraints the *Relaxed Problem*, and a solution to this problem provides a lower bound to the optimal solution of the crew scheduling problem. We develop an algorithm, called *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 another algorithm, called *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 less than 0.2% gap for a few cases, with running times in the order of minutes.

Our major research contributions in this paper are listed below:

1. We develop a space-time network construction algorithm so that the flow of crews on this network automatically satisfies all the FRA regulations and trade-union rules other than the First-In-First-Out (FIFO) requirement. The network-construction procedure is flexible enough to handle several combinations of rules and regulations and also various different crew district configurations. It is also flexible enough to handle costs which are non-linear functions of arc durations.

2. We formulate the crew scheduling problem 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 crew scheduling problem.

3. We show that the FIFO requirement, if handled by the integer-programming approach, complicates the structure of the problem and makes it computationally intractable.

4. We develop an exact algorithm called *Successive Constraint Generation (SCG)*, which first solves the relaxed version of the integer program (without FIFO constraints) and then iteratively adds constraints in order to eliminate FIFO infeasibilities.

5. We develop an approach based on a *Quadratic Cost Perturbation (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 also show that it produces the optimal solution in most cases.

6. We present extensive computational results and case studies of our algorithms on the real-life data.

The outline for the rest of the paper is as follows. In Section 2, 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. In Section 3, we describe the mathematical modeling approach, which includes construction of the space-time network and the integer programming formulation. In Section 4, we describe the solution approaches developed to efficiently solve the problem. In Section 5, we enumerate some of the practical applications of the model. In Section 6, we present computational results comparing the performance of all our algorithms, and we also present the results of a case study done on a representative scenario. Finally, in Section 7, we make concluding remarks.

2. Problem Description

In this section, we give an overview of crew scheduling problems faced by North American railroads. We proceed by first describing some of the essential terminology needed to understand the problem. We then give an overview of some of the typical regulations which govern crew management. Next, we list the set of inputs required to properly define and formulate the crew scheduling problem, and finally we give a brief description of the nature of constraints and the objective function.

2.1 Terminology

Crew District: The rail network of a railroad 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. Contrary to the airline industry, where certain crews have the flexibility to operate over a large territorial domain, in the North American railroad industry, crews are qualified to operate only in certain specific geographic territories. The physics of operating a train depend 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 to safely operate a train on that route. Consequently, most crews are qualified to operate on a limited number of routes.

Crew Pools: Within a crew district, there are several types of crews called *crew pools* or *crew types*, 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 pre-specified time window. Similarly, a crew pool consisting of senior crew personnel is assigned only to pre-designated 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 crew scheduling problems 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 does not incur any 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, typically, a crew 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 hours, the crew is ready to head back to its home terminal. However, if there is no train, then the crew may have to wait in a hotel. According to the trade-union rules, once a crew is at the away terminal for more than a pre-specified 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: *Crew deadheading* refers to the repositioning of crew between terminals. A crew normally operates a train from its home terminal to an away terminal, rests for a designated time, and then operates another train back to its home terminal. Sometimes, at the away terminal, there is 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 a passenger) 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 as the crew is considered to be on-duty while deadheading and earns wages, and railroads also incur taxi expenses. Each year, a major freight railroad may spend tens of millions of dollars in 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, and hence crews are called on-duty before train departure time. The time at which the crew has to 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 hence 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 time (or *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, 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 getting assigned to a train, the rest time and the duration of the second train may also included in the duty period of the crew. (Section 2.2 describes calculation of duty period in more detail.)

Dead Crews: By federal law, a train crew can only be on duty for a maximum of 12 consecutive hours, at which time the crew must cease all work and it becomes *dead or dog-lawed*. Dead crews are a frequent consequence of delayed trains, congestion, mechanical breakdowns, etc. In these 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.

2.2 Regulatory and Contractual Requirements

Assignment of crews to trains is governed by a variety of Federal Railway Administration (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:

- Duty-period of a crew cannot exceed 12 hours. 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 (10 hours rest followed by 2 hours call period) if duty-period is greater than 10 hours, and after 10 hours (8 hours rest followed by 2 hours call period) if duty-period is less than or equal to 10 hours.

- Whenever a crew is released from duty at the away terminal, they must go for a minimum 8 hours rest, except for these circumstances:

    (i) If the total time period corresponding to the last travel time from the home terminal followed by a rest time of less than 4 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);

    (ii) 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 in between the assignments is more than 4 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 only be operated by crews belonging to pre-specified 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 needs to be followed by crew planners. As 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 been deployed yet at railroads. Another reason for why there has been limited OR 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 to whom solutions can be sold. Also, due to 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.

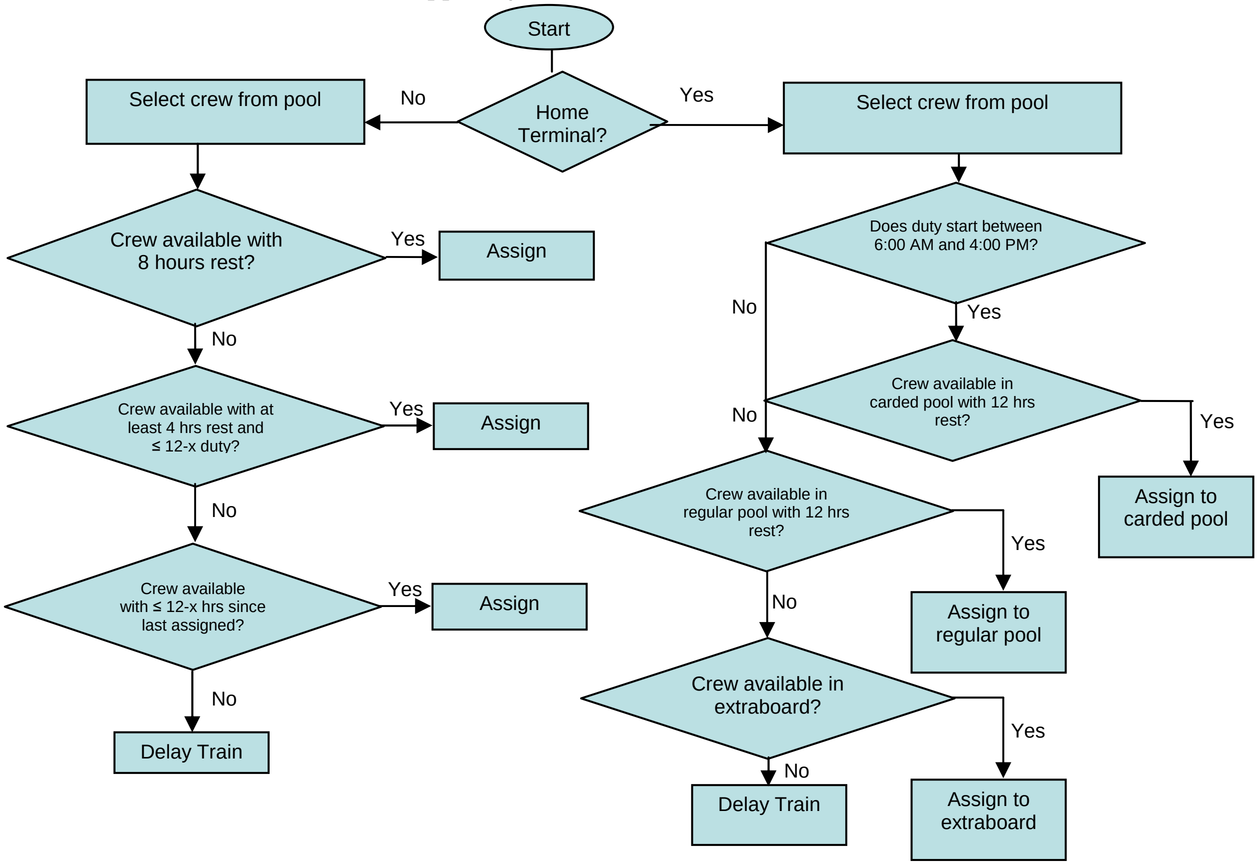


Figure 1. An example of crew assignment decision tree.

2.3 Problem Inputs

Here we describe the inputs that go into the mathematical formulation of the crew scheduling problem.

- **Train Schedule:** The train 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 it passes through. We do not consider stochasticity in the train schedule and assume that train delays are only due to the unavailability of crew and not due to train cancellations or other disruptions.

- **Crew Pool Attributes:** This includes attributes of various crew types, namely their home locations, their away locations, minimum rest time, train preferences, etc.

- **Crew Initial Position**: This provides the position of crew at the beginning of the planning horizon. This includes information of 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 the crew belongs to.

- **Train-Pool Preferences:** The train-pool preferences, if any, give us information about the set of trains that can be operated by a crew pool.

- **Away Terminal Attributes:** This gives us information about the away terminals for each crew pool. It also includes the rest rules and detention rules for each crew pool and at each away terminal.

- **Deadhead Attributes:** This gives us the time taken to travel by taxi between two terminals in a crew district.

- **Cost parameters:** Cost parameters are used to set up the objective function for the crew scheduling problem. They consist of crew wage per hour, deadhead cost per hour, detention cost per hour, and train delay cost per hour.

2.4 Constraints and Objective Function

The crew scheduling problem involves making decisions regarding the assignment of crews to trains, deadheading of crews by 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 while a crew is not assigned to more than one train at the same time. These also include assignment of certain crew pools to pre-specified trains. Assignment of crews to trains must, in addition, satisfy the contractual requirements described in Section 2.2. In our mathematical model, the operational constraints of the model are handled by the integer multi-commodity flow formulation described in Section 3.2, and the contractual restrictions are honored in the network construction phase described in Section 3.1. The objective function of the crew scheduling problem is to minimize the total cost of crew wages, the cost of deadheading, the cost of crew detentions, and the cost of train delays.

3. Mathematical Modeling

In this section, we present our mathematical modeling approach to solve the *crew scheduling problem (CSP)*. We first describe the construction of the space-time network, which is central to all our solution methodologies. In the second part of this section, we formulate the crew scheduling problem as an integer multi-commodity flow problem on this network, establish correspondence between the mathematical formulation and the crew scheduling problem and also discuss the size of the problem and inherent computational complexities.

3.1 Space-Time Network

The crew scheduling problem (CSP) is formulated as an integer multi-commodity 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 constructing 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 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. Note that for the sake of clarity, this network only represents a subset of all 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 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 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 (say l) passing through a crew district, we create a *departure node* (say l') at the first departing station of the train in the crew district and an arrival node (say 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, we create a *train arc* (l', l'') for each train l connecting the departure node and arrival node of train l. 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 the deadhead arcs which satisfy the contractual rules and regulations are created. We construct *rest arcs* to model 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 to the contractual rules and regulations. All rest arcs which 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 a supply node corresponds to crew belonging to crew pool A, then all the arcs which emanate from this node can only carry 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.

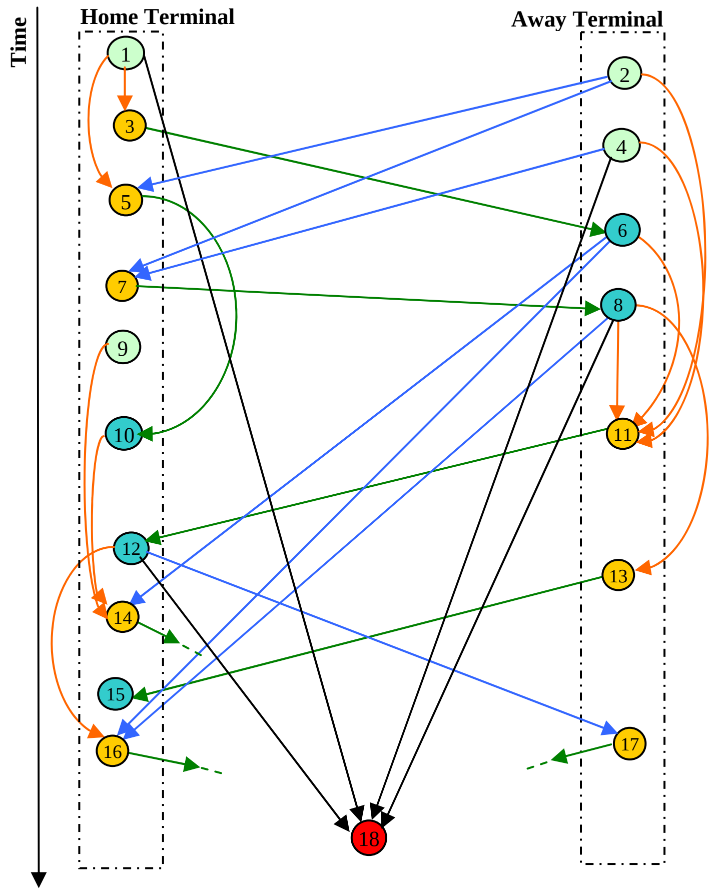


Figure 2. Space-time network for a single-ended district with a single crew type.

Node legend: green (supply), blue (arrival), yellow (departure), red (demand)
Arc legend: green (train), orange (rest), blue (deadhead), black (demand)

The space-time network described above models the flow of crews while honoring all the contractual constraints except the FIFO rule. However, it does not model the case when qualified crews are not available for assignment to a train and hence causing train delays. Next, we present the construction of additional arcs incorporating train delays. At a location, we create rest arcs and deadhead arcs which do not honor the rest regulations and 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 the train will be delayed until crew becomes qualified for train operation. However, as the delay of a train may have propagating effect in 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 make it qualified 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 crew scheduling problem. It can be noted that honoring contractual regulations while constructing the network reduces the number of constraints in the integer program significantly. Now, we present the multi-commodity integer programming formulation of the crew scheduling problem.

3.2 Integer Programming Formulation (IPF)

We formulate the crew scheduling problem as an integer multi-commodity flow problem on the space-time network described in Section 3.1. In our formulation, each crew pool represents a commodity. Crew enters the system at crew supply nodes, and hence 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. Next we present the integer programming formulation of the problem.

Notation:

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 $r \in R$
$tail(l)$:	The node from which arc l originates
$head(l)$:	The node at which arc l terminates

Decision variables:

x_l^c :	Flow of crew pool $c \in C$ on each train arc $l \in L$
x_d :	Flow on deadhead arc $d \in D$

x_r : Flow on rest arc $r \in R$

Objective function:

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

Constraints:

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

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

$$\sum_{a \in N_d^-} x_a = f \tag{3}$$

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

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

$$\sum_{r' \in A_r} x_{r'} - M(1 - x_r) \le 0, \text{ for all } r \in R \tag{6}$$

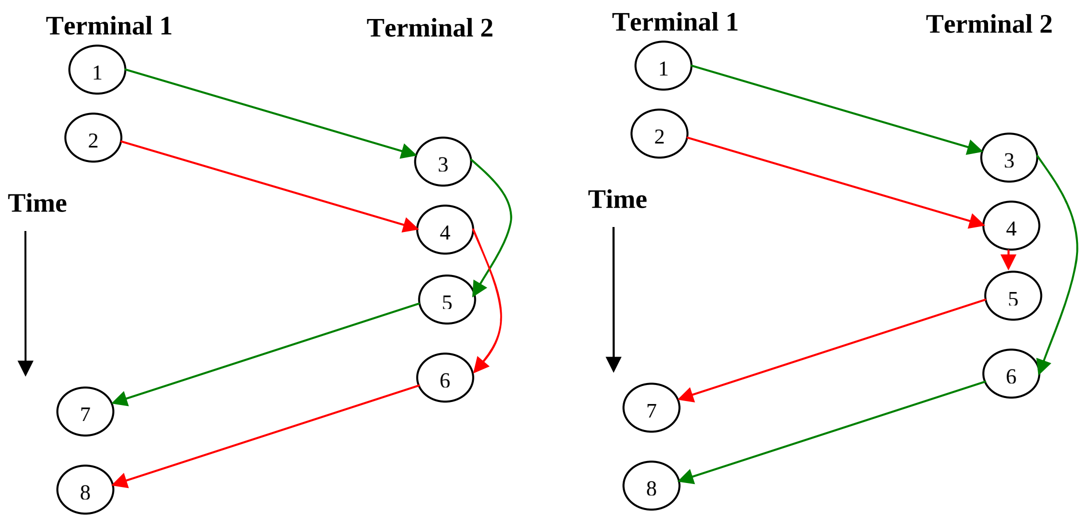
$$x_l^c \in \{0,1\} \text{ and integer, for all } l \in L, c \in C \tag{7}$$

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

$$x_r \in \{0,1\} \text{ and integer, for all } r \in R \tag{9}$$

Constraint (1) is the train cover constraint, which ensures that every train is assigned a qualified crew to operate it. Constraint (2) ensures flow balance at a crew supply node. Constraint (3) ensures the flow balance at the sink node. Constraints (4) and (5), respectively, ensure flow balance at train departure and arrival nodes. The flow balance constraints at a train arrival node ensure that the crew which is assigned to a train is subsequently assigned to a rest arc, a deadhead arc, or a sink arc which emanates from the arrival node of the train. Flow balance constraints at a train departure node ensure that the crew which is assigned to the train has been assigned to a rest arc, a deadhead arc, or a supply arc which 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 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 train 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 train 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).



(a) Valid Assignment **(b) Invalid Assignment**

Figure 3. Illustration of the FIFO rule.

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 $\sum_{r' \in A_{(4,5)}} x_{r'} \le 0 \Rightarrow x_{r'} = 0 \ \forall r' \in A_{(4,5)}$. This ensures that if there is flow on rest arc (4, 5), then 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 $\sum_{r' \in A_{(4,5)}} x_{r'} \le M \ \ \forall r' \in A_{(4,5)}$ which essentially means that the constraint is relaxed.

Let us now estimate the size of a typical instance of the crew scheduling problem in a crew district. Most crew districts have two terminals, and a typical train schedule has around 500 trains running in a couple of weeks in a crew district. Each crew district could have two to four crew types and around 50 crews. Therefore, the space-time network could have around 50 + 2 x 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 based on this rule, the number of deadhead arcs is typically around 25,000, and the number of rest arcs is around 100,000.

Since the number of rest arcs for a typical problem is of the order of 100,000, and as each rest arc has one FIFO constraint, the number of FIFO constraints in the model would be 100,000, which is too large. We would therefore be losing one of the main advantages of the network flow formulation, which is, 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 Integer Programming Formulation where we relax the FIFO constraints (6); we call this problem the *Relaxed Problem*. This problem typically has more than 100,000 variables and several thousand constraints, which make 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 commercial branch-and-bound based MIP solver provided by CPLEX 9.0. We believe that this is due to the special structure of the Relaxed Problem, which helps speed up 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 to the case where variables are integer variables. Whenever the engine branches on a non-integer variable, the value on one branch is set to zero and on the other branch is set to one. Hence, at each level of the tree, one variable's value is prefixed and can be eliminated from the model. Consequently, it is very likely that a feasible integral solution is obtained early on, 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 from the final solution of the model, we can easily extract the set of trains a crew takes over the entire planning horizon. In order 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 due to the commodity specific flow balance constraints at each node, every crew will have 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 crew scheduling problem.*

Proof: Consider a feasible flow on the space-time network. We have seen above how the path of each crew can be extracted from the solution using a simple run-through procedure. Due to the network construction methodology, the extracted path of each crew has to satisfy all the 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, hence establishing the result. ♦

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

4. Solution Approaches

In this section, we present our approaches to solve the crew scheduling problem. As the FIFO constraints are the ones which complicate the nature of the integer programming formulation, 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 integer programming formulation, its application in a real-time environment may be restricted due to 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.

4.1 Successive Constraint Generation (SCG) Algorithm

The *SCG* algorithm works by iteratively pruning out crew assignments which violate the FIFO constraints from the current solution of a more relaxed problem. We considered the following two methods for implementing constraint generation: (1) A branch-and-bound algorithm where constraints are added to the 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 where we run branch-and-bound algorithm on the relaxed problem, solve it to optimality, and then add constraints to remove infeasibilities which is followed by another run of branch-and-bound on the more constrained problem and so on.

However, on further deliberation, we chose to implement the second method over the first for the following reasons:

1. Since 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 more than the number of rest arcs with positive flow in an integral version. Also, larger the number of rest arcs with positive flow, greater is the possibility of FIFO violations. Hence, more FIFO constraints are likely to be added to a non-integral solution.

2. *SCG* implemented using method (2) 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 will not be an integral solution in general. Hence, we do not have the option to prematurely terminate until a point when we obtain at least one integral solution to the LP relaxation and consequently we do not have control over the quality of intermediate solutions in terms of number of FIFO violations.

3. We show in our computational results that Quadratic Cost Perturbation (QCP) described in Section 2.3 does an excellent job in 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 be when it is used along with *QCP*. In this approach, we would first apply *QCP* to obtain a solution with very few FIFO violations. *SCG* is then 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 since when a small number of constraints are added, the problem can be re-optimized 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 which have positive flow, and for each such rest arc assignment which violates FIFO constraints, it adds the corresponding FIFO constraints. We then re-solve the problem and re-check for FIFO infeasibilities. This process is repeated until all FIFO infeasibilities are removed.

Algorithm-SCG

Step 1: Solve the Relaxed Problem. If a feasible solution exists, then proceed to Step 2. Otherwise STOP as the problem is infeasible.

Step 2: Examine all the rest arcs with positive flow in the solution of Step 1. Add FIFO constraints to the integer program on those rest arcs assignments which violate FIFO requirements.

Step 3: If FIFO constraints are added in Step 2, re-optimize the modified integer program and go to Step 2. Otherwise STOP as we have the optimal solution.

Note that the final solution of *SCG* satisfies all the constraints of the Integer Programming Formulation (IPF), and the constraints of *SCG* are a subset of the constraints of *IPF*. Hence, the *SCG* algorithm is an exact algorithm guaranteeing optimal solution to the original problem. However, in the worst case, *SCG* could add all the FIFO constraints to the integer program and would hence become an intractable approach. Fortunately this seldom happens in practice. Our computational results show that the number of constraints added is usually much less 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 could be quite high. In our computational experiments, in some instances, *SCG* had a running time in the order of minutes while in others it had a running time in the order of hours. While these 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 which produces very good quality FIFO-compliant solutions with running times comparable to that of the Relaxed Problem.

4.2 Quadratic Cost-Perturbation (QCP) Algorithm

In the previous section, we describe a successive constraint-generation based approach to remove the FIFO violations iteratively. In this section, we present an algorithm which penalizes the FIFO violations in a solution. We show that this method guarantees zero FIFO violations in the case where there is no priority in assigning crews to trains and serves as a heuristic method for the other case when 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.

We present our cost perturbation strategy through the illustration shown in Figure 4 for the case when there is only one crew pool type. In case (a), crew assignments are made in a non-FIFO manner, and in case (b), the assignments are made in a FIFO manner. Now consider the case when crews are detained at the Terminal 2. Then, due to the nature of detention costs, the cost of the 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 all the rest arcs had a cost of zero; then both the 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) and not assignments of type (a).

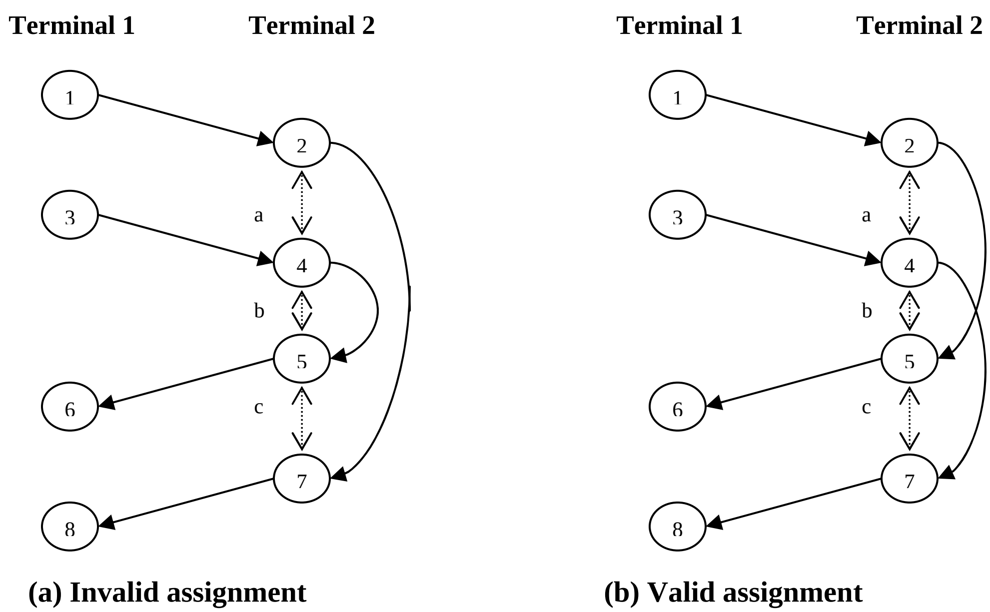


Figure 4. Illustrating the FIFO assignments.

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 2 and 4, 4 and 5, and 5 and 7 are *a*, *b*, and *c*, respectively. Consider a cost assignment which is proportional to the square of the duration of rest arcs. The constant of proportionality is represented by *k*.
Then,

Cost of assignment (a) = $k(duration\ of\ \ arc\ (2,7))^2 + k(duration\ of\ \ arc\ (4,5))^2$

$= k(a+b+c)^2 + kb^2 = k(a^2 + 2b^2 + c^2 + 2ab + 2bc + 2ca)$

Cost of assignment (b) = $k(duration\ of\ \ arc\ (2,5))^2 + k(duration\ of\ \ arc\ (4,7))^2$

$= k(a+b)^2 + k(b+c)^2 = k(a^2 + 2b^2 + c^2 + 2ab + 2bc)$

It can be observed that the cost of assignments in case (b) is less than that in case (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 will be only one crew pool type. The observation made here can also be generalized for multiple crew pools unless there is priority of crew pools in assignments to trains. If there is a priority assigned to crews in assignments to trains, then a crew can have FIFO-violated assignment to gain the priority assignments. We state our observations here as the following theorem.

Theorem 2. *Quadratic Cost Perturbation 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: (a) zero costs, (b) detention costs, or (c) train delay costs. If, for example, all the rest arcs in Figure 4 had zero costs, then as shown above, the Relaxed Problem will choose the FIFO compliant assignment because it is cheaper. If the rest arcs in Figure 4(a) had detention costs on them, then the FIFO assignment shown in Figure 4(b) will either have the same level of detention or lesser detention. Hence, the perturbation scheme will work in this case too. A similar argument would also work for train delay costs because FIFO assignments will always have equal or lesser train delays than non-FIFO assignments. ♦

Since we do not want to change the cost structure of the original problem by 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 which is computed as described above. Our computational tests in Section 6.1 show that this method works very well, and the solutions produced by Quadratic Cost Perturbation are indeed FIFO-compliant in the case where 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 where there are priorities, this approach could be used to obtain a solution with a small number of violations and then the Successive Constraint Generation 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 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 excellent quality of solutions using much less computational time. Due to its attractive running times and high solution quality, this method has the potential to be used in both the planning and the real-time environment.

5. Significances and Uses of the Model

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

5.1 Tactical Crew Scheduling

The defining problem in ***tactical crew scheduling*** is determining which crew should be assigned to operate each train. However, there are a number of sub-problems and issues that must be considered before assigning crews to trains. Railroads have around-the-clock crew calling centers with the responsibilities of 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 200-300 clerks (crew callers) to 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: Is this train covered by a designated assigned pool, or is it to be covered by First-in First-out (FIFO) assignment from the general pool? When is the next qualified crew 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.

- **Recommend which crews to place in hotels and which crews to deadhead home:** When a crew arrives at an away terminal, the crew callers have to 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 trade-off between different cost categories of crew wages, deadheads, detention costs, and rest violation costs.

- **Minimize trains delayed due to shortage of crew:** Train delays are potentially very costly because the delay of a train 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 bring back disrupted operations to normalcy. Suppose 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 done. Using this information and the information about the future train schedule, the crew scheduling models can be used to optimally re-assign crew to trains.

5.2 Crew Planning

The essence of the ***crew planning problem for operations or planning*** is to determine how many crews should be in each crew pool. It can be noted that as 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 based on 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 next.

- **Develop and evaluate crew schedules:** The crew scheduling model can also be used to compare the current crew schedule used 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, average deadhead time, etc. 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.

- **Size of crew pools:** Using the crew scheduling model, we can study the impact of varying the crew pool size on the solution quality. 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's producing a crew schedule which uses the minimum number of crew.

5.3 Crew Strategic Analysis

Strategic management involves development of policies and plans and allocating resources so as to implement these plans. The timeframe of strategic management extends over several months or even years. ***Strategic crew problems*** include forecasting future head-count 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 since it takes 18 to 24 months to hire, train, and qualify train crew personnel. Recently, corporate strategists have been struggling with the trade-off between crew costs and train delays. Our model can be used to quickly calibrate efficient frontiers for each crew district and show what number of crews minimizes the sum of train delay costs and crew costs. If railroad management is dissatisfied with that level of train performance, one can simply increase the cost of train delay, and the model will request additional crews such that a new cost-minimizing solution is obtained.

Some of the applications of the network flow model in the strategic environment are listed below.

- **Determining the number of crew districts and territory of crew districts:** We can use the crew scheduling model to re-optimize and test different crew district configurations. For example, suppose crew district 1 operates trains between location A and location B, and crew district 2 operates trains between location B and location C. Merging all three stations into a single crew district could give us better opportunity to optimize costs.

- **Effect of changing crew trade-union rules:** The crew scheduling problem is a complex optimization problem due to strict trade-union rules related to crew operation. The change of any of these rules will face a lot of resistance from the labor union. At the same time, change of any of these rules has the potential to impact 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 12 hours to 10 hours. We can run the model with the parameter setting of 10 hours and measure the change in crew cost.

- **Forecasting crew requirement:** Based on the forecasted train schedule, we can use the model to help us forecast crew requirement. We first run the model assuming that a very large number of crew is available. Since the crew supply is much more than what is required, many crews will directly flow 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-life 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 in order to achieve their long-term organizational goals.

6. Computational Results

In this section, we present computational results of our algorithms on several problem instances, and we also present a case study done on a representative instance. We implemented our algorithms in Visual Basic .NET programming language and tested them on the data provided by a major Class I railroad. We modeled our integer programs using Concert Technology 2.0 modeling language and solved them using the CPLEX 9.0 solver. We conducted all computational tests on a Pentium IV, 512 MB RAM and 2.4 GHz processor desktop computer.

6.1 Comparison of Algorithms

In this section, we compare the performances of the Relaxed Problem, the exact Integer Programming Formulation (IPF), the Successive Constraint Generation (SCG) algorithm, and the Quadratic Cost Perturbation (QCP) algorithm in several real-life 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 one, making the problem a single-commodity flow problem. In the other set of instances, the number of crew pools is two, and the problem is formulated as a multi-commodity 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 final solution. It can be noted that no FIFO constraints are added while solving the Relaxed Problem and the Quadratic Cost Perturbation (QCP). The results of our computational tests are presented in Table 1.

We have reached the following conclusions from the results:

- The solutions to the Relaxed Problem have the highest number of FIFO violations. However, the solution times are also the fastest.

- The Integer Programming Formulation (IPF) has several thousand FIFO constraints. These constraints make the problem computationally intractable, and we could not obtain feasible solution for any of the instances in one hour of computational time.

- The Successive Constraint Generation (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, and for all other instances, we terminated the algorithm when the iteration that was running at the 30^{th} minute of computational time was complete.

- The Quadratic Cost Perturbation (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. Since the Relaxed Problem provides a lower bound to the optimal solution, QCP algorithm produces the optimal solution in six instances out of eight, and the optimality gap is less than 0.2% 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 terms of both solution quality and solution time. It produces optimal or near-optimal solutions in a few minutes of running time, and it therefore has the potential to be used in both the planning and real-time environments.

6.2 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 drive decision making at a railroad. We perform the case study on a representative two-week data set which has 326 trains, 2 crew pools, and 48 crews, and we run the computational tests using the Quadratic Cost Perturbation (QCP) algorithm. The various aspects of the problem that we observe in this case study are as follows: (i) effect of varying the number of available crews, (ii) effect of varying deadhead cost, (iii) effect of varying minimum rest time at the home location, (iv) effect of varying detention time, and (v) effect of varying detention cost.

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 2 presents the computational results, and Figure 5 plots a chart between the number of available crews and solution cost.

We can make the following observations from this study:

- As the number of available crews decreases, the model tries to compensate for the lack of crews by increasing the level of deadheading and train delays.

- Initially, reducing the number of available crews does not have an 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 (38% reduction) has an insignificant impact on the solution cost, but reducing the number of crews from 24 to 22 leads to the solution cost jumping up by more than $59,000 (20% increase).

The reader should note that the objective function in this case study is not a function of the number of crews used and is only a function of total deadhead, detention and delay. This explains why solutions using different number of crews have identical cost provided their total deadhead, detention and delay are the same.

# of weeks	# of crew pools	Relaxed Problem			2. Exact Integer Programming Formulation				3. Successive Constraint Generation				4. Quadratic Cost Perturbation		
		Cost ($)	Time (Sec)	Number of FIFO constraints Violated	Cost ($)	Time (Sec)	Number of FIFO constraints		Cost ($)	Time (Sec)	Number of FIFO constraints		Cost ($)	Time (Sec)	Number of 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

Table 1. Comparison of algorithmic performance.

Num. of crew available	Num. of crew used	Num. of deadheads	Detention hours	Train Delay hours	Solution cost ($)	Increase in cost
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	-

Table 2. Effect of varying crew pool sizes.

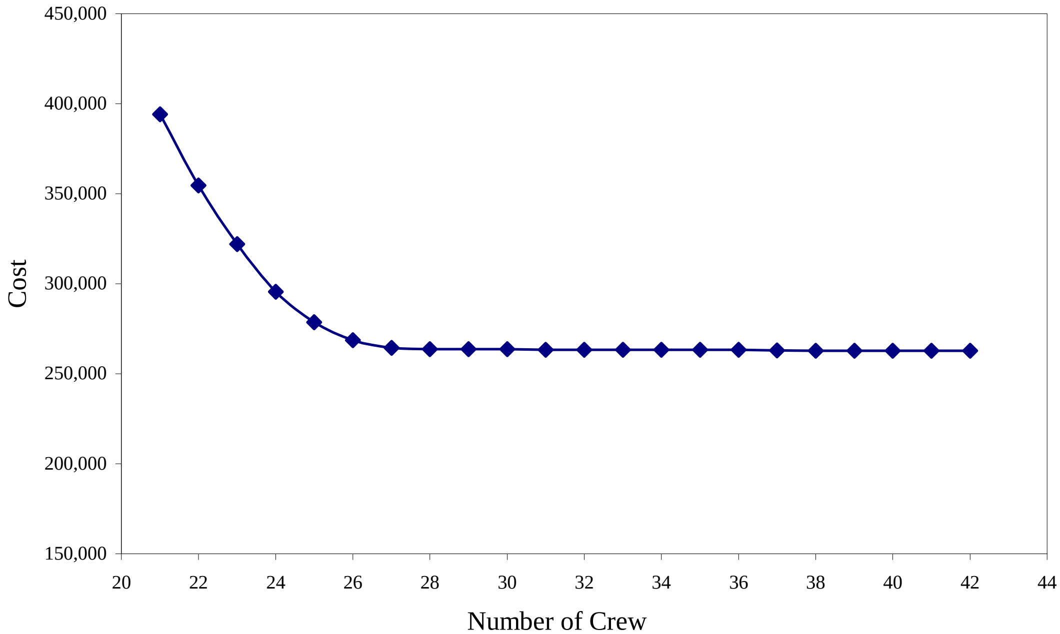


Figure 5. Solution cost versus the number of crew.

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 3.

Deadhead Cost/hr ($/hr)	Number of deadheads	Detention Hours	Train delay hours	Solution cost ($)
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

Table 3. Effect of varying deadhead cost.

We can make the following observations from this study:

- 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.

- As the deadhead cost increases, the solution of the model has fewer deadheads and more train delays. This behavior of the model gives us the insight that if the deadhead cost increases at some point in time, then the railroad needs to adapt by allowing far more flexibility in terms of the train delays. Alternatively, the management can also negotiate with crew unions and reduce the minimum rest hour 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 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 value of this parameter while measuring the impact on the solution as shown in Table 4 and Figure 6.

Minimum rest (hrs)	Average rest at home (hrs)	Train Delays at home (hrs)	Average rest at away (hrs)	Train Delays at away (hrs)	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 4. Effect of varying minimum rest time at the home location.

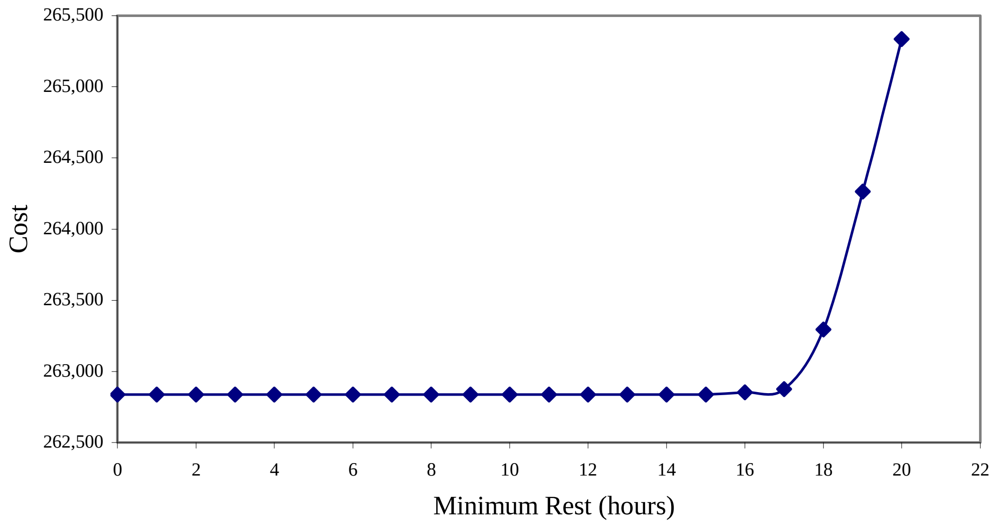


Figure 6. Solution cost versus minimum rest at home.

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, then 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 5. The default value of detention cost used by the railroad is $140 per hour.

Detention cost/hr ($/hr)	Detention hours	Number of deadheads	Train delay hours	Solution cost ($)
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

Table 5. Effect of varying detention cost.

We make the following observations from this study:

- As the detention cost per hour increases, the number of detention hours in the solution decreases.

- As the detention cost per hour increases, the solution has a greater number of deadheads and train delays. This behavior of the model gives us the insight that if the detention cost increases at some point in time, then the railroad needs to adapt by allowing more flexibility in terms of train delays and crew deadheading.

Effect of varying detention time on the solution

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 detention allowance) on the average rest time at the away location, detention hours, and overall solution cost. The default value of detention time provided by the railroad is 16 hours.

Detention time (hours)	Avg. rest at away loc.	Detention hours	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

Table 6. Effect of varying the detention time.

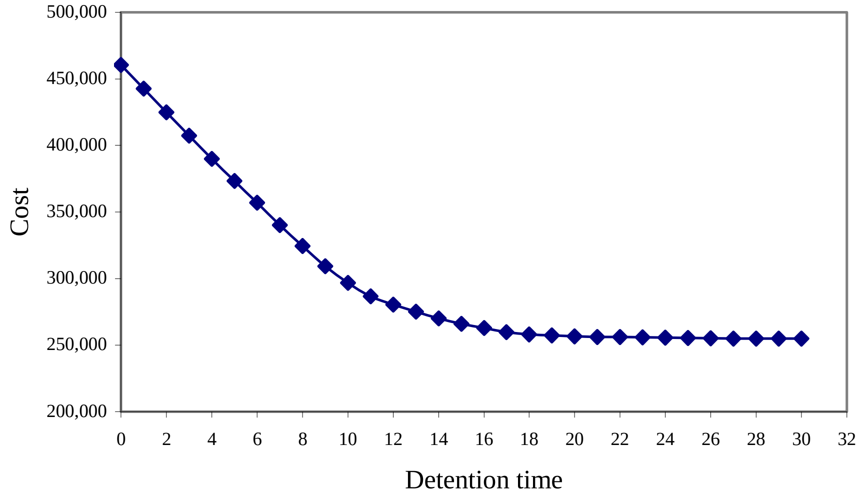


Figure 7. Solution cost versus detention time.

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.

7. Summary and Conclusions

In this paper, we describe a network flow-based approach to solve the railroad crew scheduling problem in the context of North American railroads. The crew scheduling problem for North American railroads is governed by several Federal Railway Administration (FRA) regulations and trade-union work rules. In order to develop a good crew schedule, in addition to satisfying these regulations, we also need to minimize the total wage costs, train delay costs, deadhead costs, and detention costs. 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 could have a different set of operating rules. These factors make this a complex problem to model and solve.

The network flow formulation for the crew scheduling problem developed in this paper is both flexible and robust and can be easily manipulated to represent each of the possibilities encountered in real-life. We formulate the crew scheduling problem 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 integer programming formulation. We develop two approaches to handle FIFO constraints. The first approach is a Successive Constraint Generation approach where constraints are generated iteratively to cut out FIFO violations. The second approach, which is called Quadratic Cost Perturbation, relies on perturbing the objective function to generate FIFO-compliant solutions. We provide extensive computational results comparing the performance of various approaches and show that the perturbation approach outperforms the other approaches both in terms of 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 based on future demand patterns. The model gives railroad executives 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, in doing so, make a significant impact on the railroad’s on-time performance, crew utilization, and productivity, while also improving the quality-of-life for crew and improving railroad safety.

Acknowledgements

We would like to thank Larry A. Shughart from Innovative Scheduling, who has several years of railroad experience, for helping us understand the real-life significance of crew scheduling and for providing us with valuable business insights.

References

Ahuja, R.K., T.L. Magnanti, and J.B. Orlin. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, NJ.

Assad, A.A. 1981. Analytical models in rail transportation: an annotated bibliography. *INFOR* **19**, 59-80.

Assad, A.A. 1983. Analysis of rail classification policies. *INFOR* **21**, 293-314.

Barnhart, C., E. L. Johnson, G. L Nemhauser, and P. H. Vance. 2003. Crew scheduling. In *Handbook of Transportation Science.* Edited by Randolph W. Hall. Norwell, MA: Kluwer Academic Publisher, 493-521.

Barnhart, C., E. L. Johnson, R. Anbil, and L. Hatay. 1994. A column generation technique for the long-haul crew assignment problem. In *Optimization in Industry: Volume II.* Edited by T. Ciriano, and R. Leachman. England: John Wiley and Son, 7-22.

Bodin, L., B. Golden, A.A. Assad, and M. Ball. 1983. Routing and scheduling of vehicles and crews: The state of the art. *Computer and Operations Research* **10**, 63-211.

Caprara, A., M. Fischetti, P. Toth, D. Vigo, and P.L. Guida. 1997. Algorithms for railway crew management. *Mathematical Programming* **79**, 124-141.

Carraresi, P., and G. Gallo. 1984. Network models for vehicle and crew scheduling. *European Journal of Operational Research* **16**, 139-151.

Chu, C.K., and C.H. Chan. 1998. Crew scheduling of light rail transit in Hong Kong: From modeling to implementation. *Computer and Operations Research* **25**, 887-894.

Cordeau, J.-F., P. Toth, and D. Vigo. 1998. A survey of optimization models for train routing and scheduling. *Transportation Science* **32**, 988-1005.

Desrosiers, J., Y. Dumas, M.M. Solomon and F. Soumis. 1995. Time-constrained routing and scheduling. Edited by M.O. Ball et al., *Handbooks in OR & MS*, Vol. 8, Elsevier Science, 35-139.

Ernst, A.T., H. Jiang, M. Krishnamoorthy, H. Nott, and D. Sier. 2001. An integrated optimization model for train crew management. *Annals of Operations Research* **108**, 211-224.

Florida Department of Transportation Report. 2005. Freight rail component of the Florida Rail Plan. Technical Report.

Freling, R., R.M. Lentink, A.P.M. Wagelmans. 2004. A decision support system for crew planning in passenger transportation using a flexible branch-and-price algorithm. *Annals of Operations Research* **127**, 203-222.

Gopalakrishnan, B., and E. Johnson. 2005. Airline crew scheduling: State-of-the-art. *Annals of Operations Research* **140**, 305-337.

Gorman, M.F., and M. Sarrafzadeh. 2000. An application of dynamic programming to crew balancing at Burlington Northern Santa Fe Railway. Int. J. Services Technology and Management **1**, 174-187.

Walker, C.G., J. N. Snowdon, and D. M. Ryan. 2005. Simultaneous disruption recovery of train timetable and crew roster in real time. *Computer and Operations Research* **32**, 2077-2094.

Wise, T. H. 1995. *Column Generation and Polyhedral Combinatorics for Airline Crew Scheduling.* Ph.D. Dissertation, Cornell University, Ithaca, NY.

Wren, A. (ed.). 1981. *Computer Scheduling of Public Transport*. North Holland.