Tactical Network Flow and Discrete Optimization Models and Algorithms for the Empty Railcar Transportation Problem

Arief B. Suharko

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Industrial and Systems Engineering

Hanif D. Sherali, Chair
Sheldon H. Jacobson
John E. Kobza
Scott F. Midkiff
Antonio A. Trani

December 5, 1997
Blacksburg, Virginia

Tactical Network Flow and Discrete Optimization Models and Algorithms for the Empty Railcar Transportation Problem

Arief B. Suharko

(ABSTRACT)

Prior to 1980, the practice in multilevel autorack management was to load the railcars at various origin points, ship them to the destination ramps, unload them, and then return each car to the loading point where it originated. Recognizing the inefficiency of such a practice with respect to the fleet size that had to be maintained, and the associated poor utilization due to the excessive empty miles logged, a consolidation of the railcars was initiated and completed by February 1982. Under this pooling program, a central management was established to control the repositioning of three types of empty railcars for eight principal automobile manufacturers. Today, the practice is to consolidate the fleets of all automobile manufacturers for each equipment type, and to solve the distribution problem of repositioning empty multilevel autoracks of each type from points at which they are unloaded to automobile assembly facilities where they need to be reloaded. Each such problem is referred to in the railroad industry as a *repositioning scenario.*

In this dissertation, we present two tactical models to assist in the task of centrally managing the distribution of empty railcars on a day-to-day basis for each repositioning scenario. These models take into account various practical issues such as uncertainties, priorities with respect to time and demand locations, multiple objectives related to minimizing different types of latenesses in delivery, and blocking issues. It is also of great practical interest to the central management team to have the ability to conduct various sensitivity analyses in its operation. Accordingly, the system provides for the capability to investigate various what-if scenarios such as fixing decisions on running a specified block of cars (control orders) along certain routes as dictated by business needs, and handling changes in supplies, demands, priorities, and transit time characteristics. Moreover, the solution methodology provides a flexible decision-making capability by permitting a series of runs based on a sequential decision-fixing process in a real-time operational mode. A turn-around response of about five minutes per scenario (on a Pentium PC or equivalent) is desired in practice.

This dissertation begins by developing several progressive formulations that incorporate many practical considerations in the empty railroad car distribution planning system. We investigate the performance of two principal models in this progression to gain more insights into the implementation aspects of our approach. The first model (**TDSS1**: *Tactical Decision Support System-1*) considers all the identified features of the problem except for blocking, and results in a network formulation of the problem. This model examines various

practical issues such as time and demand location-based priorities as well as uncertainty in data within a multiple objective framework.

In the second model (**TDSS2:** *Tactical Decision Support System-2)*, we add a substantial degree of complexity by addressing blocking considerations. Enforcement of block formation renders the model as a network flow problem with side-constraints and *discrete* side-variables. We show how the resulting mixed-integer-programming formulation can be enhanced via some partial convex hull constructions using the Reformulation-Linearization Technique (RLT). This tightening of the underlying linear programming relaxation is shown to permit the solution of larger problem sizes, and enables the exact solution of certain scenarios having 5,000 - 8,000 arcs. However, in order to accommodate the strict run-time limit requirements imposed in practice for larger scenarios having about 150,000 arcs, various heuristics are developed to solve this problem. In using a combination of proposed strategies, 23 principal heuristics, plus other hybrid variants, are composed for testing.

By examining the performance of various exact and heuristic procedures with respect to speed of operation and the quality of solutions produced on a test-bed of real problems, we prescribe recommendations for a production code to be used in practice. Besides providing a tool to aid in the decision-making process, a principal utility of the developed system is that it provides the opportunity to conduct various *what-if* analyses. The effects of many of the practical considerations that have been incorporated in TDSS2 can be studied via such sensitivity analyses. A special graphical user interface has been implemented that permits railcar distributors to investigate the effects of varying supplies, demands, and routes, retrieving railcars from storage, diverting en-route railcars, and exploring various customer or user-driven fixed dispositions. The user has the flexibility, therefore, to sequentially compose a decision to implement on a daily basis by using business judgment to make suggestions and studying the consequent response prompted by the model. This system is currently in use by the TTX company, Chicago, Illinois, in order to make distribution decisions for the railroad and automobile industries.

The dissertation concludes by presenting a system flowchart for the overall implemented approach, a summary of our research and provides recommendations for future algorithmic enhancements based on Lagrangian relaxation techniques.

Dedication

In the name of God, the most benevolent, ever merciful.

This work is dedicated to Dr. Hanif D. Sherali and my beloved parents.

Acknowledgments

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Motivation

Ever since its inception, the field of operations research has long been used in the transportation industry. Necessitated by keen competition among different transportation modes and an ever increasing level of pressure from customers, the industry has recognized a strong need to develop more sophisticated models to manage its complex operations.

The railroad industry offers a very important mode of transportation in North America. Forced mainly by a set of deregulating laws, this industry has been self-transforming itself from the low cost-low quality of service, toward a highly reliable and flexible player in the field. Within the industry itself, freight railcars represent a substantial portion of total railroad investment, and a higher utilization of this major capital resource is very essential to the survival of this industry in the face of fierce competition.

Before presenting the focus of this dissertation, we first provide a historical perspective and address various issues of practical consideration in railroad management.

1.1 Historical Perspective

In 1980, the Stagger's and Motor Carrier Acts were introduced to deregulate the railroad industry in the U.S. This set of laws has become the major force that has driven the restructuring of the industry. As reported in Keeler's review (1984) of the state of the rail industry prior to deregulation, when the U.S economy was still being dominated by bulk commodity production, low cost-low quality service could maintain high profitability. However, the gradual shift in the economy toward the implementation of more efficient inventory policies (such as just-in-time) meant that the railroad industry could no longer maintain the old practice of merely keeping up with the growth of its clients' demands. As a result, to stay competitive and profitable, the industry has been reinvesting in improved technology along with enhancing its management practices.

Railroad cars represent one of the largest capital investments made by most railroads. Yet, Felton (1978) found that in general, this resource was very poorly managed. Presently, there is a strong initiative in the railroad industry to reverse this trend. One essential contributing factor in achieving this is in the improved management of distributing empty railroad cars.

Mendiratta (1982) defines the empty railroad car distribution problem as the flow control of empty railcars from the time of unloading to the reloading or off-service time. He argues that decentralizing the decision-making process with respect to this empty railcar distribution by assigning this task to local (terminal) managers may lead to a poor utilization of resources due to their inability to view the state of the entire network. Instead, he suggests that many decisions related to the opportunity cost of shipping railcars should be relegated to the central (corporate) level. He asserts that the best strategy may involve a communication between the two levels.

One specific empty railroad car distribution problem involves a set of similar clients, such as automobile manufacturers (**shippers**). In this context, railroad industries have two basic options. They either deal with their clients on an individual basis or pool their resources to provide better services to the shippers.

Prior to 1980, the practice of the railroads was to load the railcars at the various origins, ship them to their destinations, and return them to their origins. In this scheme, empty railcars tended to log more miles while at the same time, these railcars could not be redirected toward other nearby origins to satisfy demand in a more timely fashion. These factors contributed to inefficiencies in the overall railroad operations as the tracks were tied up unnecessarily by empty railcars, bringing in no revenues and requiring the railroads to maintain unduly large fleet sizes.

In 1982, the railroad industry launched a program to consolidate its automobile-carrying business into the management of one common resource pool. There were two types of (multilevel) railcars that were used, the trilevels and the bilevels. Recently, a third type (TTQ-X) has also been added. Under this program, the railroads participating in pooling agreements with each automobile shipper would pool the autorack railcars of each type, and upon unloading, would redirect the empty railcars to economically convenient loading locations, rather than ship them back to the loading point of origin. A central management, the *Association of American Railroads*, was given the authority to control the empty railcar repositioning for mainly eight separate automobile manufacturers' pools of railcars. (These manufacturers are Chrysler, Ford, General Motors, Toyota, Honda, Mitsubishi, Subaru, and Nissan.) Considering the potential benefits of the new scheme, this move was welcomed by the railroads and the automobile manufacturers alike. Most recently, the railroad and automobile industry have taken one additional step, namely, to pool the multilevel autoracks of all manufacturers as well. Hence, the problem has been further consolidated into determining repositioning decisions for each railcar equipment type that is shared by a group of automobile manufacturers, and is distributed by a collection of railroad companies. Each such problem is referred to in the railroad industry as a *repositioning scenario*. This dis-

tribution of pooled resources introduces some challenging issues dealing with priorities and equities, besides increasing the complexity and size of the problem that needs to be solved.

In the next section, we discuss some of the problems that arise in implementing the management of this global pool of empty railcars.

1.2   Some Practical Considerations

In the pooling scheme, the first task faced by each automobile manufacturer is to determine the fleet size they need to acquire for their pool in any given year and to determine the apportionment of responsibility for this capital investment among the participating railroad companies. Fleet sizing itself uses the demand information for a typical month of the year, that frequently tends to be close to the peak period demand. Car shortage aversion tends to motivate the shippers to focus on peak demand, at the expense of overstating fleet requirements. As the fleet size becomes unnecessarily large, it results in an unduly high leasing cost incurred by the railroads, which in turn, reduces the competitiveness of the railroads.

Another practical aspect in the empty railcar shipment process is the consideration given to the issue of late deliveries. It is naturally more acceptable for automobile manufacturers to have late empty railcar arrivals than to have a complete shortage within a given time horizon. Obviously, in general, on-time delivery is the preferred case. However, if this preference results in large fleet sizes and low fleet utilization, then timeliness may be sacrificed to several degrees, such as to permit one-day late deliveries, two-days late deliveries, etc. Some shortages at certain demand points, nevertheless, may occur due to the fluctuations of demands and supplies at various points in the network, given that typically, a delivery that is more than three days late is considered unacceptable.

Regarding the demand locations for empty railcars, some automobile manufacturers may prioritize some of their plants higher than others with respect to receiving timely deliveries, based on dealership and consumer needs. Thus, the implementation of the pooling agreement should be able to incorporate a flexible demand prioritization strategy. Additionally, there has to be a self-adjusting mechanism in a collaborative scheme in which all the shippers' railcars are pooled together, to avoid "gaming" on the part of shippers by way of specifying dominatingly high priorities. Furthermore, in such a common pooling strategy, an important practical issue is to address the concern of each manufacturer that it receives an equitable use of railcars that is based on its assigned car-day measure.

Because of several operational factors, uncertainty issues arise in estimating the demands and supplies of the empty railcars for a given time-period, as well as in the transit times. This consideration requires a robust method that appropriately addresses the stochasticity in the problem, and whose solution tends to be less sensitive to the uncertainties.

Another relevant issue is that of *blocking* that arises due to the nature of the business and

due to economies of scale. At any given origin point on any day, it is far more efficient from the viewpoint of yard operations to compose only a few (one or two) train units that are dispatched to various destinations. These unit-trains are referred to as blocks and the phenomenon of forming blocks is known as *blocking*. Blocking needs to be enforced as it results in a more efficient use of resources. A typical blocking policy specifies a maximum number of blocks that can be formed using the supply of railcars at any originating location at any point in time, as well as some minimal size for each packet of flow or block.

Lastly, the railcar repositioning task is to be carried out on a daily basis. Many intangible business and customer related issues arise that need to be addressed, sometimes in contradiction to efficient practices but more attuned with reality. Hence, it is of practical interest to the central management team to have the ability to conduct various sensitivity analyses in its operation. The ability to investigate scenarios such as fixing decisions on running a specified block of railcars (called control orders) along certain routes as dictated by business needs, changes in supplies, demands, priorities, and transit time characteristics, must be addressed to provide a flexible decision-making capability. In turn, this requirement means that the solution method should permit a series of runs based on a sequential decision fixing process in a real-time operational mode. A turn-around response of about five minutes per scenario (on a Pentium PC or equivalent) is considered acceptable in practice.

Having mentioned the salient issues encountered in a practical implementation of the carpooling agreement, we proceed to provide a statement of the objectives of this study.

1.3   Statement of Scope, Purpose and Contributions

Many operations research models have been designed to address various issues in the railroad industry. As the industry progresses, we need to construct better models to incorporate real world aspects of its operations. This study is primarily concerned with developing a suitable model for repositioning empty railcars subject to supply, demand and transit time information over a multi-period horizon, while addressing various issues as discussed in the foregoing section, including uncertainties, priorities of demand, multiple objectives related to minimizing lateness in deliveries, and blocking issues. Also, suitable real-time solution methods need to be developed and tested using field data, with recommendations to be provided for making production runs in practice.

We begin by developing several progressive formulations that incorporate many practical considerations in the empty railroad car distribution planning system. We investigate the performance of two principal models in this progression to gain more insights into the implementation aspects of our approach. The first model (**TDSS1:** *Tactical Decision Support System-1)* considers all the identified features of the problem except for blocking, and results in a network flow formulation of the problem. The second model (**TDSS2:** *Tactical Decision Support System-2)* incorporates blocking considerations. This renders the formulation

into a mixed-integer 0-1 programming problem. We exploit the special structures of this empty railroad car distribution problem in order to construct several heuristics that deliver an acceptable solution within the real-world imposed turn-around time limit.

The importance of this research is evident as it serves to develop a practical tool for tactical decision making that will be adopted as a standard in the railroad and automobile industries. No model presently exists that addresses all the important relevant issues that have been identified. Moreover, the proposed model that does address these issues is a massively large discrete optimization problem that in itself poses a significant challenge to solve effectively.

1.4   Organization of the Dissertation

This dissertation is organized as follows. Chapter 2 presents a literature review of the relevant railroad management problems. Although our focus is on the empty railcar distribution planning system, issues such as locomotive power scheduling that could directly affect the quality of our solution are also addressed. For example, if a power scheduling system produces a locomotive shortage at an origin having been designated by our model to be a primary supply point, then obviously this will result in an ineffective decision process. Ideally, one could adopt an integrated approach that simultaneously considers all such related problems faced by the industry. However, the inherent complexity of the details of the business renders such a strategy intractable at the present time. Intertwined as they are, nevertheless, each of the problems maintains unique characteristics that are exploitable. Each problem therefore merits a more special study on its own. Thus, Chapter 2 provides a glimpse on the progress other researchers have made on different areas so that one could synthesize the overall problem of concern to the railroad industry.

Chapter 3 begins with the description of TDSS1: Tactical Decision Support System-1 that we have developed to address the empty railcar distribution planning problem. This is a network flow model that serves as a basis for several enhancements. It examines various practical issues such as time and demand location-based priorities as well as uncertainties in data within a multiple objective framework .

Chapter 4 adds a substantial degree of complexity by addressing blocking considerations. Enforcement of block formation renders the model as a network flow problem with side-constraints and *discrete* side-variables. We show how the resulting Mixed-Integer-Programming formulation can be enhanced via some partial convex hull constructions using the Reformulation-Linearization Technique (RLT) of Sherali and Adams (1990, 1994). This leads to the model TDSS2: Tactical Decision Support System-2. However, in order to accommodate the strict run-time limit requirements imposed in practice, various heuristics are developed to solve this problem. In using a combination of proposed strategies, 23 such heuristics, plus other hybrid variants, are composed for testing.

Illustrations of the quality of solutions from the network flow and blocking models are the

focus of Chapter 5. By studying comparisons among these models based on the results using a test-bed of real and synthetic, realistically sized problems, the end-users can gain valuable insights in selecting an approach that best serves their objectives. Recommendations for an initial production code to be used in practice are made based on these test results.

Finally, Chapter 6 presents a system flowchart for the overall implemented approach. We include in this chapter discussions on implementation details, a section of summary and conclusions, along with recommendations for future research. A bibliography of cited references is also provided.

Chapter 2

Literature Review

Problems faced by the freight railroad operations range widely in scope. Some, such as real-time traffic control, could be readily handled by the deployment of new technologies such as the GPS System. Others, such as scheduling problems, require more analytical Operations Research techniques. We limit the scope of our literature review to those of the second type and refer interested readers to a more appropriate source, say in the Electrical Engineering domain, for the first one. (For example, see the survey paper by Harker (1990)).

In this chapter, we first assess the current state-of-the-art in the area of locomotive power management, as motive power plays a vital role in the industry. Next, we discuss several queueing yard models that have been proposed in the literature. This section is followed by the review of some aspects of strategic (long-term) planning and tactical (short-term) planning in this area. Specifically, in *strategic planning*, one is interested in finding an optimal fleet size that must be acquired in order to satisfy anticipated annual demands while fulfilling long-term pursued objectives. On the other hand, *tactical planning* entails a day-to-day detailed management of the distribution of flow in the railroad network. Empty railcar shipment, as opposed to loaded flows, is of principal interest here, since this lends itself to centralized management. We conclude our survey by briefly examining how other transportation industries, such as trucking and airline industries, handle their routing and scheduling problems.

2.1   Locomotive Power Scheduling

Most papers on motive scheduling are concerned with cost minimization over a given time horizon. Earlier work in this area was carried out by Florian et al. (1976) and by Booler (1980). Booler, in his research, investigates the use of a linear programming model to produce an approximate integer solution to the original model. In lieu of focusing on the closeness of the solution to optimality for real-world problems, these early works are content with finding

good solutions to small problems.

Wright (1989) is the next author to present a heuristic procedure for obtaining reasonable solutions to moderately large problems. In his stochastic approach, two types of techniques have been utilized. The first is a *local improvement* method applied successively from randomly chosen starting points, while the second uses the simulated annealing method. The author claims that the second method tends to produce better results.

Smith and Sheffi (1989) construct a model based on a multi-commodity flow problem having a convex objective function and defined on a time-space network. Expected cost minimization under uncertainty is made possible in their model due to the convex objective function. Trip arcs with a high probability of having very little power are suitably penalized. The method dispatches locomotives along shortest paths (based on marginal arc costs) in the time-space network, and then attempts to improve interchanges of locomotives around cycles.

In their paper, Forbes et al. (1991) present an exact algorithm for the solution of this problem based on an approach used to solve multiple-depot bus scheduling problems. The authors assume that a daily repeating timetable and heterogeneous types of locomotives have been given in advance. Accordingly, each timetabled train should be assigned an appropriate level of locomotive power. Cost minimization objectives include the fixed cost of locomotives and the cost of *light running* (the reshuffling cost of moving the motive power from the end to the head of the train). The algorithm produces a cyclic pattern of assignments of locomotives that enables the same timetable to be used from one day to the next.

2.2   Yard Models

The function of a rail-yard is to serve as a place for inspecting arriving trains, classifying railroad cars, further processing (reshuffling) the cars, and making departure preparations for the newly assembled trains. Suzuki (1973) is one of the earliest researchers in the area who attempted to model this operation. He constructs a basic constrained multi-commodity flow problem but leaves out some practical constraints in his model.

Later on, focusing on internal rail-yard operations, Petersen (1977a, b) builds an analytic queueing model of major rail-yard operations. His interesting model, nevertheless, inevitably leads to a suboptimal strategy as individual yard optimization causes severe congestion at several key yards in the network. Bodin et al. (1980) propose a model that extends the scope of the analysis beyond the rail-yard station itself in order to lift Petersen's limitations. They consider an optimization problem concerned with between-yard movements as well as the operations within the yards themselves. The model determines optimal blocking strategies over all yards in the network simultaneously. The resulting large multi-commodity flow problem having a number of side constraints proves to be computationally very challenging, however.

Various aspects of yard models are also surveyed in Dejax and Crainic (1987) and Crainic et al. (1990). We refer the reader to these papers for further details on this issue.

2.3   Fleet Sizing

The literature contains several fleet sizing models for the railroad industry. Assad (1980) presents a survey of this field. Dejax and Crainic (1987), along with Crainic et al. (1990), also provide a general reading on other strategic planning and management issues faced by the industry.

Given the demand for loaded trips (number of trips leaving each node of the network) and the characteristics of each trip (length, uncertainty, etc.), fleet sizing models usually deal with finding an optimal required number of vehicles to satisfy demand with respect to some criterion. Turnquist (1985) classifies vehicle fleet sizing problems into two major categories, deterministic and stochastic. Within each category, he further divides it into one-to-one (single origin to single destination), one-to-many (single origin to a set of destinations), and similarly, many-to-many problems. He also notes the possibility of classifying the problems into full or partially loaded vehicle problems.

Turnquist and Jordan (1986) develop a one-to-many model for sizing a fleet of containers. They use deterministic cycle times for shipping the containers to the network and stochastic travel times within the network. The aim is to develop an equation that relates the fleet size with the probability of a container shortage.

To study the potential fleet-size reduction when two or more carriers cooperate or pool with one another, Muehlke (1984) builds a two-railroad model. Based on the results obtained, he shows that cooperative efforts among railroads are necessary if an individual railroad is to improve the level of service to shippers and reduce costs.

An earlier model by Avi-Itzhak et al. (1967) considers a railroad car pool system that is subject to stochastic demand. The authors use this model to derive the number of cars required in the pool by determining the distribution of the number of busy units for a class of consignees having small orders and several consignees having bulk orders.

Allman (1974) constructs a model that incorporates a railroad car rental pricing structure. He uses linear programming to determine an optimal mix of railroad cars for shipment to maximize profits due to the rental fees to be received. This results in an assignment model for the allocation of cars to orders. In addition, he examines the effects of aggregating individual cars into order types to reduce the size of the model in the solution process.

A paper by Beaujon and Turnquist (1991) examines decisions related to sizing a vehicle fleet and utilizing that fleet. The goal is to optimize both sets of decisions simultaneously under dynamic and uncertain conditions. The authors build a network approximation to the problem to facilitate the proposed solution procedure.

Lastly, Sherali and Tuncbilek (1997) present the most contemporary fleet sizing approach that has presently been adopted as a standard in the automobile and railroad industry. In this paper, the authors develop and integrate two fleet sizing models, a simpler static model and a more complex dynamic model. The former is based on *static* time-independent, typical month of the year data. It does not consider the actual time- varying demand pattern over the year, or the impact of the transit times in the actual rerouting decisions. The latter model is based on a time-space network representation, wherein each origin and destination location on each day of the planning horizon is represented by a distinct node. With the dynamic model, the user can gain useful information related to the storing and retrieving of empty cars. In turn, this information facilitates the formulation of suitable inventory policies through a simple break-even analysis. Furthermore, the authors demonstrate how the simpler static model can be calibrated using the dynamic model in order to produce results that are compatible with the latter model. This enables the static model to be used to conduct various "what-if" scenarios in a real-time mode.

2.4 Empty Railcar Distribution

Feeney (1957) writes one of the earliest paper that recognizes the importance of managing empty flows in the railroad industry. As part of a first attempt, he assumes that the demands and loaded flows of vehicles are given, and that they are both independent of time and are deterministic. In formulating the resulting transportation problem, he represents the arc cost as the distance or cost of moving empty freight cars between pairs of terminals in the rail network. Subsequently, several other authors (Leddon and Wrathall (1968), and researchers at the Association of American Railroads (1976)) employed static linear transportation models for known supply and demand of empty railcars of a homogeneous fleet to be solved by standard simplex codes.

As the use of time-averaged data makes the model unrealistic, several researchers started to use transshipment formulations to include dynamic effects. White and Bomberault (1969) construct such a formulation in their paper. White (1972) extends this model by proposing an inductive out-of-kilter type of algorithm to optimize the flow of a homogeneous commodity through the network, given a linear cost function. Nevertheless, his model still assumes that railcar supply and demand are perfectly known over time. Along similar lines, Mistra (1972) develops a linear programming-based model that suffers from the same limitations.

Philip and Sussman (1977) simulate the dynamics of the inventory level of empty cars at a single terminal. Mendiratta (1981) combines the existing network and inventory approaches into a novel two-level decentralized optimization model, the network model and the terminal model. The objective of the former is to maximize profits over the entire network, subject to the constraints imposed by empty car supply and demand. The terminal model is an inventory-control model that incorporates stochastic demands and lead times for delivery of empty cars. Internal transfer prices that reflect the opportunity costs of cars are used as a

means of communications between the two models.

A more sophisticated model for the distribution of pooled empty freight cars by several participating railroads is constructed by Glickman and Sherali (1980). The authors distinguish two approaches, a system focus and a company focus. The former emphasizes a total cost minimization while the latter emphasizes the benefits of a pooling agreement to the individual railroads. In this latter model, temporal variations in car supply and demand levels are taken into account. The methodology seeks to achieve a level of equity in the relative savings among the railroads involved.

Jordan and Turnquist (1983) develop a multi-period, nonlinear stochastic programming model. Within the model, they are able to incorporate the uncertainties in supplies, demands and transit times for empty cars. However, the resulting formulation turns out to be too complicated to be useful in practice.

In their model, Newton et al. (1997) incorporate blocking constraints. A block, which constitutes the grouping of several shipments of railcars between an Origin- Destination (OD) pair, once created, results in a more efficient use of the resources due to economies of scale. In solving their model, the authors resort to a column generation procedure for solving the LP relaxation embedded in a customized branch-and-bound approach. This model differs with our model to be described in later chapters in several ways. Firstly, our model is multi-period. Secondly, we use a multi-objective function. Moreover, in terms of the blocking constraints, the flow within a block is still transparent. Lastly, our model takes into account the uncertainty issue. None of these features are present in their model.

2.5   Routing and Scheduling in Other Transportation Industries

Several other businesses dealing with container shipment, trucking operations, and airline transportation share some common features with the railroad industry problem under present consideration. We briefly mention a few such connections and refer the reader to certain key papers for further information.

Jarke (1982) reviews the container management problem and recognizes the importance of an efficient information and decision support system for container transportation logistics. Crainic et al. (1993) investigate the empty container allocation problem in the context of the management of land distribution and transportation operations for international maritime shipping companies. The problem involves dispatching empty containers of various types in response to requests by export customers, and repositioning other containers to storage depots or ports in anticipation of future demand.

Deodovic (1988) reviews several fleet sizing models for the airline industry. All these models, however, cannot be applied directly for the railroad industry since in the latter, deadheading

is a necessity as loads do not originate in the immediate vicinity of all ramp (unloading) locations. The railroad operations also involve varying numbers of cars needed for each loaded movement, as opposed to a single airplane making each individual trip in the airline industry.

In their review paper, Ball et al. (1983) describe several fleet sizing models for the trucking industry. It turns out that the railroad industry shares more similarities with the trucking industry than with the airline industry for this class of problems. Nevertheless, trucking models usually permit leasing in times of shortage, which is not common for the railroads. The trucking industry also has the ability to track the cyclical tours of each individual truck. In contrast, individual railcars do not maintain their identity in the process of moving groups of cars between nodes in the transportation network. Moreover, railroad companies typically use annual demand patterns rather than short cycles of demand variations as prevalent in the trucking industry.

Aside from these conceptual commonalties, the actual details of objective functions, the nature of uncertainties, and blocking requirements, make the tactical railroad-automobile industry problem being considered herein quite unique from other transportation and distribution problems. We now proceed to address these details and construct suitable models and solution procedures for this problem.

Chapter 3

Network Flow Model Development (TDSS1)

In transporting their products to dealers, several U.S. automobile manufacturers have been using the service offered by the railroad industry. In turn, to achieve better efficiency, the participating railroad companies have agreed to pool empty railroad cars (multilevel autoracks) used in this distribution process. The task of managing this empty railroad car distribution has been assigned to RELOAD$^{\text{TM}}$, formerly, a branch of the **Association of American Railroads (AAR)**, and now absorbed within the TTX Company. RELOAD$^{\text{TM}}$ uses mathematical modeling and analysis to relocate an industry-wide fleet of 46,500 multilevel autorack railcars, distributing them to some 86 automobile assembly facilities owned by 14 auto manufacturers (of which 8 major ones dominate most of the business). This chapter describes a tactical model that analyzes the day-to-day multilevel railroad fleet management problems faced by the railroad and automobile industries. Each problem deals with the repositioning of a fleet of empty multilevel autoracks of a particular type, that constitutes a union of fleets of a common equipment type that are assigned to groups of automobile manufacturers (or shippers). The model incorporates a look-ahead feature for making informed imminent day decisions by using a suitable arc-cost penalty structure on a multi-period horizon network formulation. It also treats uncertainties in data, a pre-specification of fixed flow-orders on designated routes (*control orders*), and accommodates the ability to enforce time and demand location based priorities. A preprocessor to this model addresses various additional practical issues such as the consideration of variable queue sizes, and the treatment of weekends that correspond to zero demand periods. The particular feature of blocking is postponed for consideration to the following chapter; indeed, this renders the model into a more difficult 0-1 mixed-integer programming problem.

A common problem faced by all modes of freight transportation is the efficient distribution of empty vehicles from locations where they have been unloaded to locations where they will be reloaded. In the case of railroads in the U.S., since the early 80s, several railroad

companies have created a common empty multilevel freight car pool to serve the needs of the automobile manufacturing industry in regard to shipping automobiles via railroad autoracks. In this scheme, upon unloading, the empty cars of each type are treated as part of a common pool, and are redirected to economically convenient loading locations, rather than being shipped back to the same loading point of origin. The task of repositioning empty cars for use in loaded movements is managed centrally by the TTX Company as mentioned above. The management team at TTX handles day-to-day repositioning decisions in an attempt to minimize the total empty transit times and the number of late deliveries and shortages for various "*scenarios*" based on equipment types that are shared by some eight major automobile manufacturers. A separate fleet sizing model (due to Sherali and Tuncbilek, 1997) is used to determine fleets of cars that should be assigned to each automobile manufacturer based on a projected annual time-varying demand pattern. Accordingly, each railroad acquires cars for these fleets in proportion to the extent of loaded business contracts that it has negotiated with the corresponding automobile manufacturer. To enhance the efficiency of the empty car movements, and hence to cut down on both empty transit costs as well as fleet sizes through an increased utilization, groups of these manufacturers' fleets that correspond to the same equipment type are consolidated into a common pool. Everyday, the management team collects information on the number of available and anticipated supplies of empty cars for subsequent days within the planning time horizon, along with current and estimated demands of empty cars at each automobile plant within the given time horizon. This information is then used to suitably match supply with demand.

Prior to this effort, the tactical model being used was the basic transportation model developed by Glickman and Sherali (1985) which assigns supplies of empty cars of different types, at various locations and on various days, to satisfy demands for different car types at various locations on various days, while accounting for the limited substitutability among the different car types along with substitution costs, and also accounting for the estimated transportation times and queue sizes. This chapter reflects subsequent improvements that have been made to the basic model by incorporating look- ahead considerations along with the treatment of weekend demands and variable queue sizes, uncertainties in data, and time-based and demand location-based priorities. (**TDSS1**: *Tactical Decision Support System*-1), the model described in this chapter, is one option in the plan to revamp and enhance the tactical management of empty multilevel autoracks on a day-to-day basis. A second option that also incorporates blocking considerations (**TDSS2**) is discussed subsequently in Chapter 4.

An outline of this chapter is as follows. Section 3.1 discusses the basic structure of the problem, and Section 3.2 introduces the overall time-based multiple objective model TDSS1. Section 3.3 further enhances this model representation by addressing the issue of demand location-based priorities, while Section 3.4 covers the uncertainty issue. Finally, Section 3.5 provides a glossary of notation along with a list of input requirements, and summarizes the various steps involved in constructing the complete representation of TDSS1.

3.1 Basic Time-Space Transportation Structure

Suppose that there are n_{loc} demand locations (automobile plants) and m_{loc} supply locations (automobile unloading points or ramps). Let t_{min} be the imminent period for which active demand satisfaction decisions are under consideration (usually, one or two days from the present time, t_{now}), and let t_{max} denote the maximum time-period used in the "look-ahead" model horizon. The demand nodes for the corresponding transportation problem are obtained by replicating the n_{loc} demand locations for time-periods t_{min} to t_{max}. Also, the supply nodes of the model are obtained by replicating the m_{loc} supply locations for time-periods t_{now} to $(t_{max} + 1)$, noting that late shipments are also considered whenever unavoidable.

For convenience, we will designate $p = (i, t)$ and $q = (j, \tau)$ to be the *supply and demand nodes*, respectively, of this expanded time-space transportation bipartite graph, where (i, t) refers to origin/supply location i at time t, and (j, τ) refers to the destination/demand location j at time τ. Let m and n respectively denote the total number of these supply and demand nodes, and let the corresponding supply and demand of empty railcars at nodes p and q be denoted by s_p and d_q, respectively. The amount of demand takes into consideration a buffer queue size that is maintained at each location, as well as the appearance of weekends in the horizon during which no movement of railcars occurs. These two aspects of the problem are addressed in more detail next.

3.1.1 Variable Queue Size

Prior to our recommendations, the practice in the industry was to maintain a fixed queue size of empty railcars throughout the year at each loading location, where this size was determined by the preparation time required, the uncertainty of supply at that location, and other business related considerations. Since higher queue requirements inflate the resulting fleet size, a modification of this inefficient practice became necessary. In TDSS1, as an improved planning strategy, a dynamic, variable queue size policy is implemented for each loading location. The motivation is to provide a reasonable buffer that reflects the time-varying demand pattern, and hedges against shortages under uncertainties in supply and demand, while avoiding an unnecessary increase in fleet size.

One alternative that can be easily incorporated within TDSS1 is to change the fixed queue size requirements according to seasonal variations in demand. Either monthly variations, or variations that follow the March, June, and October peaks, with valleys in between, can be considered. To avoid sudden changes in demand requirements due to queue size variations, a gradual smoothing in the transitions from one queue size to another (particularly in an upswing) can be adopted. Another option in this same vein, and the one that has been implemented, is to permit the queue size requirements in each period to vary according to some function of the current and near-future periods' demands. Such an option is particularly

attractive since the data required by it is readily available.

Essentially, our strategy is to conceptually shift the time of the demand requirement at each location so that the queue reflects the satisfaction of future anticipated demands earlier than required. For example, if a two-day queue size is deemed appropriate at a particular loading location, then its demand profile can be shifted ahead by two days, so that the model would essentially try to satisfy each day's demand two days in advance. (Note that we also permit fractional demand day queue sizes. For example a 2.5 day queue size would account for the demand over the next two days plus half the demand on the third day from the present.) By keeping track of the queue size and the actual demand in this time-shifted framework, such a model can lead to a more efficient management of empty cars.

Specifically, let us define the following entities for each loading location. (The location subscript "j" is suppressed for ease in notation.)

AD_τ = actual demand for empty cars at time period τ.

D = number of days (possibly fractional) of future demand that determines the queue size.

RQ_τ = required queue size at time period τ in terms of number of cars
= $AD_{\tau+1} + \ldots + AD_{\tau+\lfloor D \rfloor} + \lceil [D - \lfloor D \rfloor] AD_{\tau+\lfloor D \rfloor+1} \rceil$.

ND_τ = $AD_\tau + RQ_\tau$ = net demand at time period τ.

$OH_{t_{min}}$ = number of empty cars available on-hand at the initial time period t_{min}.

SA_τ = number of empty cars that have already been pre-scheduled (extraneous to the present model decisions) to arrive at time period τ.

NH_τ = net on-hand inventory of cars at time period τ. (Initially,
$NH_{t_{min}} = OH_{t_{min}} + SA_{t_{min}}$ and this is updated by (3.3) as derived next.)

$MD_\tau \ (\equiv d_{j\tau})$ = model demand for empty cars, i.e., demand that is input into the model corresponding to time period τ.

With the above notation, the model demand can be computed as

$$MD_\tau = \ \max\{0, (ND_\tau - NH_\tau)\} \ \forall \tau \geq t_{min}, \tag{3.1}$$

where from above, assuming that the model demand is satisfied on-time,

$$ND_\tau = AD_\tau + RQ_\tau \ \forall \tau \geq t_{min}, \tag{3.2}$$

$$NH_{t_{min}} = OH_{t_{min}} + SA_{t_{min}}, \ \text{and} \ NH_\tau = NH_{\tau-1} + MD_{\tau-1} - AD_{\tau-1} + SA_\tau \ \forall \tau \geq t_{min} + 1. \tag{3.3}$$

Thus, we have

$$MD_\tau \geq RQ_\tau + AD_\tau - NH_\tau \tag{3.4}$$

where equality holds in (3.4) whenever the right-hand side is nonnegative. Also, from (3.1.1) and (3.4), we have,

$$NH_\tau \geq RQ_{\tau-1} + SA_\tau \tag{3.5}$$

where again, (3.5) holds as an equality whenever (3.4) is an equality.

Example 3.1.1

Let us assume that at some loading location, we use a queue size of two-days, that is, $D = 2$ above. Suppose further that the actual demands AD_t for each time period t are given as follows, and that the specified model demands are satisfied on time in each period. Also, assume that $t_{min} \equiv 1, NH_1 = OH_1 + SA_1 = 10, SA_2 = 4, SA_3 = 2$, and that no other scheduled arrivals exist. Then, the queue size, the net demand, the net on-hand inventory, and the model demand are obtained as follows. By definition, the queue size $RQ_\tau = AD_{\tau+1} + AD_{\tau+2}$, and the net demand ND_τ is computed as $AD_\tau + RQ_\tau$ for each time period τ. From (3.1) and the assumption that $NH_1 = 10$, we have $ND_1 - NH_1 = AD_1 + RQ_1 - NH_1 = 10 + 8 - 10 = 8$, and so, $MD_1 = \max\{0, 8\} = 8$. Also, for $\tau = 2$, from (3.3), we have $NH_2 = NH_1 + MD_1 - AD_1 + SA_2 = 10 + 8 - 10 + 4 = 12 = RQ_1 + SA_2$ as in (3.5), since (3.4) holds as an equality. Next, from (3.1), we have $MD_2 = \max\{0, 10 - 12\} = 0$. This gives $NH_3 = NH_2 + MD_2 - AD_2 + SA_3 = 12 + 0 - 5 + 2 = 9$, from (3.3), and so, $MD_3 = \max\{0, 10 - 9\} = 1$ from (3.1). In this manner, we can compute NH_τ and MD_τ for all τ. The results are presented in Table 3.1.

Table 3.1: Computations for Example 3.1.1

$Time(\tau)$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
AD_τ	10	5	3	2	5	7	10	15	20	15	10	5	3	2	0	0
RQ_τ	8	5	7	12	17	25	35	35	25	15	8	5	2	0		
ND_τ	18	10	10	14	22	32	45	50	45	30	18	10	5	2		
NH_τ	10	12	9	7	12	17	25	35	35	25	15	8	5	2		
MD_τ	8	0	1	7	10	15	20	15	10	5	3	2	0	0		

Observe that the strategy for determining the model demand MD_τ is akin to time shifting the demand in that it ensures that with on-time delivery, the car availability equals at least the net demand, which is the actual demand for the present day plus that for the next D days. For instance, in Example 3.1.1, note that the model demands MD_τ match with the time shifted actual demands, after some initial transients (i.e. from time $\tau = 4$ onward).

In fact, if we assume that there are no extraneous pre-scheduled arrivals or any pre-existing initial on-hand inventory, then (3.4) and (3.5) hold as equalities, and we have,

$$
\begin{aligned}
MD_\tau &= RQ_\tau + AD_\tau - NH_\tau \text{ (from (3.4))} \\
&= RQ_\tau + AD_\tau - RQ_{\tau-1} \text{ (from (3.5) since } SA_\tau = 0) \\
&= [AD_{\tau+1} + \ldots + AD_{\tau+\lfloor D \rfloor} + \lceil (D - \lfloor D \rfloor)AD_{\tau+\lfloor D \rfloor+1}]] + AD_\tau - [AD_\tau + \cdots \\
&\quad + AD_{\tau+\lfloor D \rfloor-1} + \lceil (D - \lfloor D \rfloor)AD_{\tau+\lfloor D \rfloor}]].
\end{aligned}
$$

Thus,
$$MD_\tau = \lceil(1 - f) \cdot AD_{\tau+\lfloor D\rfloor}\rceil + \lceil f \cdot AD_{\tau+\lfloor D\rfloor+1}\rceil, \text{ where } f \equiv D - \lfloor D\rfloor, \qquad (3.6)$$

that is, the model demand at period τ is simply the actual demand in period $\tau + D$ if D is an integer (and so f is zero), and is a suitable integerized weighted average of the demands in periods $\tau + \lfloor D\rfloor$ and $\tau + \lfloor D\rfloor + 1$ otherwise, dependent on f as defined in (3.6).

Likewise, if we only assume that no excess on-hand inventory occurs in this process so that (3.4) and (3.5) always hold as equalities, then we have,

$$\begin{aligned}
MD_\tau &= RQ_\tau + AD_\tau - NH_\tau \text{ (from (3.4))} \\
&= RQ_\tau + AD_\tau - RQ_{\tau-1} - SA_\tau \text{ (from(3.5))}
\end{aligned}$$

or that,

$$(MD_\tau + SA_\tau) = RQ_\tau + AD_\tau - RQ_{\tau-1} = \lceil(1 - f) \cdot AD_{\tau+\lfloor D\rfloor}\rceil + \lceil f \cdot AD_{\tau+\lfloor D\rfloor+1}\rceil. \quad (3.7)$$

Therefore, in this case, the model demand to be satisfied at time τ plus the already pre-scheduled arrivals at time τ would equal the time-shifted weighted actual demand given by the right-hand side of Equation (3.6).

Example 3.1.2 (D = 2 period shift)
To illustrate the above formulae, consider the case where we assume no pre-existing net on-hand inventory or extraneous pre-scheduled arrivals.

Table 3.2: Computations for Example 3.1.2.

Time (τ)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
AD_τ	5	6	4	0	0	7	6	4	0	0	6	5	3	0	0
RQ_τ	10	4	0	7	13	10	4	0	6	11	8	3	0		
ND_τ	15	10	4	7	13	17	10	4	6	11	14	8	3		
NH_τ	0	10	4	0	7	13	10	4	0	6	11	8	3		
MD_τ	15	0	0	7	6	4	0	0	6	5	3	0	0		

In this example, we see that the first set of arrivals for the given time framework will commence with the satisfaction of the net demand at $\tau = 1$, which equals the actual demand at $\tau = 1$ plus the required queue size given by the actual demands at $\tau = 2$ and $\tau = 3$. Thereafter, for $\tau \geq 2$, we see that the model demand shifts the actual demand up by two days, so that Equation (3.6) holds true with $f \equiv 0$.

3.1.2 Treatment of Weekends

The demands on Saturdays and/or Sundays at loading locations are frequently zero, depending on the particular plant's or origin's operational practices. Previous tactical decisions that simply considered zero demands on such days resulted in no car movements, since the model was oblivious to the demand for Monday, until Monday entered the planning horizon. This planning defect frequently caused shortages. To alleviate this problem, an arbitrary half-day average weekly demand was imposed for these zero demand weekend days. Instead, our recommendation (that has been adopted) is either to do nothing (since demand is being satisfied D days in advance by the modified queueing considerations), or else, to redefine the queue size RQ_τ as follows:

$RQ'_\tau = $ sum of actual demands over the next first D days that have positive demand.

Using RQ'_τ in (3.2) yields Table 3.3 for Example 3.1.2. In this table, we see how the net effect of the modified strategy is to satisfy the demand that follows some zero demand periods earlier in advance than even the $D = 2$ day time-shift requirement.

The net strategy of treating variable queue sizes and zero weekend demands is summarized below in the next subsection.

Table 3.3: Computations for Example 3.1.2 Revisited.

Time (τ)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
AD_τ	5	6	4	0	0	7	6	4	0	0	6	5	3	0	0
RQ'_τ	10	11	13	13	13	10	10	11	11	11	8				
ND_τ	15	17	17	13	13	17	16	15	11	11	14				
NH_τ	0	10	11	13	13	13	10	10	11	11	11				
MD_τ	15	7	6	0	0	4	6	5	0	0	3				

3.1.3 Summary of Variable Queue Sizes and Treatment of Weekends

A summary of the combined strategy of the foregoing two sub-sections is presented below. For each demand location, we need the following information.

D = number of days (possibly fractional) of future demand that determines the queue size.

AD_τ = actual demand for empty cars at time period t.

$OH_{t_{min}}$ = number of empty cars available on-hand at the initial time period t_{min}.

SA_τ = number of empty cars that have already been pre-scheduled (extraneous to the present model decisions) to arrive at time period τ.

Given the above input specifications, we then compute the required queue size as

$$RQ_\tau = AD_{\tau+1} + \ldots + AD_{\tau+\lfloor D \rfloor} + \lceil [D - \lfloor D \rfloor] AD_{\tau+\lfloor D \rfloor+1} \rceil. \tag{3.8}$$

Then, the demand input required for the model is computed as

MD_τ $(\equiv d_{j\tau}$ for location $j) =$ model demand for location $j = \max\{0, (ND_\tau - NH_\tau)\}$ for all $\tau \geq t_{min}$, $\tag{3.9}$

where

$$ND_\tau = AD_\tau + RQ_\tau = \text{net demand at time period } \tau, \quad \text{for all } \tau \geq t_{min}, \tag{3.10}$$

and where NH_τ = net on-hand inventory at time period τ is given by

$$NH_{t_{min}} = OH_{t_{min}} + SA_{t_{min}}, \quad \text{and } NH_\tau = NH_{\tau-1} + MD_{\tau-1} - AD_{\tau-1} + SA_\tau \ \forall \tau \geq t_{min} + 1. \tag{3.11}$$

At the end of the imminent period, when actual movements have been determined, the model demands for the next run can be re-computed based on the delivered quantities and the movements initiated that determine the revised pre-scheduled arrivals (SA_τ). In this manner, the model data can be revised in a rolling horizon approach.

Weekends, or in general, days of zero actual demands, can be accommodated within the framework of the foregoing variable queue size strategy in one of two ways.

Strategy 1. Do nothing. In this strategy, demands are being satisfied D days in advance (including zero demands).

Strategy 2. Modify the variable queue size requirement RQ_τ to RQ'_τ where RQ'_τ is defined to be the sum of actual demands over the next first D days that have positive demands. (Alternatively, this can be taken as the next D days' demands, skipping only weekends when counting these D days.)

Under either strategy, demand nodes that turn out to have zero demands can simply be left out of the constructed model being input into the program.

Having done with the discussion on the treatment of supply and demand nodes in the model, we switch our attention to the arcs connecting the various nodes in the model and the cost coefficients to be ascribed to these arcs.

3.1.4 Arc and Penalty Cost Structure

Given the knowledge of the sets of supply and demand locations present in the rail network, and considering effective business operations that account for a combination of efficiency and the costs borne by various railroads managing different sections of railroad tracks (see Glickman and Sherali, 1985), the planning staff routinely identifies various origin-destination (O-D) routes to be adopted. Based on acceptable timeliness of delivery (or acceptable degree of latenesses), in turn, an arc connection is made between suitable O-D pairs in the time-space transportation network. Let A denote this set of constructed arcs. Thus, we assume that certain arcs $(p, q) \in A$ have been specified that connect appropriate supply and demand nodes, having respective costs c_{pq} given by the expected transit time from origin i to destination j, where $p \equiv (i, t)$, and $q \equiv (j, \tau)$. Subsequently, these costs will be modified when we consider the development of the time-based nonpreemptive multiple objective function representation along with the inherent uncertainties in transit times.

Of the arcs in A, certain arcs connecting nodes $p = (i, t)$ to node $q = (j, \tau)$ for which t plus the transit time from i to j is lesser than or equal to τ will deliver railcars to location j on time. Let us call these as *on-time* arcs and color code them as *green* arcs. The model allows for 1-day, 2-days late, and 3-or-more days late (empty car) delivery arcs as well in the network. Let us refer to these arcs as *pink, red* and *gray* arcs, respectively. Similar to the *green* arcs, each of these arcs (p, q) also has a corresponding cost c_{pq} assigned to it, given by the expected transit time. Again, in the subsequent process, the model costs for these arcs will be re-computed by ascribing suitable additional lateness penalty factors. Also, to have a balanced problem, let us create a dummy demand node having demand equal to the total supply, and a dummy supply node having supply equal to the total original demand. Accordingly, the arcs connecting the dummy supply node to each real demand node represent unmet demand and are called *black* arcs. For the time being, we will assign a cost of $c_{max} = \max\{c_{pq} : (p, q)$ is a real arc in the problem$\}$ to the black arcs. Later, the penalty on these arcs will be suitably modified. Similarly, we define *white* arcs to be the arcs that connect the supply nodes (including the dummy supply node) to the dummy demand node. These white arcs represent the slack for the associated origin nodes, and have zero costs. Table 3.4 summarizes the color codes for reference purposes. For simplicity we define the set *Green* to be the set of arcs $(p, q) = (it, j\tau)$ that are designated as *green* arcs. In a similar fashion, we define *Pink, Red, Gray, Black* and *White* sets of arcs.

The next step in our modeling process is to construct the classical transportation problem (see Bazaraa, Jarvis and Sherali, 1990, for example). In this model, each decision variable x_{pq} represents the flow of product from the supply node p to the demand node q, and is assumed to be bounded between a lower bound l_{pq} and an upper bound u_{pq}. (Usually, $l_{pq} = 0$ and $u_{pq} = \min\{s_p, d_q\}$, but these quantities can be alternately specified by the user to reflect certain fixed flow designations or *control orders* as they are called - see Chapter 4 for additional details in this regard.) However, our objective function deviates from the standard transportation model as we incorporate a non-preemptive representation of a

multiple objective function structure as discussed next. Anticipating these modified costs, denoted as C_{pq} (including those on the dummy arcs), we present the transportation model for convenience in Equations (3.12) - (3.15) below, where DS and DD respectively refer to the dummy source and the dummy destination nodes.

Table 3.4: Arc Color Codes for Arcs.

Color	Arc Characteristics
Green arcs :	On-time arcs
Pink arcs :	One day late arcs
Red arcs :	Two days late arcs
Gray arcs :	Three or more days late arcs.
White arcs :	Slack arcs that connect supply nodes (including the dummy source node) to the dummy demand node. The flows on such arcs represent unused supplies.
Black arcs :	Artificial flow arcs that connect the dummy source node to the real demand nodes. The (artificial) flows on such arcs represent unmet demands.

$$\text{Minimize} \qquad \sum_{(p,q)\in A} \sum C_{pq}x_{pq} + \sum_{q} C_{DS,q}x_{DS,q} \qquad (3.12)$$

$$\text{subject to} \qquad \sum_{p:(p,q)\in A} x_{pq} + x_{DS,q} = d_q \ \forall q \qquad (3.13)$$

$$\sum_{q:(p,q)\in A} x_{pq} + x_{p,DD} = s_p \ \forall p \qquad (3.14)$$

$$l_{pq} \le x_{pq} \le u_{pq} \ \forall (p,q) \in A, x_{DS,q} \ge 0 \ \forall q, x_{p,DD} \ge 0 \ \forall p. \qquad (3.15)$$

3.2 Time-based Multiple Objective Formulation

3.2.1 Priority Level List Construction

In conducting their daily tasks, railcar managers need to meet several conflicting customer preferences. Shortages in car deliveries with respect to specified demands are surely frowned upon, although the presence of queues at each location somewhat ameliorates its effects. However, shortages are sometimes inevitable, and in such cases, it is naturally better to be late, and by as less an extent as possible, than to have unmet demands. Moreover, timeliness

Number of demand time periods in horizon

		t_{min}	t_{min+1}	...	t_{max-1}	t_{max}
	Pink	...	...	...	5	1
PRMAT =	Red	...	...	...	6	2
	Gray	...	...	...	7	3
	Black	$4(t_{max} - t_{min} + 1)$	...	...	8	4

Figure 3.1: PRMAT Level Ascribing Matrix.

in delivery is more pertinent for the earlier time periods than for later days in the horizon, since the degree of uncertainty in supply and demand grows with time over the horizon. Thus, it is apparent that our model needs to address multiple objectives.

Let us begin by defining

$$f_{(\text{Color, demand time})} = \sum_{(p,q)\equiv(it,j\tau) \in \text{ Color}: \tau= \text{ demand time}} x_{pq} \qquad (3.16)$$

where Color $=$ *Pink, Red, Gray*, and *Black*, and where demand time $\tau \in [t_{min}, t_{max}]$. (Note that when Color $=$ *Black*, the arcs $(p, q) \in$ Color correspond to $p \equiv DS$.) We will also define the expected car-days based objective function (including the tentative penalty cost of c_{max} attached to unmet demand arcs) as

$$f_{ECD} = \sum_{(p,q)\in A} \sum c_{pq}x_{pq} + \sum_q c_{max}x_{DS,q}. \qquad (3.17)$$

Next, let us construct a *level table* that indicates the level of priority attached to minimizing each of the objective functions defined in (3.17). According to Table 3.5, we would like to minimize the (total) flow on the black (unmet demand) arcs for the imminent period with the first priority, then minimize the flow associated with 3 or more days late arcs, 2 days late arcs, and one day late arcs in this order for the imminent period, and then continue down the horizon for the remaining periods, finally, seeking to minimize the expected car-days cost. Note that in this hierarchy, an on-time delivery is encouraged for the earlier periods, beginning with the imminent period, by minimizing shortages and latenesses for these periods with a relatively higher priority. The list of the level table is described more succinctly by an equivalent representation via a (time-based) *priority matrix*, denoted by *PRMAT*, as shown as in Figure 3.1.

Note that this matrix PRMAT specifies levels 1, 2, ..., $4(t_{max} - t_{min} + 1)$ for each color and time period to match that in Table 3.5, so that *the higher the level, the greater the relative desirability of minimizing the cost weighted flows on that type of arc.*

Given the priority list and the objective functions defined in (3.17), our *preemptive priority*

Table 3.5: Level Table.

Demand time-period (t_{min})	black arcs
	gray arcs
	red arcs
	pink arcs
Demand time-period ($t_{min} + 1$)	black arcs
	gray arcs
	red arcs
	pink arcs
•	•
•	•
•	•
Demand time-period t_{max}	black arcs
	gray arcs
	red arcs
	pink arcs

multiple objective problem seeks to solve

$$Lexmin\{f_{(black,t_{min})}, f_{(gray,t_{min})}, \ldots, \text{ as per the Level Table (3.5), up to}, \ldots, f_{(pink,t_{max})}, f_{ECD}\} \tag{3.18}$$

$$\text{subject to} \quad \sum_{p:(p,q)\in A} x_{pq} + x_{DS,q} = d_q \ \forall q$$

$$\sum_{q:(p,q)\in A} x_{pq} + x_{p,DD} = s_p \ \forall p$$

$$l_{pq} \le x_{pq} \le u_{pq} \ \forall (p,q) \in A, x_{DS,q} \ge 0 \ \forall q, x_{p,DD} \ge 0 \ \forall p.$$

To control various functional orderings in (3.18) according to various managerial philosophies regarding the compromise between a lateness in a certain time period versus a greater lateness or shortage in a subsequent time period, we define a parameter called **TIMEPR** whose value ranges in $\{1, ..., 5\}$. This value prompts how to construct the corresponding "user-specified" matrix PRMAT. When **TIMEPR** $= 1$ (the default), PRMAT will give a higher priority to earlier demand time-periods than later ones, arranging the color of arcs as shown in Table 3.5 for each time period of demand, with unmet demands having the highest priority of being driven to zero, followed by various degrees of diminishing lateness. When **TIMEPR** $= 2$, PRMAT will give a higher priority to reducing flows on the *black* arcs over all the demand time-periods first, before reverting to the above scheme of considering the different colored arcs within each demand time-period, taken in order of increasing time. Similarly, when **TIMEPR** $= 3$, PRMAT will assign the highest priority to reducing the flow on the black

arcs over the various demand time-periods, then it will consider the minimization of the flows on the gray arcs next over the various demand time-periods, and finally, it will revert to considering each demand period in turn, arranging the remaining colors with diminishing lateness within each period. Cases where **TIMEPR** = 4 and 5 follow the same logic as above. To illustrate this idea, PRMAT with **TIMEPR** = 1 and 3 are shown next. (A user-specific arrangement of levels for these multiple objectives within PRMAT is also provided as an option.)

TIMEPR = 1 (default)
Horizon = 5 days ($t_{max} = t_{min} + 4$)

	t_{min}	$t_{min} + 1$	$t_{min} + 2$	$t_{min} + 3$	$t_{min} + 4$
Pink arcs	17	13	9	5	1
Red arcs	18	14	10	6	2
Gray arcs	19	15	11	7	3
Black arcs	20	16	12	8	4

TIMEPR = 3
Horizon = 5 days ($t_{max} = t_{min} + 4$)

	t_{min}	$t_{min} + 1$	$t_{min} + 2$	$t_{min} + 3$	$t_{min} + 4$
Pink arcs	9	7	5	3	1
Red arcs	10	8	6	4	2
Gray arcs	15	14	13	12	11
Black arcs	20	19	18	17	16

3.2.2 Computation of Cost Coefficients for a Nonpreemptive Objective Function

Sherali and Soyster (1983) and Sherali (1982, 1988) prescribe a set of weights for which a single nonpreemptive objective function, constructed as a corresponding weighted sum of the multiple objective functions, produces an equivalent problem to the preemptive priority multiple objective program. Given our priority level list, we can use these results to find appropriate weights for each objective function to reflect its specified level of priority in the composite multi-objective hierarchy. This can be done as follows.

Consider a *preemptive priority* linear optimization problem given by

$$Lexmin\{c_1 \cdot x, c_2 \cdot x, \ldots, c_r \cdot x : x \in X\} \tag{3.19}$$

where X is a non-empty polytope. (The case of unbounded sets X, as well as nonlinear and discrete problems are also treated in Sherali (1982) and Sherali and Soyster (1983)). For each $i = 1, \ldots, r$, let γ_i denote the range of values that $c_i \cdot x$ can take over X (assumed

to be non-zero without loss of generality), and let α_i denote the smallest non-zero absolute difference in the value of $c_i \cdot x$ realized between any pair of vertices of X. Define

$$\gamma = \text{maximum}_{i=1,\ldots,r}\gamma_i \tag{3.20}$$

and

$$\alpha = \text{minimum}_{i=1,\ldots,r}\alpha_i. \tag{3.21}$$

Then, for any value of

$$\mathcal{M} \geq \lceil 1 + \frac{\gamma}{\alpha} \rceil \tag{3.22}$$

Sherali and Soyster show that the optimal solution set for the *nonpreemptive* problem

$$\text{Minimize}\{\sum_{i=1}^{r}\mathcal{M}^{r-i} \cdot c_i \cdot x : x \in X\} \tag{3.23}$$

precisely matches that for (3.19). Note that for our problem, it is legitimate to take $\alpha = 1$ in Equation (3.22). (Sherali (1982) gives various other procedures for computing tighter bounds on $\mathcal{M}$ than that specified by (3.22) for several special problems.)

Furthermore, consider a transportation problem on m supply nodes and n demand nodes, and having an arc set A. Suppose that we balance this problem by adding a dummy source and a dummy sink node as before, and that we are interested in knowing a valid penalty $\mathcal{M}$ that could be ascribed to the unmet ("black") demand arcs so as to make the total flow on these artificial arcs as small as possible while minimizing the cost related to the remaining real flow. Note that this is a preemptive priority bicriterion problem of minimizing the total shortage with the first priority, and minimizing the real transportation cost with the second priority, while satisfying the supply and demand constraints. Sherali (1988) proves that this problem may be solved as an equivalent nonpreemptive problem in which

$$\mathcal{M} = 1 \quad +(\text{ sum of the } W \text{ largest } c_{pq} \text{ values})$$
$$-(\text{ sum of the } (W-1) \text{ smallest } c_{pq} \text{ values} \tag{3.24}$$
$$\text{where, } W = \min\{m, n, |A|/2\}. \tag{3.25}$$

Observe that if we simply had *green* and *black* arcs, we could use Equations (3.24) and (3.25) to derive the penalized costs C_{pq} for (3.12). However, given our multiple objective structure (3.18), in theory, we need to apply (3.20) - (3.23). Since it is legitimate to take γ greater than or equal to the value specified by (3.20), we could easily compute a valid upper bound by using the bounds on the flows. Similarly, as mentioned above, due to the integrality of our data and the total unimodularity of the transportation constraints, we can take $\alpha = 1$. However, because of the large number of objective functions under present consideration, the theoretical values of these weights for use in (3.23) can get prohibitively high, and might not even be representable on the computer. Moreover, we will subsequently be introducing customer based priorities, which will in turn split each objective function in

(3.17) into several prioritized objectives, one for each demand location. Although we can directly solve the preemptive priority problem via network flow techniques by sequentially restricting arc flows based on reduced costs or duality considerations while progressing from one objective function to the next, we will need to use a nonpreemptive representation when we consider blocking issues. Besides, it is more a *desire* than an *absolute necessity* to enforce strict preemption in the multiple objective function structure of Table 3.5. Hence, we will adapt the constructs in the aforementioned theoretical papers in a heuristic fashion.

First, following the derivation in Sherali (1988), let us define the base penalty parameter $\mathcal{M}$ as given by (3.24). For the sake of computational convenience, we will use an upper bound on this value as given by

$$\mathcal{M} = 1 + W \cdot c_{max} - (W - 1) \cdot c_{min} \tag{3.26}$$

where c_{max} = maximum (original) cost over all *green, pink, red,* and *gray* arcs in the problem, c_{min} = minimum (original) cost over all *green, pink, red,* and *gray* arcs in the problem. Now, given the base penalty $\mathcal{M}$, and given a parameter $(1 < \theta \leq 4)$ selected as specified below, we tentatively derive the nonpreemptive objective function as

$$f_{ECD} + \sum_{l=1}^{l_{max}} (\theta^l \mathcal{M} \cdot (\text{ objective} f_{(\text{color, demand time})} \text{ at level } l). \tag{3.27}$$

Note that in lieu of using weights given by exponents of some suitable penalty parameter as in (3.23), we have adopted a scheme in (3.27) that composes weights as multiples of penalty $\mathcal{M}$ (motivated by (3.24)), where the multiples are exponents of a parameter θ to be determined later. This yields the cost C_{pq} for arc (p, q) in the single nonpreemptive objective function as

$$C_{pq} = \begin{cases} c_{pq} & \text{for } (p, q) \in Green \cup White \\ c_{pq} + \theta^l \cdot \mathcal{M}, & \text{otherwise for } (p, q) \in Pink \cup Red \cup Gray \cup Black \end{cases} \tag{3.28}$$

where l = level of the arc (p, q) in the hierarchy of objective functions. Note that the maximum penalized cost in (3.28) will occur for the black (unmet demand) arc corresponding to the imminent time period, and is given by

$$MC = c_{max} + \theta^{l_{max}} \cdot \mathcal{M} \tag{3.29}$$

where $l_{max} = 4 \cdot (t_{max} - t_{min} + 1)$ is the maximum level in the hierarchy. Hence, if V is the largest integer value permissible for use in the program, we compute a suitable value of θ as

$$\max\{1 < \theta \leq 4 : MC \leq V\}. \tag{3.30}$$

Based on empirical studies, we discovered that by taking θ as large as about 4 (whenever permissible), we obtain a sufficient level of discrimination among the objective functions, as ascertained by actually conducting a sequential minimization of these multiple objectives

(see Section 5.4.3 for some related results). This motivates the upper bound on θ in (3.30). Moreover, whenever the horizon look-ahead duration $(t_{max} - t_{min} + 1)$ is at least 4 - 5 days, the value of θ given by (3.30) is typically less than 3. Assuming that $c_{max} + \mathcal{M} < V$, this value of θ is given by $\min\{4, ((V - c_{max})/\mathcal{M})^{1/l_{max}}\}$.

We now address a refinement in (3.28) for the gray arcs. Note that (3.28) ascribes the same penalty for all the gray arcs, while the lateness (≥ 3 days) for the gray arcs might vary. Hence, in this case, we derive a revised formula as described below.

Consider the set of gray arcs at some level l. The penalty cost for this arc is taken to lie in the interval $[a_l, b_l]$ centered at $\theta^l \cdot \mathcal{M}$, where the left-hand interval end-point a_l is given by the mid-point of $\theta^l \cdot \mathcal{M}$ and the previous level's weight $\theta^{l-1} \cdot \mathcal{M}$. Hence, we obtain

$$a_l = \left[\theta^l - \frac{\theta^l - \theta^{l-1}}{2}\right]\mathcal{M} \text{ and } b_l = \left[\theta^l + \frac{\theta^l - \theta^{l-1}}{2}\right]\mathcal{M}, \qquad (3.31)$$

with the actual values within this interval determined as follows. Denote

$$maxgray = \text{ maximum lateness value } (\geq 3) \text{ for the gray arcs at this level } l \qquad (3.32)$$

and let

$$d_l = (maxgray - 2). \qquad (3.33)$$

Then, we divide $[a_l, b_l]$ into d_l equal subintervals, and set the penalty cost for a lateness $LAT_{pq} \in [3, maxgray]$ at the midpoint of the $(LAT_{pq} - 2)^{th}$ subinterval. Hence, we get for any gray arc (p, q) at level l,

$$C_{pq} = c_{pq} + \left\lceil\left[a_l + \frac{\theta^l - \theta^{l-1}}{d_l}(LAT_{pq} - 2.5)\right]\mathcal{M}\right\rceil. \qquad (3.34)$$

The following section provides additional motivation for this construct, where a similar approach is adopted to revise these tentative costs by virtue of considering various demand location priorities specified by the automobile manufacturers along with related equity issues. This is discussed next, followed by a discussion on further modifications in order to address uncertainty in the problem.

3.3   Demand Location-based Priority Considerations

3.3.1   Market-driven Tool

As a tactical planning tool, our model is envisaged to accommodate market-driven preferences. The automobile manufacturers own various plants at different locations, and based on the characteristics of automobile demand in the particular regions, they would like to be
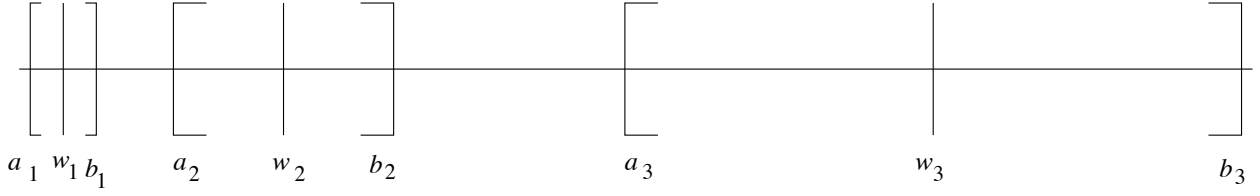
Figure 3.2: Penalty Intervals for Different Levels.

able to ascribe relative priorities for satisfying empty-car demand in a timely fashion at these different plants. To facilitate this option, we incorporate relative priorities that are specified by each shipper on a scale between 1 and 10 for each demand location corresponding to the plants owned by this shipper. This additional prioritization scheme means that for the same color and demand time (or level), the objective function $f_{(Color, demand\ time)}$ needs to be further decomposed or partitioned into as many components as there are distinct specified priorities for the involved demand locations j in this function. In this decomposition, the higher the location priority level, the earlier should the corresponding component of the overall objective function appear in the hierarchy of objective functions. However, instead of recomputing the exponential penalties based on an increased number of levels, in order to reflect a simple linear ordinal distinction between these priorities, we linearly spread the associated penalty values for arcs at any level l over a range $(\theta^l - \theta^{l-1})\mathcal{M}$ centered at the penalty value $\theta^l \mathcal{M}$, depending on the given location priority. Specifically, for each level l, define

$$a_l = \left[\theta^l - \frac{\theta^l - \theta^{l-1}}{2}\right]\mathcal{M} \text{ and } b_l = \left[\theta^l + \frac{\theta^l - \theta^{l-1}}{2}\right]\mathcal{M}, \qquad (3.35)$$

and let $w_l = \theta^l \mathcal{M}$ be the center of the interval $[a_l, b_l]$. Figure 3.2 illustrates these ranges. Note that for $l \geq 2$, the left-hand end-point a_l of the lth interval is given by the average of the mid-point of its interval and the mid-point w_{l-1} of the previous interval, and its distance from the right-hand end-point of the previous interval, is given by $(a_l - b_{l-1}) = 0.5 \cdot \theta^{l-2} \cdot (\theta - 1)^2 \mathcal{M}$. This difference reflects the distinction being made between minimizing the shortages corresponding to the lowest priority location at the current level, and the highest priority location at the previous level. Observe that as the levels increase, this distinction becomes more stark. This is desirable since the imminent period shortages appear at higher levels, and so, this hierarchy of penalties automatically imparts a greater distinction between the higher levels, thereby reflecting a relatively greater importance to the imminent period decisions. For example, if $\theta = 4$, we will have $[a_1, b_1] = [2.5, 5.5]\mathcal{M}$, $[a_2, b_2] = [10, 22]\mathcal{M}$, $[a_3, b_3] = [40, 88]\mathcal{M}$, $[a_4, b_4] = [160, 352]\mathcal{M}$, etc. Assuming that $t_{max} - t_{min} + 1 = 5$, $[a_{l_{max}-1}, b_{l_{max}-1}] = [1.72 \times 10^{11}, 3.78 \times 10^{11}]\mathcal{M}$, and $[a_{l_{max}}, b_{l_{max}}] = [6.87 \times 10^{11}, 15.12 \times 10^{11}]\mathcal{M}$. (Note the hierarchy in penalty values.)

The actual penalized cost factor for each arc (p, q) at level l is then spread uniformly across the interval $[a_l, b_l]$, being taken as a_l if the corresponding location priority $P_{pq} = 1$ and as

b_l if $P_{pq} = 10$. (Note that for a given arc (p, q), for convenience, we designate the specified location priority to be P_{pq}, although this depends only on the location j corresponding to the demand node $q \equiv (j, \tau)$.) For the gray arcs, this spreading of penalties based on priorities is done for each sub-interval of $[a_l, b_l]$ that corresponds to a particular (gray) lateness. Thus, by this process, the computed penalty factor for each level and demand level priority is precisely the weight attached to the corresponding decomposed objective function in the nonpreemptive weighting scheme. The costs for arcs (p, q) of each color are then derived as follows, dependent on their level l and the demand location based priority P_{pq}.

1. $C_{pq} = 0$ for $(p, q) \in White$. (3.36)

2. $C_{pq} = c_{pq}$ for $(p, q) \in Green$. (3.37)

3. $C_{pq} = c_{pq} + \left\lceil \dfrac{\theta^{l-1}}{18} \left[\theta(7 + 2\ P_{pq}) + (11 - 2\ P_{pq}) \right] \mathcal{M} \right\rceil$ (3.38)

   for $(p, q) \in Pink \cup Red$.

4. $C_{pq} = c_{pq} + \left\lceil \left[a_l + \left(\dfrac{LAT_{pq} - 3}{d_l} \right)(\theta^l - \theta^{l-1}) + \left(\dfrac{P_{pq} - 1}{9} \right) \dfrac{\theta^l - \theta^{l-1}}{d_l} \right] \mathcal{M} \right\rceil$ (3.39)

   for $(p, q) \in Gray$, where d_l is given by Equation (3.33). This equation can be rewritten as

   $C_{pq} = c_{pq} + \left\lceil \dfrac{\theta^{l-1}}{18 d_l} [9d(\theta + 1) + 2(\theta - 1)(9\ LAT_{pq} + P_{pq} - 28)] \mathcal{M} \right\rceil$ (3.40)

5. $C_{pq} = c_{max} + \left\lceil \dfrac{\theta^{l-1}}{18} [\theta(7 + 2P_{pq}) + (11 - 2P_{pq})] \mathcal{M} \right\rceil$ (3.41)

   for $(p, q) \in Black$.

From the equation for the black arcs, we can deduce that the maximum possible cost, MC, in the model is now given by

$$MC = c_{max} + \frac{\theta^{l_{max}-1}}{2}(3\theta - 1)\mathcal{M}$$ (3.42)

where $l_{max} = 4 \cdot (t_{max} - t_{min} + 1)$ is the highest level in the model. Using the same motivation as for (3.30), the value of θ is computed as

$$\max\{1 < \theta \leq 4 : MC \leq V\}$$ (3.43)

by using a simple Newton-based scheme.

3.3.2 Equity-driven Tool

An issue common in any pooling strategy is the distribution of cost or profit among the participants. In our model, we also need to determine a fair scheme for distributing cars

among the shippers. The key question that arises is that in times of shortages, which shipper should be given a relatively higher priority for having demand satisfied in a timely fashion, and to what extent? We begin this sub-section by describing the current accepted equity philosophy among shippers. Subsequently, we use this criteria to derive a quantitative priority structure that governs the distribution implicitly within the model according to this philosophy.

Presently, the fleet sizing model due to Sherali and Tuncbilek (1997) is being run separately for each car type, but with all the shippers pooled as a group. (These car types are bilevels, trilevels, and special TTQ-X cars used by Chrysler.) Based on this fleet sizing exercise, the total loaded plus empty annual car-days is computed for each shipper, and this indicator serves as one input into shipper-railroad negotiations that ultimately determine the actual fleet size that is *assigned* to each shipper and that is provided by each railroad company. The equity recommendation that appears to have been accepted by the involved parties is to conduct empty car distribution decisions for the joint pool of shippers in a manner that results in an empty plus loaded car-days for each shipper that is directly proportional to the assigned car-days.

The manner in which this is presently being implemented is to partition the horizon into two-week periods, and to try and satisfy this equity relationship in an ad-hoc fashion for each such period by deciding where shortages should be absorbed whenever they occur. This practice is oblivious to the time-variation in demand among the various shippers, and attempts to force an average value on an inherently time-varying process. Besides, this is not being done within the framework of the model.

To formalize this mechanism and to somewhat alleviate its shortcomings, three specific recommendations are made. First, the equity computations should be done on a rolling horizon basis, rather than on non-overlapping blocks of time. Second, this rolling horizon duration should be sufficiently long so as to make the proportionality relationship with the average value more meaningful, since afterall, this type of a proportionality is a long-term issue that needs to have been met over the year under consideration. Third, this needs to be knitted into the framework of the tactical model.

Toward this end, suppose that the tactical model is to be run for some designated time horizon, and that for each of K shippers, we wish to consider the total loaded plus empty car-days over this duration as well as over some prior days that correspond to some total duration T. This value T might equal three months or the cumulative time since the beginning of the year, whichever is greater. Let Q_k denote the number of loaded plus empty car-days over the prior segment of this duration ("prior" with respect to the model run), plus the (known or estimated) loaded car-days over the model horizon. Then, in terms of the model flow decisions x, the total loaded plus empty car-days over the duration T would equal

$$Q_k + \sum_{(p,q) \in A_k} \sum c_{pq} x_{pq} \tag{3.44}$$

where A_k denotes the subset of arcs A that correspond to shipper k, for each $k = 1, \ldots, K$.

Now, if FS_k denotes the assigned fleet size for shipper k, we would like the quantity in (3.44) to be directly proportional (via a constant σ_k, say) to the assigned car-days $T(FS_k)$. Ideally, all the quantities σ_k should equal some constant value σ. However, to permit a degree of flexibility, we can allow a spread of some amount 2δ in the σ_k values, and hence incorporate the following constraints.

$$Q_k + \sum_{(p,q)\in A_k} \sum c_{pq} x_{pq} = T(FS_k)\sigma_k \quad \forall k = 1, \ldots, K \tag{3.45}$$

$$-\delta \leq \sigma_k - \sigma \leq \delta \quad \forall k = 1, \ldots, K. \tag{3.46}$$

Note that $\sigma_k, k = 1, \ldots, K$, as well as σ are variables (in addition to x), but δ is a pre-specified constant. To soften these constraints, δ could also be designated as a variable, and an associated penalty term $(\mathcal{M}_\delta)\delta$ could be incorporated within the objective function, where $\mathcal{M}_\delta$ reflects the relative penalty imposed for having a deviation in the proportionality constant σ_k from a common value. In lieu of using the constraints (3.45) - (3.46), since these constraints ruin our underlying network structure, we incorporate the concept of equity embodied by these constraints within the priority structure of the previous section through a suitable scaling mechanism.

Note that since the shippers have been pooled in the present model, and since each shipper specifies priorities on a scale of 1-10 (with 10 being the highest priority of being timely in satisfying demand), it becomes necessary to control the relative magnitudes of specified priorities among the shippers. For example, if some shipper specifies a priority of 10 for all its demand locations, this could bias the flow in its favor. Although the equity constraints (3.45) and (3.46) would tend to counter this effect, we would like to balance the priorities to indirectly reflect these constraints via suitable scaling factors $F_k \in (0, 1]$. In other words, we would like to derive scaling factors $F_k \in (0, 1]$ with which to multiply all priorities specified by shipper $k, k = 1, \ldots, K$, with the motivation to make the scaled priorities reflect past performance and existing inequities, and hence more closely monitor the attainment of equity, while dispensing with the explicit equity constraints (3.45) and (3.46).

Toward this end, let AP_k denote the average weighted plant priority based on the shipper designated priorities $P_{kj} \in [1, 10]$ for each plant $j \in \mathcal{P}_k$, for each $k = 1, \ldots, K$, where $\mathcal{P}_k$ is the set of demand locations associated with shipper $k, k = 1, \ldots, K$. (Note that P_{kj} corresponds to the arc-based priority specification P_{pq} defined earlier, where arc (p, q) corresponds to $q = (j, \tau)$ for $j \in \mathcal{P}_k$.) Specifically,

$$AP_k = \frac{\sum_{j\in\mathcal{P}_k} \sum_\tau \overline{P}_{kj} d_{j\tau}}{\sum_{j\in\mathcal{P}_k} \sum_\tau d_{j\tau}} \quad \forall k = 1, \ldots, K. \tag{3.47}$$

Now, as in (3.45), let R_k denote the compensation in the car-days Q_k required in order to make $Q_k + R_k$ directly proportional to $(T)(FS_k)$, where FS_k is the fleet size assigned to shipper k as defined above. That is, for some proportionality constant σ, we would like R_k to satisfy

$$Q_k + R_k = \sigma(T)(FS_k) \quad \forall k = 1, ..., K. \tag{3.48}$$

Among the alternative solutions to (3.48), in order to select a particular solution that makes R_k commensurate with Q_k, let us set

$$\sigma = \frac{1}{KT} \sum_{k=1}^{K} \left(\frac{Q_k}{FS_k} \right). \tag{3.49}$$

Note that σ denotes the average of the car utilization factors $Q_k / T \cdot FS_k$ for $k = 1, \ldots, K$, over the duration T, for the K shippers. Hence, the equity relationship (3.48) requires the total compensated car-days $Q_k + R_k$ to match the assigned car days $(T \cdot FS_k)$ times this average car utilization factor for each shipper $k = 1, \ldots, K$. Substituting this value of σ in (3.48), we obtain

$$R_k = \left(\frac{FS_k}{K} \right) \sum_{k=1}^{K} \left(\frac{Q_r}{FS_r} \right) - Q_k \quad \forall k = 1, \ldots, K. \tag{3.50}$$

Note that this compensation R_k can be of either sign. In particular, if Q_k is directly proportional to FS_k $\forall k$, i.e., Q_k / FS_k is a constant over $k = 1, \ldots, K$, then we would have $R_k \equiv 0$ $\forall k = 1, \ldots, K$. Let R be the range of the spread of the R_k values, i.e.,

$$R = R_{max} - R_{min} \tag{3.51}$$

$$\text{where } R_{max} = \max_k \{R_k\} \text{ and } R_{min} = \min_k \{R_k\}. \tag{3.52}$$

Our intent now is to derive scaling factor values $F_k \in (0, 1], k = 1, \ldots, K$, so that the relative spread of the scaled average weighted priorities $(AP_k)F_k$ would be congruent to the spread of the R_k values. Hence, the relative values of the scaled average weighted priorities would then reflect a similar proportional spread to that of the relative car-day compensation requirements.

To achieve this, let the parameter $\lambda = 0$ if $R = 0$, and if $R \neq 0$, then select any value $0 < \lambda < 1$ (a recommended value is $\lambda = 0.5$). The interpretation of λ, as seen below, is that it represents *the ratio of the range for the scaled average priorities to their maximum value (whenever non-zero).*

Consider the following computations. Let

$$\overline{R}_k = \begin{cases} \frac{R_{max} - R_k}{R} & \text{if } R \neq 0 \\ 0 & \text{if } R = 0 \end{cases} \quad \forall k = 1, \ldots, K. \tag{3.53}$$

Note that $\overline{R}_k$ is a convex combination weight which determines

$$R_k = (\overline{R}_k) R_{min} + (1 - \overline{R}_k) R_{max} \tag{3.54}$$

within the range $[R_{min}, R_{max}]$. We would like $(AP_k)F_k$ to split the range $[L, U]$, say, of the scaled average weighted priorities in the same proportion as R_k splits $[R_{min}, R_{max}]$, for each $k = 1, \ldots, K$. Suppose that we set

$$U = \text{minimum}_{k=1,\ldots,K} \left[\frac{AP_k}{1 - \lambda \overline{R}_k} \right], \quad L = U(1 - \lambda) \tag{3.55}$$

and we compute

$$F_k = \frac{U(1 - \lambda \overline{R}_k)}{AP_k} \quad \forall k = 1, \dots, K. \tag{3.56}$$

Then, consider the following result.

Proposition 3.1. Let $[L, U]$ and $F_k, k = 1, \dots, K$ be given by (3.55) and (3.56). Then $0 < F_k \leq 1 \ \forall k = 1, \dots, K, U = \max_k \{F_k(AP_k)\}, L = \min_k \{F_k(AP_k)\}$, and

$$(AP_k)F_k = (\overline{R}_k)L + (1 - \overline{R}_k)U \quad \forall k = 1, \dots, K. \tag{3.57}$$

Proof. Since $(1 - \lambda \overline{R}_k) > 0 \ \forall k$, we clearly have $F_k > 0 \ \forall k$. Moreover from (3.55),

$$U \leq \frac{AP_k}{1 - \lambda \overline{R}_k} \ \forall k \quad \Rightarrow F_k \equiv \frac{U(1 - \lambda \overline{R}_k)}{AP_k} \leq 1, \quad \forall k = 1, \dots, K. \tag{3.58}$$

Furthermore, by (3.56), we have

$$(AP_k)F_k = U[1 - \lambda \overline{R}_k] = \overline{R}_k U(1 - \lambda) + (1 - \overline{R}_k)U = (\overline{R}_k)L + (1 - \overline{R}_k)U \ \forall k. \tag{3.59}$$

Hence, (3.57) holds true, and so, $L \leq (AP_k)F_k \leq U \ \forall k$. Moreover, if $R \neq 0$, then since $\overline{R}_k = 0$ for some $k \in \{1, \dots, K\}$ and $\overline{R}_k = 1$ for some $k \in \{1, \dots, K\}$, we have

$$L = \min_k \{F_k(AP_k)\} \text{ and } U = \max_k \{F_k(AP_k)\}. \tag{3.60}$$

On the other hand, if $R = 0$, then $\overline{R}_k = 0 \ \forall k, \lambda = 0$, and $(AP_k)F_k = U = L \ \forall k = 1, \dots, K$, and so (3.60) again holds true. This completes the proof.□

Example 3.3.1.

Suppose that $K = 3$ and that we have the data given in Table 3.6.

Table 3.6: Data for Example 3.3.1.

	FS_k	Q_k	AP_k
$k = 1$	20	100	6
$k = 2$	30	80	8
$k = 3$	40	120	7

Hence, $\Sigma_r(Q_r/FS_r) = 5 + 2.66 + 3 = 10.66$, and from (3.50), we obtain $R_1 = -28.88, R_2 = 26.66$, and $R_3 = 22.22$. Using (3.53), this yields $\overline{R}_1 = 1, \overline{R}_2 = 0$, and $\overline{R}_3 = 0.08$. Choosing $\lambda = 0.5$, we compute U and L from (3.55) as

$$U = \min\{12, 8, 7.29\} = 7.29, \text{ and } L = 7.29(0.5) = 3.645.$$

Then, from (3.56), we obtain the following scaling factors.

$$F_1 = 0.6075, F_2 = 0.91125, \text{ and } F_3 = 1.$$

Note that the corresponding scaled weighted average priorities are given by

$$(AP_1)F_1 = 3.645, (AP_2)F_2 = 7.29, \text{ and } (AP_3)F_3 = 7.00.$$

Examining the given data tabulated in Table 3.6 with respect to relative values of FS_k versus relative values of Q_k, these results appear to be intuitively reasonable.

Example 3.3.2.

Suppose that $K = 2$ and that we have the information given in Table 3.7.

Table 3.7: Information for Example 3.3.2.

	R_k	AP_k
$k = 1$	-10	10
$k = 2$	10	5.5

Using (3.53), we obtain

$$\overline{R}_1 = 1, \text{ and } \overline{R}_2 = 0.$$

Choosing $\lambda = 0.5$, we compute U and L from (3.55) as

$$U = \min\{20, 5.5\} = 5.5, \text{ and } L = 5.5(0.5) = 2.75.$$

Then, from (3.56), we obtain the following scaling factors.

$$F_1 = 0.275, \text{ and } F_2 = 1.$$

Note that the corresponding scaled weighted average priorities are given by

$$(AP_1)F_1 = 2.75, \text{ and } (AP_2)F_2 = 5.5.$$

This example illustrates that while Shipper 1 has been enjoying a disproportionate usage of empty railcars with respect to its allocation and has yet specified a priority of 10 for all its plants (as opposed to an average scaled priority of 5.5 specified by Shipper 2), our scaling scheme rectifies this situation by reducing Shipper 1's priorities to 2.75 and retaining Shipper 2's priorities as specified.

Example 3.3.3.

Suppose that $K = 2$ and that we have the information given in Table 3.8. Hence, we have the same computed values of R_k as in Example 3.3.2, but this time, both shippers have specified priorities of 10 for all their plants.

Table 3.8: Information for Example 3.3.3.

	R_k	AP_k
$k = 1$	-10	10
$k = 2$	10	10

Using (3.53), this yields

$$\overline{R}_1 = 1, \text{ and } \overline{R}_2 = 0.$$

Choosing $\lambda = 0.5$, we compute U and L from (3.55) as

$$U = \min\{20, 10\} = 10, \text{ and } L = 10(0.5) = 5.$$

Then, from (3.56), we obtain the following scaling factors.

$$F_1 = 0.5, \text{ and } F_2 = 1.$$

Note that the corresponding scaled weighted average priorities are given by

$$(AP_1)F_1 = 5, \text{ and } (AP_2)F_2 = 10.$$

Hence, in this example, for the same assigned priority values, the scaling factors are governed by the compensation R_k in the car-days required, and are given by 1 and $(1 - \lambda)$, respectively, for the shipper requiring a positive and a negative compensation. This also motivates our recommended choice of $\lambda = 0.5$.

Example 3.3.4.

Suppose that $K = 2$ and that we have the data given in Table 3.8, except that $R_1 = R_2$. In this case, the corresponding scaled weighted average priorities are simply given by the original AP_k values because $R = 0$.

In summary, our objective function coefficients C_{pq} are thus far determined as in (3.36) - (3.41), except that each P_{pq} is replaced by $P'_{pq} = P_{pq} \cdot F_k$ where $q \equiv (j, \tau)$ with $j \in \mathcal{P}_k$, and where the scaling factors F_k are computed as follows, using $\lambda = 0.5$.

$$AP_k = \frac{\sum_{j \in \mathcal{P}_k} \sum_{\tau} \overline{P}_{kj} d_{j\tau}}{\sum_{j \in \mathcal{P}_k} \sum_{\tau} d_{j\tau}} \quad \forall k = 1, \dots, K,$$

$$R_k = \left(\frac{FS_k}{K}\right) \sum_{k=1}^{K} \left(\frac{Q_r}{FS_r}\right) - Q_k \quad \forall k = 1, \dots, K,$$

$$\overline{R}_k = \begin{cases} \frac{R_{max} - R_k}{R} & \text{if } R \neq 0 \\ 0 & \text{if } R = 0 \end{cases} \quad \forall k = 1, \dots, K,$$

$$U = \underset{k = 1, \dots, K}{\text{minimum}} \left[\frac{AP_k}{1 - \lambda \overline{R}_k}\right], \quad L = U(1 - \lambda), \text{ and}$$

$$F_k = \frac{U(1 - \lambda \overline{R}_k)}{AP_k} \quad \forall k = 1, \dots, K.$$

Thus, given the shipper designated priorities (P_{pq} or equivalently $\overline{P}_{kj}$), the amount of demand at a particular location ($d_{j\tau}$), a scaling parameter λ (= 0.5, say), the assigned fleet size for shipper (FS_k), the number of loaded plus empty car-days over the prior segment of this duration as well as the (known or estimated) loaded car-days over the model horizon (Q_k), and the set of demand locations associated with shipper k ($\mathcal{P}_k$), we compute the scaling factors $F_k, k = 1, \ldots, K$ as stated above. The scaled priorities $P'_{pq} \ \forall (p, q)$ are then determined by multiplying the shipper-specified priorities P_{pq} with the appropriate factor F_k (where $q \equiv (j, \tau)$ with $j \in \mathcal{P}_k$). The objective function coefficients are then determined via Equations (3.36) - (3.41) using P'_{pq} in lieu of P_{pq}.

3.4   Uncertainty (Robust Optimization) Considerations

In practice, uncertainty always accompanies the data being fed to the model. This uncertainty can be related to the supplies, demands, or most importantly, to the transit times. The source of this uncertainty could be of natural cause or man-made. As far as the uncertainty in supply and demand data is considered, we assume that conservative estimates (under and over, respectively, for supplies and demands) are derived so that there is a reasonable likelihood that there will exist at least the specified level of supply and no more than the specified level of demand. Following normal business practices, these estimates are derived on a daily basis, and are naturally quite reliable for imminent periods, and less so for later periods. This conforms well with our prioritized hierarchy of objective functions for which earlier period decisions and effects are highlighted, while later period decisions and effects are damped out and are less consequential with respect to imminent decisions.

On the other hand, transit times determine the timeliness of deliveries, and a great deal of emphasis has been placed on this feature. In the presence of a high level of uncertainty in transit times, the solution to our tactical model is of questionable validity if we simply use expected transit times in the data. In the realm of mathematical programming, the model could incorporate these noisy data either reactively or proactively. The former usually involves a sensitivity analysis that is conducted after a solution is found. The latter typically uses the technique of stochastic linear programming which tends to inflate the problem size drastically. In our case, an alternative proactive method is suggested. Although the context here is different, this approach is similar in spirit to that of *robust optimization* as introduced by Mulvey et al. (1995) to ease the handling of uncertainty with respect to the growth of the problem size in the context of two-stage optimization with recourse. With this technique, the obtained solution inherently becomes less sensitive to the noisy data. In this section, we derive our objective cost coefficients by incorporating uncertainty, treating not just the expected value of the transit times, but also directly addressing its variability through the use of the established hierarchical penalty structure.

In the present situation, for each arc $(it, j\tau) \in A$, let us assume that we have a histogram based on historical data that gives us different possible (integral) transit times t^s_{ij}, along with

associated probabilities p_{ij}^s, for various scenarios s. Note that we assume that these transit time distributions are invariant over the model horizon, but the distributions used in each model run could depend on the time of the year, or even on present weather conditions. Also, the railroad industry has now compiled such histograms based on historical data. Naturally, it is imperative to update these histograms as changes in practice affect the performance of empty car movements. The cost that we would like to ascribe to the arc $(it, j\tau)$ is given by

$$C_{it,j\tau} = \sum_s p_{ij}^s[t_{ij}^s + \mathcal{M}^s(it, j\tau)] \tag{3.61}$$

where for each scenario s, in the overall weighted sum of the multiple objective functions, the term t_{ij}^s in (3.61) corresponds to the empty car-days component of the objective function, with

$$c_{pq} \equiv c_{it,j\tau} \equiv \sum_s p_{ij}^s t_{ij}^s, \tag{3.62}$$

and the term $\mathcal{M}^s(it, j\tau)$ corresponds to the penalty component in the objective function that is related to the priority and level of this arc as described in Sections 3.1 and 3.2.

Hence, in the multiple objective scheme, different fractions p_{ij}^s of the flow on any arc $(it, j\tau)$ are effectively ascribed to the different objective functions, dependent on the lateness of the flow on this arc under scenario s. Rewriting (3.61), we get

$$C_{it,j\tau} = c_{it,j\tau} + \sum_s p_{ij}^s \mathcal{M}^s(it, j\tau) \tag{3.63}$$

where, as before, $c_{pq} \equiv c_{it,j\tau}$ is the expected transit time from origin i to destination j. In the second term in (3.63), we note that $\mathcal{M}^s(it, j\tau) \equiv 0$ if this scenario corresponds to an on- time (or early) delivery, and the value of $\mathcal{M}^s(it, j\tau)$ changes only according to the lateness of the delivery, given the demand location priority. Hence, the cost ascribed to each route is equal to its expected transit times plus a sum of terms of the type given by a possible lateness penalty times the probability of having that degree of lateness. Therefore, whenever the variability holds the possibility of producing a significant extent of lateness, the corresponding cost reflects the undesirability of using such a route. Note that in the context of our color coding scheme, routes are now viewed as being "multicolored" depending on the probabilities of the various lateness scenarios.

Since typically, a lateness of greater than three days is unacceptable, and since the handling of multiple scenarios for each route can get prohibitive for large-scale problems, we construct (up to) three relevant lateness scenarios as follows. Let us define

$$h_{it,j\tau}^1 = \text{minimum}\{h \geq 1 : \text{ the transit time } (\tau - t + h) \text{ has a positive probability}\}. \tag{3.64}$$

Note that if $h_{it,j\tau}^1$ exists, then this is the least possible *lateness* that can occur on this arc. Hence, this is the least lateness scenario that has a positive associated probability and for which a corresponding penalty needs to be computed in (3.63). If $h_{it,j\tau}^1$ does not exist in (3.64), then set $(h_{it,j\tau}^s, \overline{p}_{it,j\tau}^s) \equiv (0, 0) \ \forall s = 1, 2,$ and 3. Otherwise, let the corresponding

probability associated with (3.64) be $\overline{p}^1_{it,j\tau}$. In addition, there might be a maximum number h_{max} of days of lateness that should be permitted, and for which it is reasonable to construct a corresponding arc. (Note that the use of such a value h_{max} is important from a practical viewpoint, as well as to keep the size of the problem manageable.) Accordingly, if $h^1_{it,j\tau} > h_{max}$, we do not construct such an arc in the model representation. (Also, see Section 3.4.1 below for an alternative decision rule on constructing arcs.)

Next let

$$h^2_{it,j\tau} = (h^1_{it,j\tau} + 1)$$

and

$$\overline{p}^2_{it,j\tau} = \text{corresponding probability of having a transit time of } \tau - t + h^2_{it,j\tau}. \tag{3.65}$$

$$\text{If } \overline{p}^2_{it,j\tau} = 0, \text{then set } (h^2_{it,j\tau}, \overline{p}^2_{it,j\tau}) \equiv (0,0). \tag{3.66}$$

Now, for the third scenario, we aggregate all additional lateness information. To accomplish this, define

$$\overline{p}^3_{it,j\tau} = \text{probability of having a transit time } \geq \tau - t + (h^2_{it,j\tau} + 1). \tag{3.67}$$

$$\text{If } \overline{p}^3_{it,j\tau} = 0, \text{ then set } (h^3_{it,j\tau}, \overline{p}^3_{it,j\tau}) \equiv (0,0). \tag{3.68}$$

Otherwise, compute $h^3_{it,j\tau}$ as the rounded up value ($\lceil \cdot \rceil$) of the *conditional* expected lateness, given that the transit time is $\geq \tau - t + (h^2_{it,j\tau} + 1)$. This is given by

$$h^3_{it,j\tau} = \left\lceil \sum_{T \geq (\tau - t + h^2_{it,j\tau} + 1)} [t + T - \tau] \cdot P(\text{Transit time } = T/\text{transit time } \geq \tau - t + h^2_{it,j\tau} + 1) \right\rceil. \tag{3.69}$$

Hence,

$$h^3_{it,j\tau} = \left\lceil (t - \tau) + \{\text{Conditional expected transit time, given that it is } \geq \tau - t + h^2_{it,j\tau} + 1\} \right\rceil. \tag{3.70}$$

We now use these 3 two-tuples $(h^s_{it,j\tau}, \overline{p}^s_{it,j\tau}), s = 1, 2, 3$, in order to evaluate the second term in (3.63), and hence compute the arc costs as

$$C_{it,j\tau} = c_{it,j\tau} + \sum_{s=1}^{3} \overline{p}^s_{it,j\tau} \mathcal{M}^s(it, j\tau) \quad \forall (it, j\tau) \in A. \tag{3.71}$$

Note that the conditional expectation to be used in (3.70) can be computed once for each histogram corresponding to each origin-destination pair (i, j) and be stored for use in the model, as shown in the example given below.
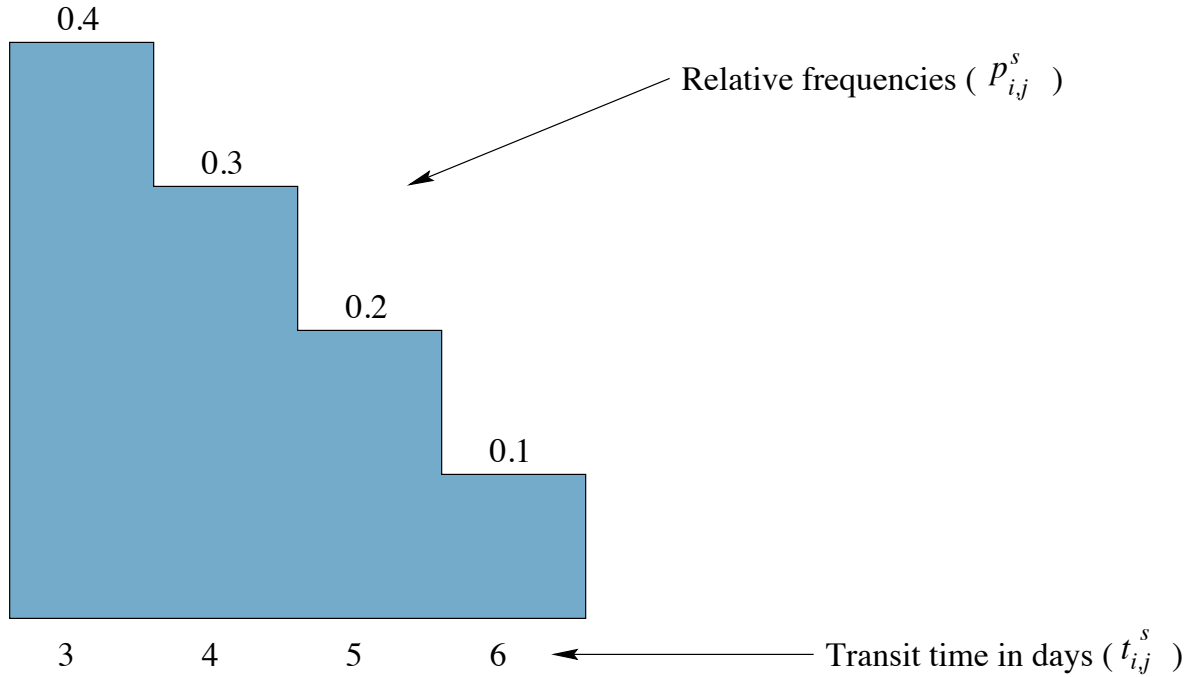
Figure 3.3: Histogram for Example 3.4.1.

Example 3.4.1.

To illustrate, consider the following example. The histogram for the transit times between an origin i and a destination j is shown in Figure 3.3.

For this, we record the following information.

Scenario index s	Transit time t_{ij}^s	Probability p_{ij}^s	$\leq$ Cumulative probability	$\geq$ Cumulative probability	Conditional expected transit time, given that it is $\geq t_{ij}^s$
1	3	0.4	0.4	1.0	$c_{it,j\tau} = 4.0$
2	4	0.3	0.7	0.6	$\frac{4(0.3)+5(0.2)+6(0.1)}{0.6} = 4.67$
3	5	0.2	0.9	0.3	$\frac{5(0.2)+6(0.1)}{0.3} = 5.33$
4	6	0.1	1.0	0.1	6

Accordingly, Table 3.9 gives the values for $(h_{it,j\tau}^s, \overline{p}_{it,j\tau}^s)$, for $s = 1$, 2, and 3, corresponding to various scenarios identified by $(\tau - t)$ and a maximum permissible lateness of $h_{max} = 3$ days.

Remark 3.1. Note that in general, we will need to compute the conditional expectation for use in (3.70) for transit times that span from the third possible ordered transit time value up to the second last value. Since we had just four possible transit times in the above example,

we needed the conditional expectation corresponding to only the third value of $t_{ij} = 5$ in this case.

Table 3.9: Table of (Lateness,Probability) Two-tuples for Various Cases for Example 3.4.1.

Case of $\tau - t$	$(h_{it,j\tau}^1, \overline{p}_{it,j\tau}^1)$	$(h_{it,j\tau}^2, \overline{p}_{it,j\tau}^2)$	$(h_{it,j\tau}^3, \overline{p}_{it,j\tau}^3)$
≥ 6	(0, 0)	(0, 0)	(0, 0)
5	(1, 0.1)	(0, 0)	(0, 0)
4	(1, 0.2)	(2, 0.1)	(0, 0)
3	(1, 0.3)	(2, 0.2)	(3, 0.1)
2	(1, 0.4)	(2, 0.3)	(4, 0.3)
			where $\lceil -2 + 5.33 \rceil = 4$
1	(2, 0.4)	(3, 0.3)	(5, 0.3)
			where $\lceil -1 + 5.33 \rceil = 5$
0	(3, 0.4)	(4, 0.3)	(6, 0.3)
			where $\lceil 5.33 \rceil = 6$
$\leq$ -1	Arc should not be generated (for $h_{max} \equiv 3$).		

3.4.1 Chance-constraints for Deciding on Constructing Arcs

In general, we can use the maximum lateness parameter h_{max} in deciding whether or not a particular arc should be constructed. In particular, the recommendation made thus far is that if

$$h_{it,j\tau}^1 > h_{max}, \text{ then do not construct } (it, j\tau) \text{ for inclusion in } A. \tag{3.72}$$

Note that from (3.64), we have

$$h_{it,j\tau}^1 > h_{max} \Leftrightarrow (\tau - t) \leq \text{ minimum}_s\{t_{i,j}^s\} - h_{max} - 1. \tag{3.73}$$

Hence, for the previous example, given $h_{max} = 3$ and minimum$_s\{t_{i,j}^s\} = 3$, we do not include arcs in the problem whenever $(\tau - t) \leq -1$.

Note that in the foregoing example, when $(\tau - t) \leq -1$, the probability of having a lateness of three days or less is zero. An alternative strategy that is more reflective of the stochasticity in the problem is to use a chance constraint of the type

$$P(\text{lateness } \leq 3) \geq p_{acc}. \tag{3.74}$$

Given that a three day late delivery is considered as being on the threshold of unacceptable service based on planned queue-size buffers, constraint (3.74) requires that the probability

of having lateness contained within this limit should at least be at some acceptable service probability level p_{acc} for an arc to be considered for construction. Constraint (3.74) can be rewritten as follows :

$$P(t + t_{i,j}^s - \tau \le 3) \ge p_{acc}$$
$$\text{or} \quad P(t_{i,j}^s \le (3 + \tau - t)) \ge p_{acc}. \tag{3.75}$$

Defining $t_{i,j}^{acc}$ to be the *minimum transit time* for which

$$P(t_{i,j}^s \le t_{i,j}^{acc}) \ge p_{acc}, \tag{3.76}$$

we have that (3.75) is equivalent to

$$(3 + \tau - t) \ge t_{i,j}^{acc} \text{ or } (\tau - t) \ge (t_{i,j}^{acc} - 3). \tag{3.77}$$

Hence, analogous to (3.73), if

$$(\tau - t) \le (t_{i,j}^{acc} - 4) \Rightarrow \text{ do not construct } (it, j\tau) \text{ for inclusion in } A. \tag{3.78}$$

Using the above example, for $p_{acc} = 0.5$, we have from (3.76) that $t_{i,j}^{acc} = 4$ days, and so from (3.77), we would not construct this arc whenever $(\tau - t) \le 0$. Based on (3.77), we would have the decision rule shown in Table 3.10. Note that the decision rules (3.73) and (3.77) match in this example whenever $\text{minimum}_s\{t_{i,j}^s\} - h_{max} - 1 = t_{i,j}^{acc} - 4$, i.e., whenever

$$h_{max} \equiv h_{max}^{equiv} = \text{ minimum}_s\{t_{i,j}^s\} + 3 - t_{i,j}^{acc} = 6 - t_{i,j}^{acc}. \tag{3.79}$$

The corresponding equivalent values of h_{max}, denoted h_{max}^{equiv}, are computed in Table 3.10.

Table 3.10: Chance-constraint Example Computations.

p_{acc}	$t_{i,j}^{acc}$	Do not construct arc when $(\tau - t)$ is $\le$ following :	h_{max}^{equiv}
(0.9, 1]	6	2	0
(0.7, 0.9]	5	1	1
(0.4, 0.7]	4	0	2
(0, 0.4]	3	-1	3

Based on results observed from some real-data histograms, we recommend the use of $p_{acc} = 0.5$. As indicated above, for the foregoing example, we would therefore not construct an arc whenever $(\tau - t) \le 0$. In terms of h_{max}, this would be equivalent to stating that with the quickest possible transit time of 3 days, if the lateness will be $> h_{max}^{equiv} = 2$ days late (i.e. ≥ 3 days late), then we should not consider such an arc in the model. From a conservative viewpoint, since smaller values of p_{acc} result simply in more arcs being considered in the model, it might be better to err on the side of selecting p_{acc} too small, rather than having it too large and thereby excluding too many routes from consideration. With this same motivation, relatively larger values of p_{acc} can be used for later days in the horizon than for more imminent periods.

3.4.2 Summary of Approach

Drawing a conclusion to this section, we note that uncertainty could be incorporated within the model via a preprocessing of the data. This process includes arc exclusion steps based on a chance-constraint or a maximum lateness specification, and includes the computation of the defined three scenario latenesses along with their associated probabilities. Once these numerical values are ascertained, they could be used in the cost penalty recomputation described in Sections 3.2 and 3.3.

Other uncertain data pertaining to supplies and demands are handled via suitable estimates as mentioned above. Also, as stated before, we need to be concerned about obtaining accurate estimates mainly for the earlier time periods. Besides being inherently relatively more fuzzy, the later demands or supplies have a damped effect on the imminent period decisions by virtue of our time priority and network penalty structure.

3.5 Summary of Overall Model Construction

In this section, we provide a brief summary of the input data requirements and the steps for constructing the model. This summary also serves to provide a glossary of notation.

3.5.1 Glossary of Notation and Input Requirements

m_{loc} = the total number of supply locations.

n_{loc} = the total number of demand locations.

t_{min} = the imminent period for which active demand satisfaction decisions are under considerations (usually, one or two days from the present time).

t_{max} = the maximum time-period used in the "look-ahead" model horizon.

D = number of days (possibly fractional) of future demand that determines the queue size at each demand location.

AD_{τ} = actual demand for empty cars at time period τ for each demand location.

$OH_{t_{min}}$ = number of empty cars available on-hand at the initial time period t_{min} for each demand location.

SA_{τ} = number of empty cars that have already been pre-scheduled (extraneous to the present model decisions) to arrive at time period τ for each demand location.

m = the total number of supply nodes in the time-space transportation network.

n = the total number of demand nodes in the time-space transportation network.

A = the set of real arcs that connect supply with demand nodes in the time-

	space transportation network.
p	$\equiv (i,t)$ = a particular supply node that represents the supply location i, at time t, $p = 1, \ldots, m$.
q	$\equiv (j,\tau)$ = a particular demand node that represents the demand location j, at time t, $q = 1, \ldots, n$.
s_p	= the supply of empty railcars at a supply node p.
d_q	= the demand of empty railcars at a demand node q.
White	= the set of arcs that represent the slacks in the supply constraints.
Green	= the set of on-time arcs.
Pink	= the set of 1-day late arcs.
Red	= the set of 2-days late arcs.
Gray	= the set of 3-or-more days late arcs.
Black	= the set of arcs that represents unmet demands.
DS	= the dummy supply node in the transportation problem.
DD	= the dummy demand node in the transportation problem.
x_{pq}	= the flow on arc (p,q). This is a *decision variable*.
c_{pq}	= the (original) cost of arc (p,q) that represents the expected transit time. (This is also designated as t_{ij} where $p \equiv (i,t)$ and $q \equiv (j,\tau)$ for some t and τ.
c_{max}	= maximum (original) cost over all *green, pink, red*, and *gray* arcs in the problem.
c_{min}	= minimum (original) cost over all *green, pink, red*, and *gray* arcs in the problem.
l_{pq}	= the lower bound on flow for arc (p,q).
u_{pq}	= the upper bound on flow for arc (p,q).
PRMAT	= the time-based priority matrix.
W	= the minimum of $\{m, n, |A|/2\}$
$\mathcal{M}$	= the base penalty to be ascribed to the unmet demand arcs (given by Equation (3.26)).
V	= the largest integer value permissible for use in the program.
l	= the level of arcs.
l_{max}	= the maximum level in the time-based priority hierarchy = $4 \cdot (t_{max} - t_{min} + 1)$.
MC	= the maximum cost in the problem (given by Equation (3.29)).
θ	= the parameter governing the spread of the penalty ranges.
maxgray	= maximum lateness value for the gray arcs at level l.
d_l	= the shifted maximum lateness value for the gray arcs at level l.
LAT_{pq}	= the specified lateness corresponding to a gray arcs (p,q).
a_l	= the lower bound of the level l's penalty range (given by Equation (3.31)).
b_l	= the upper bound of the level l's penalty range (given by Equation (3.31)).
C_{pq}	= the modified penalized cost of arc (p,q).

K = the total number of shippers.

k = the shipper's index $(k = 1, \ldots, K)$.

T = the duration over which equity is being sought for the present model run.

Q_k = the number of loaded plus empty car-days over the prior time segment of duration T.

A_k = the subset of arcs A that correspond to shipper k.

FS_k = the assigned fleet size for shipper k.

F_k = the demand location priority scaling factor for shipper k.

$\overline{P}_{kj}$ = the shipper designated priority for demand location j (also, see P_{pq}).

$\mathcal{P}_k$ = the set of demand locations associated with shipper k.

P_{pq} = the demand location priority for a given arc (p, q) (also, see $\overline{P}_{kj}$).

P'_{pq} = the scaled demand location priority for a given arc (p, q).

AP_k = the average weighted plant priority based on $\overline{P}_{kj}$.

R_k = the compensation in the car-days Q_k to make $Q_k + R_k$ directly proportional to $(T)(FS_k)$.

R = the range of the spread of the R_k values.

R_{max} = $\max_k\{R_k\}$.

R_{min} = $\min_k\{R_k\}$.

λ = an equity-scaling parameter that represents the ratio of the range of the scaled average weighted priorities to their maximum values (recommended value is $\lambda = 0.5$).

$\overline{R}_k$ = the convex combination weight for R.

U = the upper limit of the range for the scaled average weighted priorities.

L = the lower limit of the range for the scaled average weighted priorities.

t_{ij}^s = the transit time under a scenario s between supply location i and demand location j.

p_{ij}^s = the probability of the transit time between supply location i and demand location j being equal to t_{ij}^s.

t_{ij} = expected transit time = $\sum_s p_{ij}^s t_{ij}^s$

$\mathcal{M}^s(it, j\tau)$ = the penalty component in the objective function that is related to the priority and level of arc $(it, j\tau)$.

$h_{it,j\tau}^1$ = minimum$\{h \geq 1 :$ the transit time $(\tau - t + h)$ has a positive probability$\}$.

$\overline{p}_{it,j\tau}^1 \cdot$ = the corresponding probability of $h_{it,j\tau}^1$ if it exists.

$h_{it,j\tau}^2$ = $(h_{it,j\tau}^1 + 1)$

$\overline{p}_{it,j\tau}^2$ = corresponding probability of having a transit time of $\tau - t + h_{it,j\tau}^2$.

$h_{it,j\tau}^3$ = the rounded-up value of the conditional expected lateness, given that the transit time is $\geq \tau - t + (h_{it,j\tau}^2 + 1)$.

$\overline{p}_{it,j\tau}^3$ = probability of having a transit time $\geq \tau - t + (h_{it,j\tau}^2 + 1)$.

h_{max} = a maximum number of days of lateness that should be permitted, and for which it is reasonable to construct a corresponding arc.

p_{acc} = an acceptable service probability level for an arc to be considered for construction.

$t_{i,j}^{acc}$ = the *minimum transit time* for which $P(t_{i,j}^s \leq t_{i,j}^{acc}) \geq p_{acc}$.

h_{max}^{equiv} = the equivalent value of h_{max} corresponding to the specified chance-constraint.

3.5.2   Steps for Constructing the Model

- For each demand location, compute the required queue size, RQ_τ, via Equation (3.8) for each period τ.

- For each demand location, compute the demand input at time period τ, $MD_\tau (\equiv d_{j\tau}$ for location j), for all $\tau \geq t_{min}$, via Equations (3.9), (3.10) and (3.11).

- Treat weekend demands via either of the following two strategies :

  1. Do nothing.
  2. Modify the variable queue size requirement RQ_τ to RQ_τ' where RQ_τ' is defined as in Subsection 3.1.2.

- Eliminate demand nodes having zero demands. (This is desirable, but not necessary.)

- Compute the three scenario lateness and probability two-tuples $(h_{it,j\tau}^s, \overline{p}_{it,j\tau}^s)$ for $s = 1$, 2, 3, via Equations (3.64) - (3.70).

- Determine the arcs to be included in the model using h_{max} and/or p_{acc} via Equations (3.72) and (3.75).

- Compute the scaling factors $F_k, k = 1, ..., K$ via Equations (3.47), (3.50), (3.53), (3.55), and (3.56).

- Compute the scaled priorities $P_{pq}' = P_{pq} \cdot F_k$ (where $q \equiv (j, t)$, with $q \in \mathcal{P}_k$) $\forall (p, q) \in A$.

- Compute θ via Equations (3.43) where MC is given by Equation (3.42).

- For each demand time τ, identify all the gray arcs by examining the different considered aggregate scenarios $s = 1$, 2, 3 for the arcs $(it, j\tau) \in A$, and determining those that correspond to a lateness of three or more days. For the corresponding level l, compute $a_l, maxgray$, and d_l using Equations (3.31), (3.32), and (3.33). This information is used in the next computation.

- For each of the three aggregate scenarios $s = 1$, 2, 3, corresponding to each arc $(it, j\tau) \in A$, compute $\mathcal{M}^s(it, j\tau)$ via the second terms of Equations (3.36) - (3.40) depending on whether the scenario s makes the arc *pink, red*, or *gray*, respectively, where l is the level depending on the scenario lateness and the demand time τ, and where P_{pq}' is substituted for P_{pq}. Note that $\mathcal{M}^s(it, j\tau) = 0$ for arcs in the sets *White* and *Green*.

- Compute $c_{it,j\tau}$ via Equation (3.62) and hence compute $C_{it,j\tau}$ via Equation (3.71) for each $(it, j\tau) \in A$.

- Compute $C_{DS,q}$ for black arcs via Equation (3.41) using P'_{pq} in lieu of P_{pq} therein.

- Solve the resulting transportation problem TDSS1 defined as follows.

TDSS1: Minimize $\qquad \displaystyle\sum_{(p,q)\in A}\sum C_{pq}x_{pq} + \sum_q C_{DS,q}x_{DS,q}$

subject to $\qquad \displaystyle\sum_{p:(p,q)\in A} x_{pq} + x_{DS,q} = d_q \ \forall q$

$$\sum_{q:(p,q)\in A} x_{pq} + x_{p,DD} = s_p \ \forall p$$

$$l_{pq} \le x_{pq} \le u_{pq} \forall (p,q) \in A, x_{DS,q} \ge 0 \ \forall q, x_{p,DD} \ge 0 \ \forall p.$$

Chapter 4

Model Under Blocking Considerations (TDSS2)

4.1 Motivation

Recall that the distribution system being developed serves to coordinate the timely distribution of empty autorack railcars that are jointly owned by several automobile manufacturers (shippers) and are operated by a group of participating railroads. Different types of empty cars (mainly bilevels, trilevels, and TTQ-X cars) are transferred from various originating locations at specific points in time to satisfy demands in a timely fashion at various assembly facilities owned by the automobile manufacturers.

Due to the nature of the business and economies of scale, blocking needs to be enforced as it results in a more efficient use of resources. This blocking is intended to avoid splitting the supply of cars at any originating location at any point in time toward more than a specified number of locations, as well as to ensure that each block is at least of a certain minimal size.

In light of the model TDSS1 that we have developed in Chapter 3 thus far, it is apparent that this model lacks the capability of ensuring the above blocking requirement. The network flow-based model, however, can be modified to *encourage* blocking rather than to enforce it. This can be done by manipulating the weights of the arcs in the network to make some arcs more attractive than others, but this requires an *a priori* decision of which destinations should be preferably fed from each origin. Hence, such a strategy lacks rigor and appeal. Therefore, to address this issue, we enhance our model by explicitly incorporating zero-one blocking decisions, and hence derive a mixed integer programming (MIP) model.

We begin our discussion by developing the proposed discrete model TDSS2 as an extension of the original network flow model TDSS1. Further enhancements of the model are performed using the Reformulation-Linearization Technique (RLT) of Sherali and Adams (1990, 1994)

to construct partial convex hull representations in order to tighten the continuous relaxation of the basic model. Following this, we design several heuristic procedures that permit us to obtain good quality solutions within a reasonable amount of time in order to facilitate a real-time interactive use of the model.

For convenience, the network-flow model of Chapter 3 is rewritten as follows.

$$\textbf{TDSS1: Minimize} \quad \sum_{(p,q)\ \in A} \sum \left(c_{pq} + \sum_{s=1}^{3} \overline{p}_{pq}^{s} \mathcal{M}^{s}(p,q)\right) x_{pq} + \sum_{q} \mathcal{M}(DS,q) x_{DS,q} \quad (4.1)$$

$$\text{subject to} \quad \sum_{p} x_{pq} + x_{DS,q} = d_q \ \forall q \quad (4.2)$$

$$\sum_{q} x_{pq} \leq s_p \ \forall p \quad (4.3)$$

$$l_{pq} \leq x_{pq} \leq u_{pq} \ \forall p, q, \ \text{and} \ x_{DS,q} \geq 0 \ \forall q. \quad (4.4)$$

Here, x_{pq}, where $p \equiv (i,t)$ and $q \equiv (j,\tau)$ denotes the number of empty cars (of some particular type) sent from origin node i at time t to satisfy the demand at location j corresponding to time period τ. This transfer of cars from the origin location i to the destination location j (along the designated route) might take $t_{pq}^{s'}$ time units with probability $p_{pq}^{s'}$ under scenario s', for $s' = 1, \ldots, S$. The value c_{pq} is given by

$$c_{pq} = \sum_{s'=1}^{S} t_{pq}^{s'} p_{pq}^{s'} \ \forall (p,q) \in A.$$

Moreover, $\mathcal{M}^{s}(p,q)$ corresponds to the penalty component in the objective function that is related to the priority and level of the arc (p,q) under the reduced scenario $s \in \{1,2,3\}$, having an associated probability of $\overline{p}_{pq}^{s}$ as described in Chapter 3. The unmet demand penalties $\mathcal{M}(DS, j\tau) \equiv \mathcal{M}(DS,q) \equiv C_{DS,q}$ are similarly defined as in Chapter 3.

Furthermore, in (4.4) of Model **TDSS1**, we have designated implied lower and upper bounds l_{pq} and u_{pq}, respectively, on each arc (p,q). Unless otherwise specified, these lower and upper bounds can be initially set at

$$l_{pq} = 0 \ \text{and} \ u_{pq} = \text{minimum}\{s_p, d_q\} \ \ \forall (p,q) \quad (4.5)$$

and can be further modified based on a logical preprocessing of (4.2) - (4.4). However, as we shall see later, these bounds on flows might be specified to conform with user-defined fixed-flow designations (*control orders*) as well. Note that for each variable x_{pq}, by examining the constraints that this variable appears in, we can perform logical reductions to derive as tight implied flow bounds on this variable as possible. This is explained in more detail below, and serves to enhance the solution capability of the discrete model developed next.

At several geographical locations around the country, there exist groups of neighboring demand locations for which the shipment of empty cars is considered as being sent to a single

super-demand location from the viewpoint of blocking considerations. Such blocks of shipment for each of these location groups might be sent either to a central hub location from where it is split into the component shipments for the various destinations comprising the group according to the specified flow dispositions, or the block might be delivered by depositing empty cars appropriately *en route* while visiting these members of the group in some sequential fashion. In either case, for modeling purposes, we can treat the various routes from origins to such individual demand location destinations as distinct routes, each having its corresponding transit-time cost and lateness penalty. However, from a blocking point of view, a shipment to any combination of such destinations within a group that originates at a source i at time t (i.e., originates at supply node $p \equiv (i, t)$), is considered as a single block. Note that, we cannot simply aggregate demands for the different destinations within a group, because each member in a group might correspond to a different manufacturer and/or have differing specified priorities.

To model this consideration, suppose that there are G such groups of destinations, where J_g denotes the demand locations j that are consolidated within group g, for $g = 1, \ldots, G$. Note that $J_g \cap J_g{'} = \emptyset$ for $g \neq g'$, and that $\cup J_g$ is assumed to be comprised of the entire set of demand locations. Possibly, several "groups" J_g might be singletons. Accordingly, let us define our *blocking binary variables* y as follows.

$$y_{pg} = \begin{cases} 1 & \text{if } x_{it,j\tau} > 0 \text{ for } p \equiv (i, t) \text{ and for any } (j, \tau), \text{ where } j \in J_g \\ 0 & \text{otherwise} \end{cases} \quad \forall (p, g). \quad (4.6)$$

Furthermore, for each destination location j, let

$$g(j) = \text{group index such that } j \in J_{g(j)}, \quad (4.7)$$

and define

$$J(g) = \{q \equiv (j, \tau) : j \in J_g\}. \quad (4.8)$$

For ease in notation, for any $q \equiv (j, \tau)$, we will refer to the corresponding group $g(j)$ also as $g(q)$ without any confusion.

The principal blocking relationship desired by the railroad and automobile industries is to restrict the supply at any origin location i for a given time period t to be split for shipment toward no more than some n_p demand locations. (Usually, $n_p = 1$, but sometimes, this can be 2 or 3, and it might possibly be time-dependent as well.) Furthermore, there might be a minimum block size requirement whenever any block is formed. In order to incorporate this feature, let us define the minimal block size L_{pg} for a group g to which shipment is sent from the origin node p as

$$L_{pg} = \text{maximum}\{l_{pg}^{\min}, \sum_{q \in J(g)} l_{pq}\} \ \forall p, g, \quad (4.9)$$

where l_{pq} is the specified lower bound on the flow x_{pq} for arc $(p, q) \in A$, and where

$$l_{pg}^{\min} = \text{minimum allowable size of a block that can be formed at origin node p for shipment to destination group } g.$$

We observe here that we require

$$1 \leq l_{pg}^{\min} \leq \text{minimum}\{s_p, \sum_{q \in J(g)} d_q\} \; \forall p, g. \tag{4.10}$$

Also, if there is no restriction on the size of the block used and we only need to limit the number of blocks n_p that are formed at any source p, then we can set $l_{pg}^{\min} \equiv 1$. Likewise, we can derive an upper bound U_{pg} on the total flow from source $p \equiv (i, t)$ to group g. This value can be taken as

$$U_{pg} = \text{minimum}\{s_p, \sum_{q \in J(g)} \max\{d_q, l_{pq}\}\} \; \forall p, g. \tag{4.11}$$

The formula for U_{pg} in (4.11) needs some clarification. Because of blocking considerations, it might be that the lower bound l_{pq} specified on some arc (p, q) exceeds d_q, in which case, there will be an excess delivery to node q. However, the flow to node q along (p, q) would never exceed $\max\{d_q, l_{pq}\}$. Noting that the total flow out of p cannot exceed s_p, we derive Equation (4.11).

Accordingly, as a generalization of (4.5), we assume that

$$0 \leq l_{pq} \leq s_p \text{ and that } u_{pq} = \min\{s_p, \max\{l_{pq}, d_q\}\} \; \forall (p, q). \tag{4.12}$$

Naturally, we must also have

$$\sum_q l_{pq} \leq s_p, \; \forall p. \tag{4.13}$$

In (4.12), note that

$$u_{pq} = \min\{s_p, d_q\} \text{ if } l_{pq} \leq d_q, \text{ and } u_{pq} = l_{pq} \text{ if } l_{pq} \geq d_q. \tag{4.14}$$

The specified data is checked in a preprocessing step to ensure that the foregoing relationships hold true.

With these definitions, we construct the following modification of TDSS1, where C_{pq} is the coefficient of x_{pq} in (4.1), A is the set of real arcs (p, q) in this model, and where the constraint $y \in Y$ in (4.21) below represents some special restrictions on the y-variables as discussed in the sequel.

$$\textbf{TDSS2: Minimize} \quad \sum_{(p,q) \in A} \sum C_{pq} x_{pq} + \sum_q \mathcal{M}(DS, q) x_{DS,q} \tag{4.15}$$

$$\text{subject to} \quad \sum_{p:(p,q) \in A} x_{pq} + x_{DS,q} \geq d_q \; \forall q \tag{4.16}$$

$$\sum_{q:(p,q) \in A} x_{pq} \leq s_p \; \forall p \tag{4.17}$$

$$\sum_{g=1}^{G} y_{pg} \leq n_p \ \forall p \tag{4.18}$$

$$l_{pq} \leq x_{pq} \leq u_{pq} y_{p,g(q)} \ \forall (p,q) \in A \tag{4.19}$$

$$L_{pg} y_{pg} \leq \sum_{q \in J(g)} x_{pq} \leq U_{pg} y_{pg} \ \forall p, g = 1, \ldots, G \tag{4.20}$$

$$y \in Y \tag{4.21}$$

$$y \text{ binary.} \tag{4.22}$$

Observe that in the model TDSS2, we have restricted the blocking variables y to be binary valued, but we have permitted the flows to be simply continuous variables. The following result establishes the fact that given feasibility for TDSS2, there exists an optimal solution at which the flow decisions x are integer valued. Hence, the integrality of the flows x is an automatic consequence of solving TDSS2, provided that the solution procedure employed determines an optimal *extreme point* completion x^*, given a set of binary optimal blocking decisions y^*.

Theorem 4.1 Consider Problem TDSS2, and for any $y \in Y, y$ binary, such that (4.18) is satisfied, let $X(y)$ denote the feasible region of TDSS2 in the variables x with y fixed as specified. If $X(y) \neq \emptyset$, it has integral extreme points, and so, Problem TDSS2 has an optimum (if one exists) at which x automatically takes on integral values.
Proof. Define the variables

$$z_{pg} = \sum_{q \in J(g)} x_{pq} \quad \forall \ p, g = 1, \ldots, G. \tag{4.23}$$

Then, we can equivalently express the constraints of $X(y)$ as follows

$$\sum_{p} x_{pq} + x_{DS,q} \geq d_q \ \ \forall q \tag{4.24}$$

$$- \sum_{q \in J(g)} x_{pq} + z_{pg} = 0, \ \ \forall p, g \tag{4.25}$$

$$\sum_{g} z_{pg} \leq s_p \ \ \forall p \tag{4.26}$$

$$l_{pq} \leq x_{pq} \leq u_{pq} y_{p,g(q)} \ \ \forall (p,q) \in A \tag{4.27}$$

$$L_{pg} y_{pg} \leq z_{pg} \leq U_{pg} y_{pg} \ \ \forall p, g. \tag{4.28}$$

When written in the form (4.24)-(4.28), we see that $X(y)$ in the variables (x, z), where z is related to x via the identities (4.25), represents network flow constraints in bounded variables, with each variable x_{pg} and z_{pg} having a structural column comprised of a +1, a -1, and zeros otherwise. (See Bazaraa, Jarvis, and Sherali (1990); Figure 4.1 depicts this network.) Hence, if feasible, $X(y)$ has integral extreme points. Therefore, if Problem TDSS2 is feasible, it has an optimum y^*, and fixing $y = y^*$, the corresponding extreme point optimum to the resultant problem over $X(y^*)$ would then yield an integral flow solution x^*. This completes the proof.$\square$
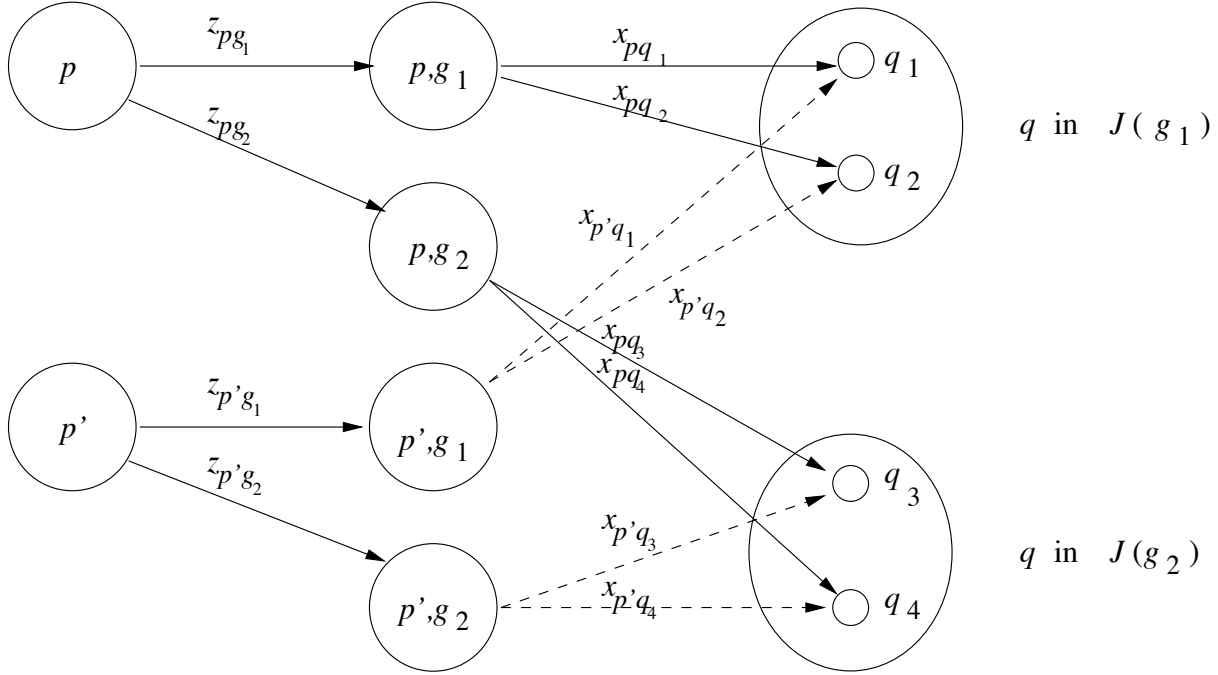
Figure 4.1: Illustration of Network of Theorem 4.1.

Remark 4.1. Observe that we could obtain an identical model (of similar strength with respect to the LP relaxation as well) by using s_p in place of U_{pg} in (4.20). This follows from the fact that whenever $s_p > \sum_{q \in J(g)} \max\{d_q, l_{pq}\}$ in (4.11), we have $u_{pq} = \max\{d_q, l_{pq}\}$ $\forall q \in J(g)$ from (4.12), and therefore (4.19) implies that

$$\sum_{q \in J(g)} x_{pq} \leq \sum_{q \in J(g)} u_{pq} y_{p,g(q)} = y_{pg} \sum_{q \in J(g)} \max\{d_q, l_{pq}\} = U_{pg} y_{pg}. \tag{4.29}$$

Hence, the right-hand inequality in (4.20) is then implied. However, since it is possible that $U_{pg} < s_p$, we will find it beneficial from the viewpoint of performing logical reductions as well as for possible future extensions to use the tighter bound U_{pg} in (4.20) as stated. □

Remark 4.2. Note that for each (p, q), if $l_{pq} > 0$ then we can fix $y_{pg(q)} = 1$ by virtue of (4.19). Note also that if at all blocking is performed from origin location i at time t (i.e. from source $p \equiv (i, t)$) to create a shipment for demand group g, i.e., if $y_{pg} = 1$, then the total number of cars sent toward the locations in J_g with the intent of satisfying demands in various periods τ, should be at least $l_{pg}^{\min}$, and by (4.19), also at least $\sum_{q \in J(g)} l_{pq}$, and hence by (4.9), at least L_{pg}. Similarly, this total should not exceed s_p or $\sum_{q \in J(g)} \max\{d_q, l_{pq}\}$ as discussed above, and hence be bounded above by U_{pg} as defined in (4.11). Note by (4.9), (4.10), (4.12) and (4.13), that $L_{pg} \leq U_{pg}$ $\forall p, g$. This yields constraint (4.20). Observe that in order to preserve feasibility of this model under such minimal block-size constraints, it becomes necessary to permit the demand at any node $q \equiv (j, \tau)$ to be satisfied possibly in excess of d_q as in (4.16). This would mean that the queue at this location j might increase, thereby requiring

a corresponding adjustment in future demands at this location. (This would not be the case if $l_{pg}^{\min} \equiv 1$, $\forall p, g$ and $\sum_p l_{pq} \leq d_q$ $\forall q$.) Furthermore, note that blocking stipulations could result in excess cars remaining unshipped at originating locations. If desired, these residual cars could be redistributed in a subsequent run of the model by appropriately fixing certain designated decisions based on the previous run or by imposing suitable penalties on this residual slack flow. Such fixed flow designations might also force more railcars toward a sink node q than the original demand d_q, as indicated above, thereby resulting in an *overmet demand* situation.□

Remark 4.3. As an alternative blocking formulation strategy, we could make n_p in (4.18) a bounded integer variable, and penalize it if it exceeds a permissible threshold amount. For example, suppose that we would like to use no more than two blocks from node p if possible, but we permit up to four blocks to be created if necessary, with an appropriate penalty imposed for the extra blocks. Hence, we can let n_p be a variable in (4.18), and impose the additional constraints

$$n_p \quad = \quad 2 + z_{p1} + 2z_{p2}, \tag{4.30}$$
$$\text{where } z_{p1} + z_{p2} \leq 1, \quad z_{p1} \text{ and } z_{p2} \text{ binary}, \tag{4.31}$$

and penalize z_{p1} and z_{p2} in the objective function, depending on the use of one or two excess blocks, respectively. For now, we will consider n_p as a given parameter in (4.18), and will leave the option of using the approach of (4.30) - (4.31) as a future consideration.□

Remark 4.4. The constraints $y \in Y$ incorporate special conditions on the binary variables that need to be incorporated within the model. These include restrictions corresponding to fixing certain variables at 0 or 1 values, or equating some binary variables to each other. For example, consider a situation in which a location i can send cars to location j or k. Furthermore, suppose that location j has a demand for at least 20 cars of type BA (a particular type of bilevel) out of a total demand of 100 cars, and location i has a supply of 30 cars of type BA. (The time aspect is suppressed in this discussion for clarity). Then, the demand location (or node) j can be split into two nodes j_1 and j_2 having respective demands of 20 and 80, with supplies, demand, and 0-1 variable designations as depicted in Figure 4.2.

Note that the 0-1 variables on the arcs (i, j_1) and (i, j_2) have been equated to the single variable y_1, since these arcs jointly represent a shipment between the same origin and destination locations at a given time. This is similar to using the same binary variable $y_{it,j}$ to represent the activation or "blocking" of flow to a particular demand group, which would be an alternative way of modeling this situation. All such restrictions that are explicitly enforced separately (perhaps for the purpose of conducting subsequent what-if analyses) compose the constraints represented in $y \in Y$. Of course, these types of substitutions would be filtered into the implemented model. □
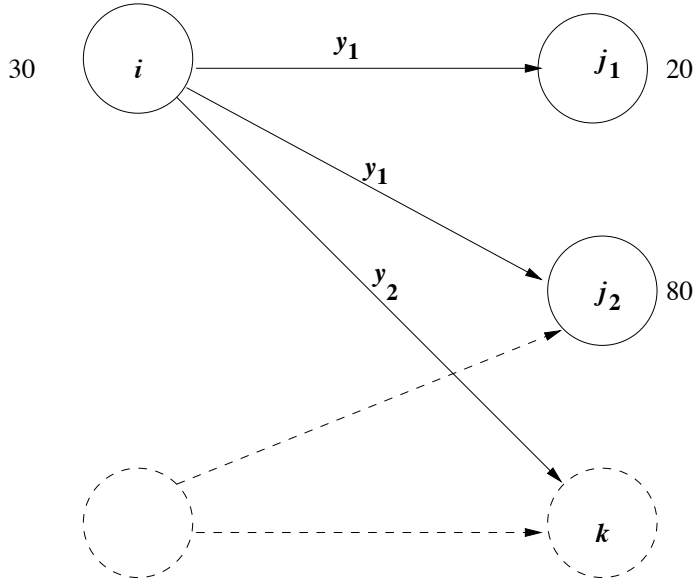
Figure 4.2: Illustration of a Special Blocking Scheme.

4.2 RLT-based Partial Convex Hull Representations

The reader might have observed that in the model TDSS2, we have included an upper bounding constraint in (4.20) that is redundant in the *integer sense* (i.e. when y is binary valued) as demonstrated in Remark 4.1. This follows because for each p, g, if $y_{pg} = 0$, then $x_{pg} = 0 \ \forall q \in J(g)$ from (4.19), and so the right-hand inequality in (4.20) holds true. Similarly, when $y_{pg} = 1$, if $U_{pg} = s_p$, we have from (4.17) that $\sum_{q \in J(g)} x_{pq} \leq s_p \equiv U_{pg}$, and if U_{pg} is given by the second term is the minimand in (4.13), then as in (4.29) of Remark 4.1, we again have that the right-hand inequality in (4.20) is implied. In this section, we first reveal the purpose of this constraint by demonstrating that it tightens the continuous relaxation by actually constructing a partial convex hull representation. This significantly enhances the solution capability of the model as we shall see. Further model enhancements are also discussed but its implementation is left for future consideration.

To elucidate this partial convex hull property, consider the example network of Figure 4.3 that represents the essence of the structure in TDSS2, where the flows x are restricted to be nonnegative and have upper bounds implied by the supplies and the demands (see Equation (4.14)).

Let us state the associated constraints of TDSS2 as follows, except that, let us suppress the upper bounding constraints in (4.20) that are redundant in the integer sense as mentioned above.
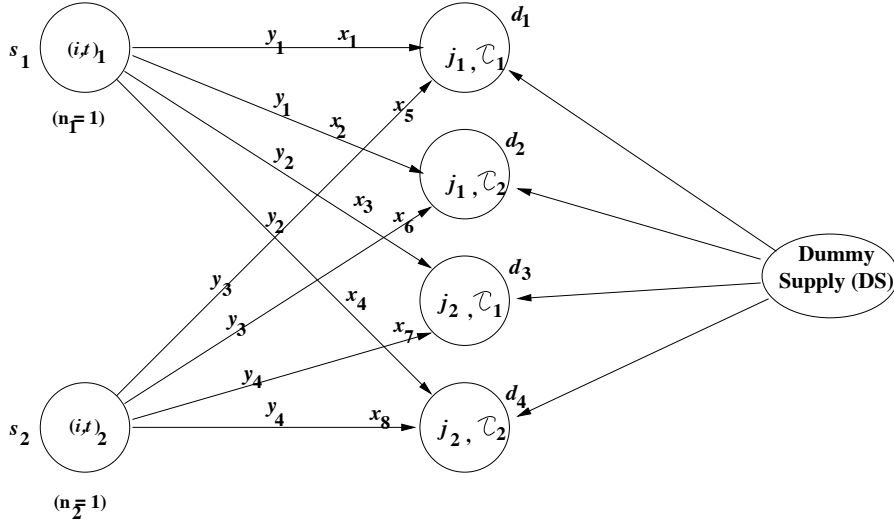
Figure 4.3: Illustration of the TDSS2 Network Structure

$$x_1 + x_2 + x_3 + x_4 \leq s_1 \tag{4.32}$$
$$x_5 + x_6 + x_7 + x_8 \leq s_2 \tag{4.33}$$
$$x_1 + x_5 + x_{DS1} \geq d_1 \tag{4.34}$$
$$x_2 + x_6 + x_{DS2} \geq d_2 \tag{4.35}$$
$$x_3 + x_7 + x_{DS3} \geq d_3 \tag{4.36}$$
$$x_4 + x_8 + x_{DS4} \geq d_4 \tag{4.37}$$
$$y_1 + y_2 \leq 1 \tag{4.38}$$
$$y_3 + y_4 \leq 1 \tag{4.39}$$
$$0 \leq x_1 \leq u_1 y_1, 0 \leq x_2 \leq u_2 y_1, 0 \leq x_3 \leq u_3 y_2, 0 \leq x_4 \leq u_4 y_2 \tag{4.40}$$
$$0 \leq x_5 \leq u_5 y_3, 0 \leq x_6 \leq u_6 y_3, 0 \leq x_7 \leq u_7 y_4, 0 \leq x_8 \leq u_8 y_4 \tag{4.41}$$
$$x_1 + x_2 \geq y_1 L_1 \tag{4.42}$$
$$x_3 + x_4 \geq y_2 L_2 \tag{4.43}$$
$$x_5 + x_6 \geq y_3 L_3 \tag{4.44}$$
$$x_7 + x_8 \geq y_4 L_4 \tag{4.45}$$
$$y \text{ binary}, \ x \geq 0. \tag{4.46}$$

Let us extract from this the following structure that pertains to source node 1:

$$W_1 = \{(x, y) \ : \ x_1 + x_2 + x_3 + x_4 \leq s_1, \tag{4.47}$$
$$x_1 + x_2 \geq y_1 L_1, x_3 + x_4 \geq y_2 L_2, \tag{4.48}$$

$$0 \le x_1 \le u_1 y_1, 0 \le x_2 \le u_2 y_1, 0 \le x_3 \le u_3 y_2, 0 \le x_4 \le u_4 y_2, \quad (4.49)$$

$$y_1 + y_2 \le 1, y \text{ binary}\}. \quad (4.50)$$

Note that W_1 is an instance of the general set W_p of the following type, where $n_p \equiv 1$, and $l_{pq} \equiv 0 \ \forall q$ in the notation of TDSS2.

$$W_p = \{(x, y) \ : \ \sum_{g=1}^{G} \sum_{q \in J(g):(p,q) \in A} x_{pq} \le s_p \quad (4.51)$$

$$\sum_{q \in J(g)} x_{pq} \ge L_{pg} y_{pg} \ \forall g = 1, \ldots, G \quad (4.52)$$

$$0 \le x_{pq} \le u_{pq} y_{pg} \ \forall q \in J(g) : (p, q) \in A, g = 1, \ldots, G \quad (4.53)$$

$$\sum_{g=1}^{G} y_{pg} \le 1, y \text{ binary}\}. \quad (4.54)$$

Note that (4.51), (4.52), (4.53), and (4.54) correspond precisely to (4.17), the left-hand inequality in (4.20), (4.19), and (4.18), (along with (4.22)), respectively, for some source p, where $l_{pq} \equiv 0 \ \forall q$ and $n_p = 1$. Now, consider the following result, where conv$(\cdot)$ denotes the convex hull operator.

Theorem 4.2. Let W_p be defined by (4.51)-(54). Then, we have

$$\text{conv}(W_p) = \{(x, y) \ : \ \sum_{q \in J(g)} x_{pq} \le s_p y_{pg} \ \forall g = 1, \ldots, G \quad (4.55)$$

$$\sum_{q \in J(g)} x_{pq} \ge L_{pg} y_{pg} \ \forall g = 1, \ldots, G \quad (4.56)$$

$$0 \le x_{pq} \le u_{pq} y_{pg} \ \forall q \in J(g) : (p, q) \in A, g = 1, \ldots, G \quad (4.57)$$

$$\sum_{g=1}^{G} y_{pg} \le 1, y \ge 0\}. \quad (4.58)$$

Proof. It is sufficient to show that (i) any feasible solution to W_p belongs to the set (4.55)-(4.58), and that (ii) the set (4.55)-(4.58) has y binary valued at all its vertices. To show part (i), consider any feasible solution to W_p given by (4.51)-(4.54). If $y_{pg} = 0 \ \forall g$, then $x_{pq} \equiv 0 \ \forall q$ and this is feasible to (4.53)-(4.58). Otherwise, we have some $y_{pk} = 1$ for $k \in \{1, \ldots, G\}$ along with $y_{pg} = 0 \ \forall g \neq k$, and also $\sum_{q \in J(k)} x_{pq} \le s_p, \sum_{q \in J(k)} x_{pq} \ge L_{pk}, 0 \le x_{pq} \le u_{pq} \ \forall q \in J(k)$, along with $x_{pq} = 0 \ \forall q \in J(g)$ for all $g \neq k$. This solution is readily verified to be also feasible to (4.55)-(4.58), and so part (i) holds true.

To establish part (ii), consider the minimization of any linear objective function $\alpha^t x + \beta^t y$ subject to the constraints in (4.55)-(4.58). Call this problem **LP**. We must show that whenever an optimum exists, there exists an optimum at which y takes on binary values. Toward this end, assume that LP is feasible and define

$$\Delta_{pg} = \text{minimum}\{ \sum_{q \in J(g)} \alpha_{pq} x_{pq} : \sum_{q \in J(g)} x_{pq} \le s_p, \sum_{q \in J(g)} x_{pq} \ge L_{pg},$$

$$0 \le x_{pq} \le u_{pq}, \ \forall q \in J(g)\}, \text{ for each } g = 1, \ldots, G. \quad (4.59)$$

Then, by the separability of the problem LP, we can decompose it as

$$\textbf{LP:} \quad \underset{\sum_{g=1}^{G} y_{pg} \leq 1, y \geq 0}{\text{minimize}} \quad \{\beta^t y + \sum_{g=1}^{G} \min_{x} \sum_{q \in J(g)} \alpha_{pq} x_{pq} \tag{4.60}$$

$$\text{subject to} \quad \sum_{q \in J(g)} x_{pq} \leq s_p y_{pg}$$

$$\sum_{q \in J(g)} x_{pq} \geq L_{pg} y_{pg}$$

$$0 \leq x_{pq} \leq u_{pq} y_{pg} \ \forall q \in J(g)\}.$$

Since the inner minimization problem in (4.60) is identical to (4.59) when the right-hand sides of all the constraints in (4.59) are multiplied by y_{pg}, by linear programming duality, this inner minimization problem in (4.60) has an objective value of $\Delta_{pg} y_{pg} \ \forall g = 1, \ldots, G$. Consequently, we obtain that LP is equivalent to the projected problem

$$\text{minimize}\{\sum_{g=1}^{G} (\beta_{pg} + \Delta_{pg}) y_{pg} : \sum_{g=1}^{G} y_{pg} \leq 1, y \geq 0\}. \tag{4.61}$$

Note that (4.61) has an optimum at which y takes on binary values, and this completes the proof.□

Remark 4.5. As an alternative proof to Theorem 4.1, we could apply the RLT procedure of Sherali, Adams, and Driscoll (1996) by multiplying the constraints using the factors $y_{pg} \ \forall g = 1, \ldots, G$ and $(1 - \sum_{g=1}^{G} y_{pg})$, recognizing that $y_{pj} y_{pk} = 0 \ \forall j \neq k$, and $y_{pg}^2 = y_{pg} \ \forall g$. However, if we replace (4.54) by the constraint $\sum_{g=1}^{G} y_{pg} \leq n_p, y$ binary as in (4.18), where $n_p \geq 2$, then we would need to multiply the constraints using polynomial factors of order n_p composed from the bound-factors y_{pg} and $(1 - y_{pg}), g = 1, \ldots, G$, in order to construct the convex hull representation. Nonetheless, even when $n_p \geq 2$, the new set of constraints (4.55) are still valid, although they no longer imply (4.51) as they do in (4.55)-(4.58) (via (4.55) and (4.58)). Hence, in all cases, with the motivation that n_p is usually 1, we have incorporated the right-hand inequality in (4.20) based on (4.55), where $U_{pg} \leq s_p$ is also a justified coefficient to use in this constraint as discussed in Remark 4.1.□

To illustrate the comment in Remark 4.5, let us show how the relaxation can be further tightened when two blocks can possibly be formed. Accordingly, let us examine the corresponding substructure

$$W_p^2 = \{(x, y) : \quad \sum_{g=1}^{G} \sum_{q \in J(g):(p,q) \in A} x_{pq} \leq s_p \tag{4.62}$$

$$\sum_{q \in J(g)} x_{pq} \geq L_{pg} y_{pg} \ \forall g = 1, \ldots, G \tag{4.63}$$

$$0 \leq x_{pq} \leq u_{pq} y_{pg} \ \forall q \in J(g) : (p, q) \in A, g = 1, \ldots G \tag{4.64}$$

$$\sum_{g=1}^{G} y_{pg} \leq 2, y \text{ binary}\}. \tag{4.65}$$

Let $\lambda_r = 1$ denote the situation in which $y_{pr} = 1$ and $y_{pg} = 0 \ \forall g \neq r$, and let $\lambda_r = 0$ otherwise. Likewise, let $\lambda_{rt} = 1$ denote the situation in which $y_{pr} = y_{pt} = 1$ and $y_{pg} = 0 \ \forall g \neq r$ or t, where $r < t$, and let $\lambda_{rt} = 0$ otherwise. Then, treating λ as a binary variable vector, we can assert that

$$\sum_r \lambda_r + \sum_{r<t} \sum \lambda_{rt} \leq 1. \tag{4.66}$$

Now, consider the following result.

Theorem 4.3. The convex hull of W_p^2 is given by

$$W_p^{2*} \equiv \{(x, y) : x_{pq} = x_{pq}^r + \sum_{t<r} x_{pq}^{tr} + \sum_{t>r} x_{pq}^{rt} \ \forall q \in J(r), \ \forall r = 1, \ldots, G \tag{4.67}$$

$$y_{pr} = \lambda_r + \sum_{t>r} \lambda_{rt} + \sum_{t<r} \lambda_{tr} \ \forall r = 1, \ldots, G \tag{4.68}$$

$$\left[\begin{array}{l} L_{pr}\lambda_r \leq \sum_{q \in J(r)} x_{pq}^r \leq s_p \lambda_r \\[2mm] 0 \leq x_{pq}^r \leq u_{pq}\lambda_r \ \forall q \in J(r) \end{array} \right] \forall r = 1, \ldots, G \tag{4.69}$$

$$\left[\begin{array}{l} \sum_{q \in J(r)} x_{pq}^{rt} + \sum_{q \in J(t)} x_{pq}^{rt} \leq s_p \lambda_{rt} \\[3mm] \sum_{q \in J(r)} x_{pq}^{rt} \geq L_{pr}\lambda_{rt}, \ \sum_{q \in J(t)} x_{pq}^{rt} \geq L_{pt}\lambda_{rt} \\[3mm] 0 \leq x_{pq}^{rt} \leq u_{pq}\lambda_{rt} \ \forall q \in J(r), 0 \leq x_{pq}^{rt} \leq u_{pq}\lambda_{rt} \ \forall q \in J(t) \end{array} \right] \forall r \leq t \tag{4.70}$$

$$\sum_r \lambda_r + \sum_{r<t} \sum \lambda_{rt} \leq 1, \lambda \geq 0\}. \tag{4.71}$$

Proof. Similar to the proof of Theorem 4.2, we need to exhibit that (i) any feasible solution to W_p^2 belongs to the set W_p^{2*} defined by (4.67)-(4.71) and that (ii) the set W_p^{2*} has y binary valued at all its vertices. Part (i) can be verified as follows. Given $(x, y) \in W_p^2$, if $y \equiv 0$, then let $\lambda \equiv 0$. Furthermore, we must have $x \equiv 0$, and we let $x^r \equiv 0 \ \forall r$ and $x^{rt} \equiv 0 \ \forall r < t$. (Here, x^r denotes the vector $(x_{pq}^r, q \in J(r)) \ \forall r$, and x^{rt} denotes the vector $(x_{pq}^{rt}, q \in J(r) \cup J(t)) \forall r < t$.) If only one variable $y_{pr} = 1$ with $y_{pg} = 0 \ \forall g \neq r$, let $\lambda_r = 1, \lambda_g = 0 \ \forall g \neq r$, and let $\lambda_{ij} = 0 \ \forall i < j$. Accordingly, let $x_{pq}^r = x_{pq} \ \forall q \in J(r)$, and set $x^g = 0 \ \forall g \neq r$, and $x^{ij} \equiv 0 \ \forall i < j$. On the other hand, if some two variables $y_{pr} = y_{pt} = 1$, where $r < t$, with $y_{pg} = 0 \ \forall g \neq r, t$, then let $\lambda_i = 0 \ \forall i, \lambda_{rt} = 1$ and $\lambda_{ij} = 0 \ \forall i < j, (i, j) \neq (r, t)$. Similarly, set $x_{pq}^{rt} = x_{pq} \ \forall q \in J(r) \cup J(t)$, and set $x^i \equiv 0 \ \forall i$, and $x^{ij} \equiv 0 \ \forall i < j, (i, j) \neq (r, t)$. Then it is readily verified that the resulting solution is feasible to $(W_p^{2*}$. This proves Part (i).

To establish Part (ii), we need to show that for any objective vector (α, β), the linear program **LP**: Minimize $\{\alpha^t x + \beta^t y : (x, y) \in W_p^{2*}\}$ has an optimal solution at which y takes on binary values, whenever an optimum exists. Assuming that an optimum exists, i.e., assuming that LP is feasible, define the following entities.

$$\Delta^r \;=\; \text{minimum} \sum_{q \in J(r)} \alpha_{pq} x_{pq}^r$$

$$\text{subject to}$$

$$L_{pr} \le \sum_{q \in J(r)} x_{pq}^r \le s_p \tag{4.72}$$

$$0 \le x_{pq}^r \le u_{pq} \;\; \forall q \in J(r)$$

for each $r = 1, \ldots, G$. Similarly, for each $r < t$, define

$$\Delta^{rt} = \text{minimum} \sum_{q \in J(r) \cup J(t)} \alpha_{pq} x_{pq}^{rt}$$

$$\text{subject to}$$

$$\sum_{q \in J(r) \cup J(t)} x_{pq}^{rt} \le s_p, \tag{4.73}$$

$$\sum_{q \in J(r)} x_{pq}^{rt} \ge L_{pr}, \;\; \sum_{q \in J(t)} x_{pq}^{rt} \ge L_{pt}$$

$$0 \le x_{pq}^{rt} \le u_{pq} \;\; \forall q \in J(r) \cup J(t).$$

Then, as in the proof of Theorem 4.2, using linear programming duality and examining W_p^{2*} versus (4.72) and (4.73), we deduce that

$$\mathbf{LP} \;\equiv\; \text{Minimize} \left\{ \; \sum_r (\Delta^r + \beta_{pr}) \lambda_r + \sum_{r < t} \sum (\Delta^{rt} + \beta_{pr} + \beta_{pt}) \lambda_{rt} \right.$$

$$\text{subject to} \quad \sum_r \lambda_r + \sum_{r < t} \sum \lambda_{rt} \le 1 \tag{4.74}$$

$$\lambda \ge 0.$$

Clearly, λ is binary valued at an optimal solution to (4.74). Consequently, noting (4.68), and (4.71), so is y, and this completes the proof.□

Note that in order to construct the convex hull in the case treated by Theorem 4.3, we needed to define multiple commodities x_{pq}^r for $q \in J(r), r = 1, \ldots, G$, and x_{pq}^{rt} for $q \in J(r) \cup J(t)$, $\forall r = 1, \ldots, G-1, t = r+1, \ldots, G$, to collectively represent the flow quantities x as per (4.55). Hence, for each set $J(r), r = 1, \ldots, G$, each associated flow quantity x_{pq} for $q \in J(r)$ inherits G copies, one copy (x_{pq}^r) corresponding to the rth set being blocked alone, and $(G-1)$ copies $(x_{pq}^{rt}$ for $t > r$ and x_{pq}^{tr} for $t < r)$ corresponding to the rth set being blocked in combination with one other of the remaining $(G-1)$ sets. (All flows are zeros if no blocks are formed out of source node p.) Hence, a tighter relaxation can be generated in this fashion, but at the expense of a larger problem. The benefits of this approach need to be investigated, perhaps as a follow-on study.

To illustrate the utility of the right-hand inequalities in (4.20), we solved the following test problem with and without these constraints using the commercial package CPLEX-MIP.

Example 4.1
Problem data specifications:
Number of source nodes = 424.
Number of demand nodes = 66.
Number of arcs = 5495.

Solution without the right-hand inequalities in (4.20):
Optimal objective value = 1.028×10^9.
Number of branch-and-bound nodes enumerated = 956.
Number of iterations = 9525.
CPU time = 80.96 seconds.

Solution with the right-hand inequalities in (4.20):
Optimal objective value = 1.028×10^9.
Number of branch-and-bound nodes enumerated = 1.
Number of iterations = 513.
CPU time = 5.42 seconds.

4.3   Design of Specialized Heuristic Solution Procedures

As dictated by strict practical requirements, our model needs to have a run-time that is less than five minutes. The presence of integer variables in Model **TDSS2** precludes the exact solution of this problem within this time limit, as seen in Table 4.1 given below. This table considers seven practical, realistic test problems of various sizes, and presents computational experience on using the commercial package CPLEX-MIP to solve these problems. Indeed, the difficulty of solving TDSS2 to optimality as evident from Table 4.1, is to be expected since it can be shown that our model belongs to the NP-Hard class of optimization problems. Thus, in this section, we discuss several alternatives that have a more predictive run-time characteristic.

However, before we turn our attention to designing these alternative schemes, in order to reduce the size of the problem that we represent in the model **TDSS2**, as well as to tighten this representation, let us perform certain logical reductions on the problem, *prior to* inputting this problem into some exact or heuristic solution approach. As observed earlier, when $l_{pq} > 0$, by (4.19), we must have $y_{p,g(q)} = 1$, which in turn could possibly eliminate other y, and therefore x, variables from the problem via (4.18)-(4.20). Alternatively, some *a priori* fixed blocks might be designated (or prohibited) by the car distributor, based on which certain blocking y-variables are given to be fixed at 1 or 0.

Hence, for each source p, let us define

$$G_p^+ = \{g : y_{pg} = 1 \text{ is fixed}\}, \tag{4.75}$$

Table 4.1: Exact Solution of Test Problems Using CPLEX-MIP.

Problem	# of Source Nodes	# of Demand Nodes	# of Arcs	# of Branch-and-Bound Nodes Enumerated	CPU time (mins)
1	424	66	5,495	0	0.08
2	424	66	5,495	100	0.33
3	424	66	5,495	1012	3.05
4	1057	417	65,876	N/A	*
5	1144	444	75,490	N/A	*
6	1200	432	87,706	N/A	*
7	1362	530	146,970	N/A	*

* exceeded preset limit of 240 cpu mins.

and

$$G_p^f = \{g : y_{pg} \text{ is } free, \text{ i.e., the blocking variable } y_{pg} \text{ can possibly take on a value of 0 or 1}\}. \tag{4.76}$$

Furthermore, let

$$n_p = \text{maximum number of } additional \text{ blocks that can be dispatched from source } p \text{ (other than the fixed ones).} \tag{4.77}$$

Notice that G_p^f is relevant only for sources p for which $s_p > 0$ and $n_p > 0$. Furthermore in order to tighten the relaxation of TDSS2, for each source node p, we can replace s_p by

$$s_p \leftarrow \min\{s_p, \sum_{q \in J(g):g \in G_p^+} u_{pq} + \sum(n_p \text{ largest values of } \sum_{q \in J(g)} u_{pq} \text{ from among } g \in G_p^f)\} \tag{4.78}$$

where n_p is the *residual* number of blocks that be formed at source p as defined in (4.77). Whenever the second term is the minimand in (4.78) is smaller, we obtain a potential tightening of the relaxation of TDSS2 via (4.17), (4.19), and (4.20). Hence, this is likely to be most effective when several source nodes have blocking parameters $n_p = 1$, which is frequently the case in practice. To illustrate the effect of (4.78), consider the following example.

Example 4.2
Problem Data Specifications:
Number of source nodes = 424
Number of demand nodes = 66
Number of arcs = 5,495.

Solution without the tightening of constraints via (4.78):
Optimal objective value = 1.11×10^9.
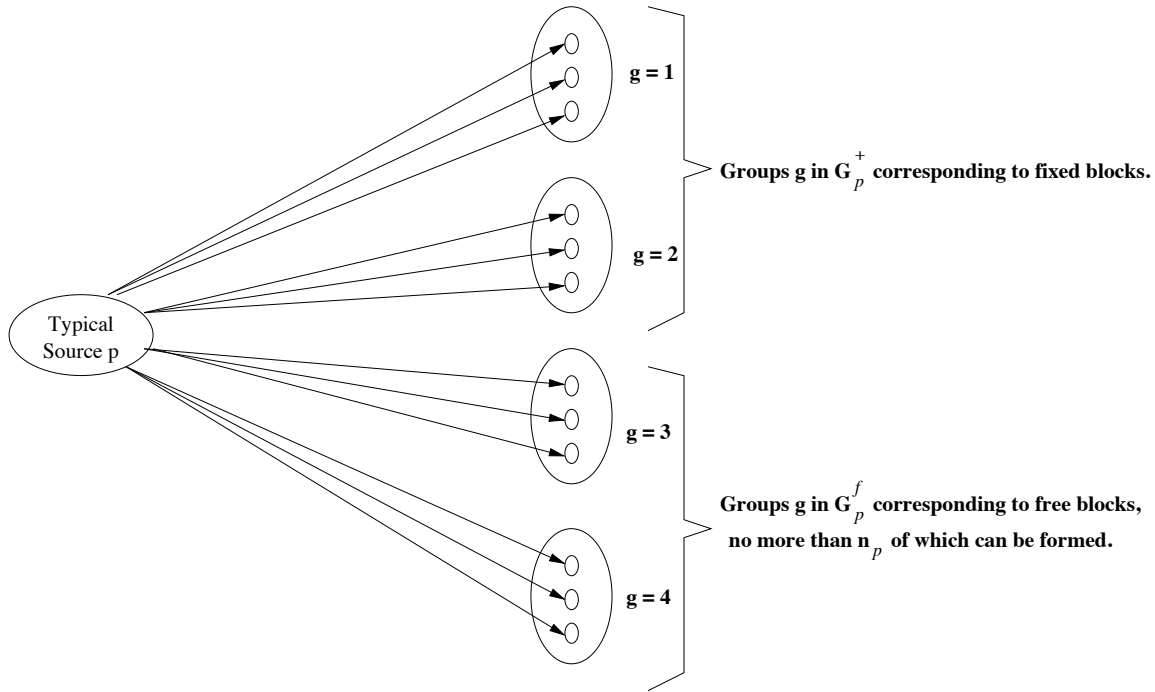
Figure 4.4: Group Partitions for Each Source.

Number of branch-and-bound nodes enumerated = 370.
Number of iterations = 5153.
CPU time = 48.95 seconds.

Solution with the tightening of constraints via (4.78):
Optimal objective value = 1.11×10^9.
Number of branch-and-bound nodes enumerated = 328.
Number of iterations = 4761.
CPU time = 45.74 seconds.

Based on the foregoing constructs, we can rewrite model **TDSS2** as follows, where we note that in (4.19), we have $l_{pq} \equiv 0$ for $q \in G_p^f$, and that the constraints in $y \in Y$ have been accommodated within this restatement according to the fixed y-variables. In other words, pairs of y variables that are equated to each other are assumed to define a common group, the y variables that are fixed at 0 result in an elimination of the correspoding source-to-group connections, and the y variables that are fixed at 1 are represented via the sets (4.75). Figure 4.4 illustrates the conceptual description of the problem.

$$\textbf{TDSS2: } \text{Minimize} \quad \sum_p \sum_q C_{pq} x_{pq} + \sum_q \mathcal{M}(DS, q) x_{DS,q} \qquad (4.79)$$

$$\text{subject to} \quad \sum_p x_{pq} + x_{DS,q} \geq d_q \; \forall q \tag{4.80}$$

$$\sum_q x_{pq} \leq s_p \; \forall p \tag{4.81}$$

$$\sum_{g \in G_p^f} y_{pg} \leq n_p \; \; \forall p \ni G_p^f \neq \emptyset \tag{4.82}$$

$$l_{pq} \leq x_{pq} \leq u_{pq} \; \; \forall p, q \in J(g), g \in G_p^+ \tag{4.83}$$

$$0 \leq x_{pq} \leq u_{pq} y_{pg} \; \; \forall p, q \in J(g), g \in G_p^f \tag{4.84}$$

$$L_{pg} \leq \sum_{q \in J(g)} x_{pq} \leq U_{pg} \; \forall p, g \in G_p^+ \tag{4.85}$$

$$L_{pg} y_{pg} \leq \sum_{q \in J(g)} x_{pq} \leq U_{pg} y_{pg} \; \forall p, g \in G_p^f \tag{4.86}$$

$$x_{pq} \equiv 0 \; \forall q \in J(g) : y_{pg} = 0, \forall p,$$
$$\text{and } y_{pg} \text{ binary } \forall p, g \in G_p^f. \tag{4.87}$$

Note that for the heuristic developed next, given (4.9)-(4.14), we do not need to be concerned with the upper bounding restrictions in (4.83)-(4.86), since these will be automatically handled by the supply and demand constraints, along with the activation of appropriate blocks. Furthermore, we will assume that the minimum block size $l_{pg}^{\min}$ in (4.10) is 1, and we will let l_{pq} and n_p govern minimal block flows and the number of blocks formed, respectively.

4.3.1 Rudimentary Primal Heuristic

To clarify the presentation, we describe in this section one basic rudimentary heuristic procedure that will constitute a particular single option within the framework of the generalized heuristic scheme that is developed in the sequel. This method executes three basic steps as follows.

Step 1: Order the demand nodes $q \equiv (j, \tau)$ in a list $\mathcal{D}$ according to the following rules.

(a) The nodes corresponding to the imminent period come first, then those corresponding to the next period, and so on.

(b) For each time period, the nodes corresponding to locations having a higher nearest integer-rounded plant priority index come first. (A fraction of 0.5 is rounded up.) Ties for these nearest integer-rounded plant priorities are broken by placing the node having the greater demand value first. For example, given two demand locations A and B with respective priorities $P_A = 7.4$ and $P_B = 6.8$, since their nearest integer is 7 in both cases, these nodes would be ordered according to their demand values.

Step 2: As described above, perform the logical fixings of the y-variables to 0 or 1 values based on the lower bounds l_{pq}, wherever possible, and define the sets G_p^+ and $G_p^f \; \forall p$ as in

Equations (4.75) and (4.76). Define the residual number of blocks n_p that can be formed at each source p as in (4.77), update s_p $\forall p$ via (4.78), and examine Model TDSS2 given in the form (4.79)-(4.87). Initialize $x_{pq} = l_{pq}$ $\forall (p,q) \in A$, and adjust the corresponding residual supplies and demands for each $(p,q) \in A$ such that $l_{pq} > 0$. The revised residual lower and upper bounds for any further flow augmentations conducted for arc (p,q) (i.e., the bounds on any further increase in the value of x_{pq}) are 0 and $u_{pq} - l_{pq}$, respectively. (This list of residual upper bounds is carried separate from the original data and is updated as the algorithm progresses according to the scheme described below.) Note that now, all arcs have lower bounds of zeros, and nonnegative upper bounds. In addition, flows that are prohibited due to the corresponding blocking variable being fixed at zero (see Equation (4.87)) have both lower and upper bounds equal to zero. Similarly, at any stage of the heuristic, we will maintain lists of residual supplies and demands. Hence, s_p and d_q will henceforth refer to *residual* supply and demand quantities, respectively.

Step 3: Perform the procedure illustrated in the flow-chart of Figure 4.5. Note that the final step of the foregoing heuristic procedure improves the constructed solution using the network flow option NETOPT of CPLEX, by feeding in the corresponding y variable values (y^*) obtained as a fixed input, and then determining an optimal completion flow solution x^* (see Theorem 4.1). The heuristic flow solution that is constructed prior to this final step can be used as an advanced-start for this purpose, although NETOPT is fast enough so as to not require an advanced-start. On the other hand, if **TDSS2** is to be solved to optimality using CPLEX-MIP, the initial linear programming relaxation itself turns out to be a bottleneck for large sized problems (number of arcs exceeding about 100,000). In this case, the solution (x^*, y^*) can be used to provide an advanced-start to CPLEX-MIP, or more directly, the arcs on which flow is augmented via the schema of Figure 4.5 can be input as candidates for crashing into the basis. Theorem 4.4 below establishes that these variable columns are linearly independent. (Note that if the flow on the arcs for which $l_{pq} > 0$ have not been further augmented by the heuristic, such arcs can be considered nonbasic at their lower bounds.) Furthermore, using the solution y^*, we might designate the unit y-variables as basic (although these could be taken as being nonbasic at their upper bounds). This basis can be completed using suitable slack variables, and CPLEX would then automatically attempt to crash the suggested variables into the basis, using artificial variables as necessary.

Theorem 4.4. The columns of the arc variables in **TDSS2** for which the procedure of Figure 4.5 augments flows are linearly independent.

Proof. By the nature of the underlying transportation network structure, it is sufficient to show that the graph constructed using the arcs on which the flow has been augmented contains no cycles. On the contrary, suppose that it does, and consider the very first time that a cycle is created when some arc (p,q) has its flow augmented. Since this is the first time a cycle has been created, it must be that (p,q) has been added to a tree subgraph of the current graph comprised of augmented flow arcs. Figure 4.6 illustrates a generic cycle formed by the nature of the transportation structure.
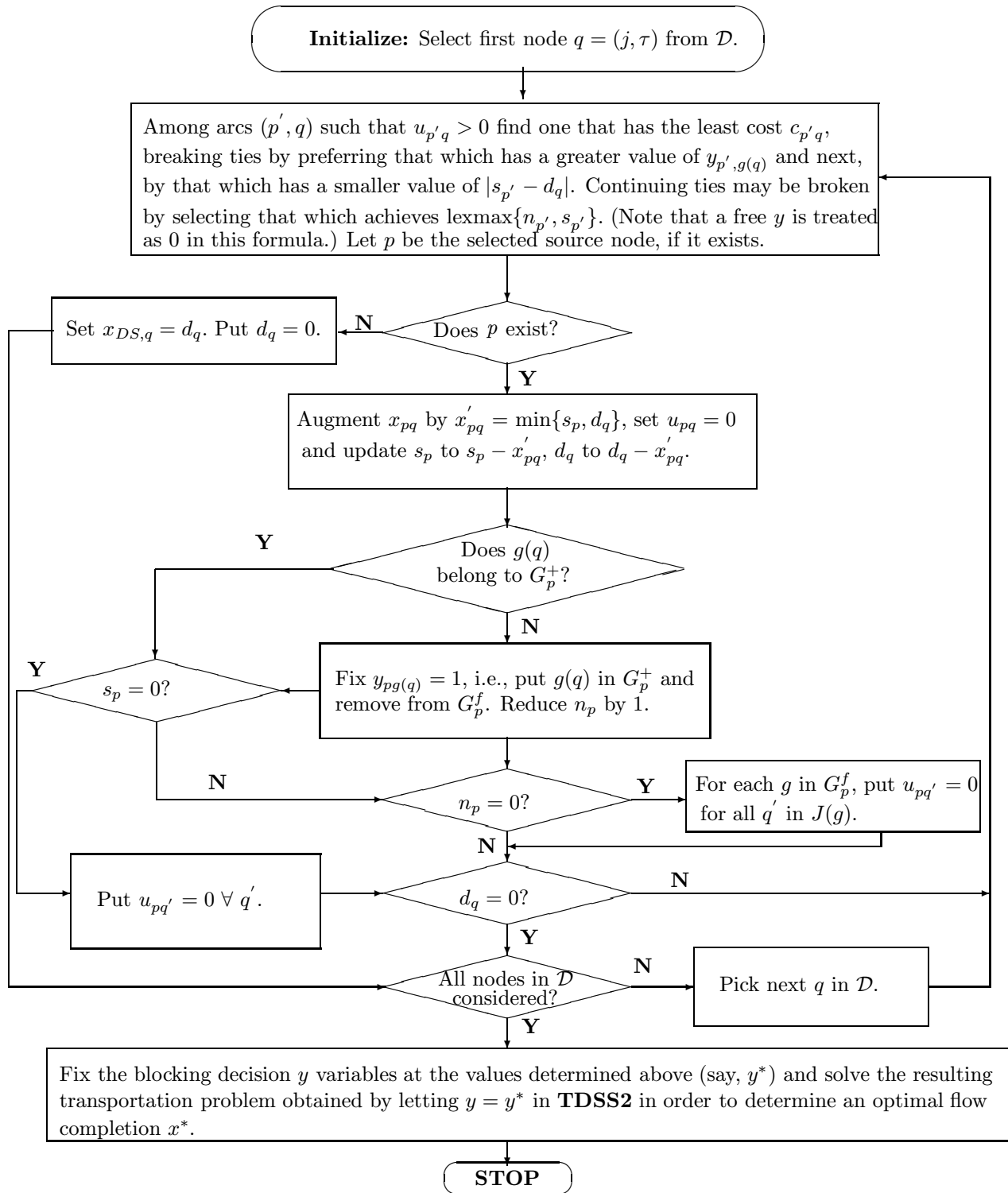
Figure 4.5: Flow Chart for a Rudimentary Primal Heuristic.
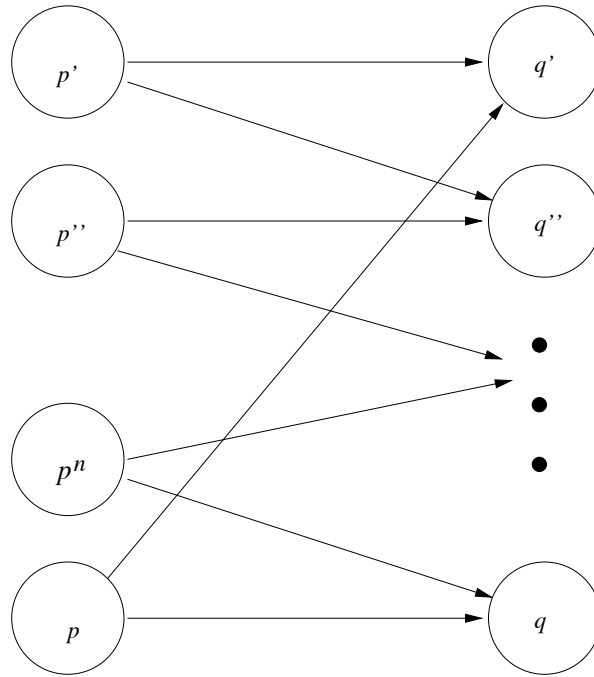
Figure 4.6: Generic cycle formed upon introducing (p, q).

Consider node q' in Figure 4.6, which was necessarily considered before node q. If arc (p, q') was considered before arc (p', q') when treating node q', then it must have exhausted s_p but must not have satisfied the residual demand at q', which is a contradiction to the selection of arc (p, q) for augmentation. Hence, arc (p', q') was considered before arc (p, q'). Since another flow augmentation to q' occurred after (p', q') was considered, this means that (p', q') must have exhausted $s_{p'}$, but must not have satisfied the demand at q'. However, p' also feeds q''. This implies that (p', q'') must have been considered before (p', q') and must have satisfied all the demand at q'' but not exhausted $s_{p'}$. Again, this in turn means that the arc (p'', q'') must have been considered prior to this, exhausting all the supply at p''. Hence, the flow on the other arc incident at p'' was augmented before this, satisfying the demand of the node incident to it without exhausting $s_{p''}$. If this node incident to p'' is q itself, we have a contradiction to the need for considering (p, q). Otherwise, continuing in this fashion and noting the pattern that for the pair of arcs incident at each source $p', p'', \cdots, p^n$ in the configuration of the cycle as depicted in Figure 4.6, the top one exhausts the supply at the corresponding source node and the bottom one satisfies the demand at the corresponding incident node, we have that the demand at q was already satisfied when (p^n, q) was considered, a contradiction. This completes the proof. □

4.3.2 Generalized Blocking Heuristics

In this section, we present the overall structure of the generalized Blocking Heuristics that have been developed in providing a fast solution to the model **TDSS2**. To facilitate further discussion, let us define three terminologies as follows.

1. **Depth of Search** : This defines the next set of nodes Q selected from the top of the demand node list $\mathcal{D}$ for consideration in satisfying demands. We consider four Depths of Search below.

   - **DS 1** : $|Q| = 1$, where Q is selected from the top of $\mathcal{D}$. This corresponds to the Rudimentary Primal Heuristic of Section 4.3.1.

   - **DS 2** : Q is selected as the (next) set of nodes having the same time period.

   - **DS 3** : Q is selected as the (next) set of nodes such that the time period stays the same, and the *absolute* difference in priority between selected *consecutive* nodes is ≤ 0.5.

   - **DS 4** : Q is selected as the next two time period's nodes. If the first choice of Q exceeds 80 nodes in size, then Q is determined as the (next) set of nodes of cardinality min{80, number of nodes left to be considered from $\mathcal{D}$}.

2. **Vogel's Functions** : These are opportunity cost-based functions that are designed to judiciously select the next demand node to be (partially) satisfied from among the demand nodes in Q, when a choice exists (i.e., when $|Q| > 1$). Three functions are investigated in our model as listed below. The first of these is the standard Vogel's computation used in transportation planning (see Bazaraa et al., 1990, for example), while the other two are alternative proposed opportunity cost assessments. Note also that if for any selected demand node q, if the only arc incident at q that has a positive residual capacity is the arc (DS, q), the flow on this arc is augmented by d_q and node q is dropped from further consideration. In addition, recall below that s_p and d_q represent *residual* supply and demand quantities.

   - **VF 1** : Let $v_1(q) = [$ second smallest over p' of $\{c_{p'q} : u_{p'q} > 0\}$ - smallest over p' of $\{c_{p'q} : u_{p'q} > 0\}]$. Pick that $q \in Q$ which yields the maximum value of $v_1(q)$.

   - **VF 2** : Let $v_2(q) = d_q \cdot v_1(q)$, where $v_1(q)$ is as defined for **VF1** above. Select that $q \in Q$ which yields the maximum value of $v_2(q)$.

   - **VF 3** : Let $v_3(q) = v_3^*(q) - v_3'(q)$, where

$$v_3'(q) = \min\{ \sum_{i:u_{iq}>0} c_{iq}x_{iq} : \sum_{i:u_{iq}>0} x_{iq} = d_q, 0 \leq x_{iq} \leq s_i \ \forall i\}.$$

Note that $v_3'(q)$ is the minimum cost solution for satisfying the (residual) demand at q using the available residual supplies. Furthermore, define $v_3^*(q)$ to be the value of $v_3'(q)$

computed after restricting $x_{i*q} = 0$ (or making $u_{i*q} = s_{i*} = 0$ temporarily), where i^* achieves the minimum over i of $\{c_{iq} : u_{iq} > 0\}$. Then, select that $q \in Q$ which yields the maximum value of $v_3(q)$.

3. **Dispatching Rules :** These rules prescribe a supply node to (partially) satisfy the demand at the node $q \in Q$ chosen by the activated Vogel's Function. We consider two Dispatching Rules as described below.

- **DR1** : Pick (a real source) p, where $u_{pq} > 0$, and such that p achieves the lexicographic minimum over p' of $\{c_{p'q}, -y_{p',g(q)}, |s_{p'} - d_q|, -n_{p'}, -s_{p'}\}$. (Recall that this is the Dispatching Rule used for the Rudimentary Primal Heuristic of Section 4.3.1- see Figure 4.5.)

- **DR2** : Find p according to **DR1**. If $y_{p,g(q)} = 1$, use p. Else, find $\overline{p}$ such that $\overline{p} \in \text{arglexmin}\{c_{p'q}, |s_{p'} - d_q|, -n_{p'}, -s_{p'} : u_{p'q} > 0, y_{p',g(q)} = 1, \text{ and } c_{p'q} \leq 1.1c_{pq}\}$. If such a $\overline{p}$ exists, use $\overline{p}$ in lieu of p. (The motivation here is that if there exists another source $\overline{p}$ from which a block to the demand group containing node q has already been formed, and for which the increase in expected transit time to node q over that from source node p is no more than 10%, then such a node might be a preferable alternative for feeding q). Else, use p.

Note that for the Depth of Search **DS1**, we do not need any Vogel's Function. Hence, there are 20 combinations of **DS, VF**, and **DR** that we can select from, depending on specified values of these option parameters. We will denote **DS** = 1, $\cdots$, 4, **VF** = 1, $\cdots$, 3, and **DR** = 1, 2 to correspond to the respective Depth of Search, Vogel's Function, and Dispatching Rule strategies. For example, **(DS, VF, DR)** = (4, 3, 2) represents the composition of a heuristic procedure using the last of each of these options.

Now, we are ready to describe the general structure of the *Blocking Heuristic*. Figure 4.7 presents a flow chart for this procedure, based on any specified choice of **(DS, VF, DR)**.

4.3.3   NETOPT and LP-based Heuristics

This heuristic employs one of two options as its initial step. In the first option, the linear programming relaxation (**LP**, say) of **TDSS2** is run for a fixed maximum number of iterations (say, 1,000) and the resulting solution (x', y') (perhaps infeasible) is extracted. In the second option, the transportation model **TDSS1** is solved to optimality using the NETOPT solver in CPLEX, and the corresponding optimal flow x' is determined. If the resulting solution is optimal to LP or TDSS1, as the case might be, and is feasible to the blocking restrictions, then it is optimal to TDSS2 and hence can be used as the prescribed solution. Otherwise, the procedure of Figure 4.7 for the Blocking Heuristic is adopted, except that the Depth of Search strategy is fixed at $DS = 1$ (and hence $VF = \emptyset$), and the Dispatching Rule is modified as follows.
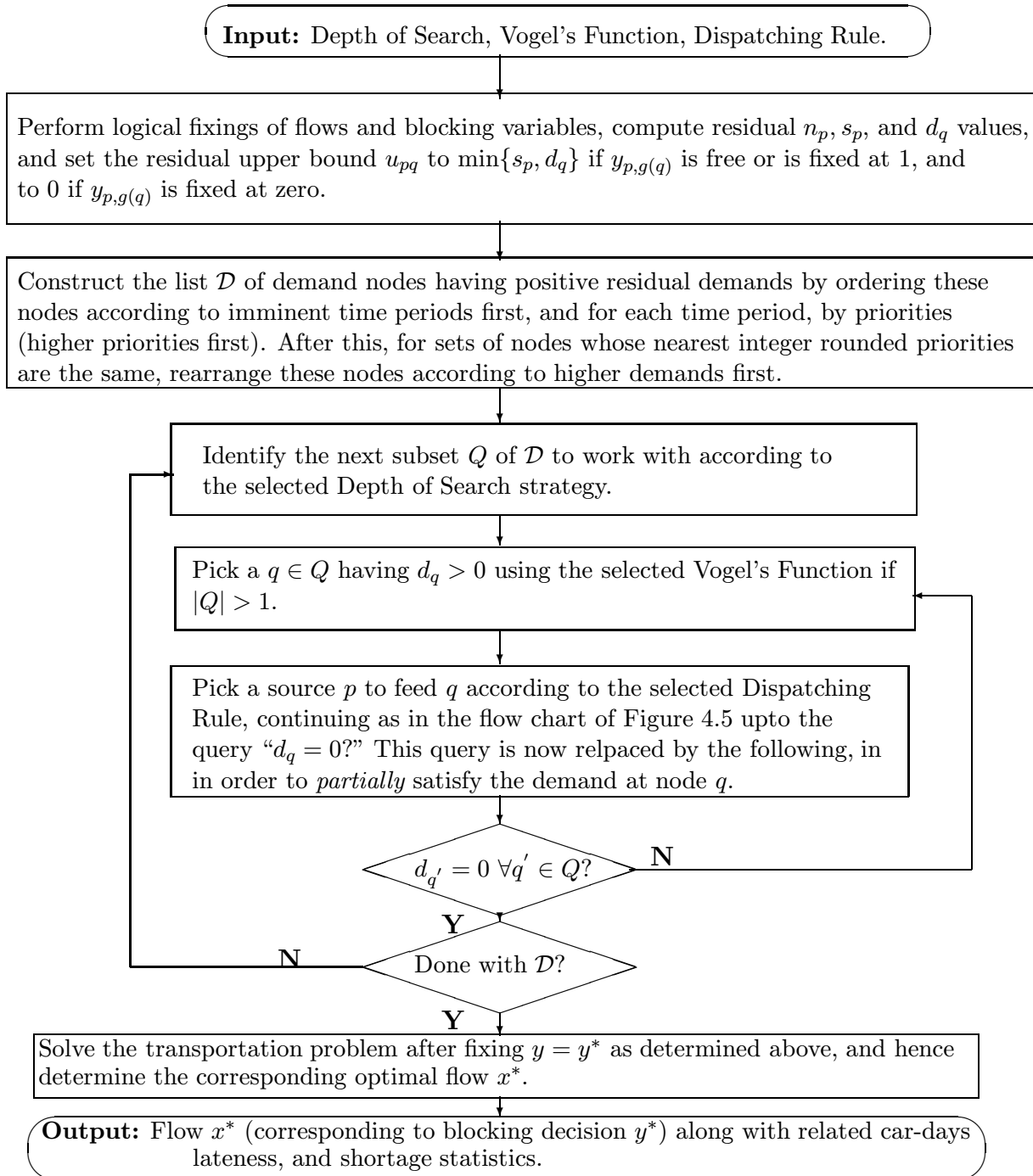
Figure 4.7: General Structure of the Blocking Heuristic.

Given a selected $q \in \mathcal{D}$, pick a source node p to feed this according to one of the following rules. (When using the NETOPT option, these rules are identical since $y' \equiv 0$ in this case.)

DRLP1 : Find p such that

$$p \in \text{arglexmax}_{p'} \{y'_{p',g(q)}, x'_{p',q}, -c_{p'q}, -|s_{p'} - d_q|, n_{p'}, s_{p'} : u_{p'q} > 0\}.$$

DRLP2 : Find p such that

$$p \in \text{arglexmax}_{p'} \{x'_{p',q}, y'_{p',g(q)}, -c_{p'q}, -|s_{p'} - d_q|, n_{p'}, s_{p'} : u_{p'q} > 0\}.$$

Remark 4.6. Note that if TDSS1 returns a solution that is feasible to the blocking restrictions, and we continue with the application of the foregoing heuristic instead of terminating the procedure with the recognition that the solution at hand is indeed optimal to TDSS2, then we might produce a different, possibly suboptimal, solution. The reason for this is that the dispatching rule of Figure 4.5 might ascribe a different flow on a selected arc from that obtained in TDSS1, because of the values of the current residual supplies and demands. For example, for the test case TTX1 in Chapter 5 (see Tables 5.4 and 5.11), TDSS1 yields a solution of value 1.03 x 10^9 with no blocking violations. Hence, this solves TDSS2 as well, and is the same solution as that produced by the NETOPT-based heuristic. However, if we ignore this and apply the NETOPT-based heuristic scheme, we obtain a solution having an objective value of 1.04 x 10^9 with an extra railcar short on day 2.

Remark 4.7. In the foregoing LP-based heuristic, we attempted using an iteration limit of 3,000 as well, but the increase in effort was neither acceptable from the viewpoint of our computational limit, nor was it justifiable with respect to the resulting performance. We also attempted a hybrid variation of the Blocking and LP-based heuristic in which a solution from a Blocking Heuristic was used to construct an advanced-start basis for the linear programming run in the LP-based heuristic. This too did not yield competitive results. Likewise, we attempted supply-side heuristic variants of the Blocking Heuristic in which we considered a list of supply nodes and then designed rules to select a source node from this list and to dispatch its supply to (partially) satisfy the demand of some identified demand node. Our preliminary runs indicated that the demand-side strategy used by the above Blocking Heuristics yielded a superior performance. Hence, the procedures that we now test within the framework of the overall scheme of Figure 4.7 in the computational results presented in Chapter 5 are the following 23 methods.

Blocking Heuristics:

1. **(DS, VF, DR)** = (1, ∅, 1)

2. **(DS, VF, DR)** = (1, ∅, 2)

3. **(DS, VF, DR)** = (2, 1, 1)

4. **(DS, VF, DR)** = (2, 1, 2)

5. **(DS, VF, DR)** = (2, 2, 1)

6. **(DS, VF, DR)** = (2, 2, 2)

7. **(DS, VF, DR)** = (2, 3, 1)

8. **(DS, VF, DR)** = (2, 3, 2)

9. **(DS, VF, DR)** = (3, 1, 1)

10. **(DS, VF, DR)** = (3, 1, 2)

11. **(DS, VF, DR)** = (3, 2, 1)

12. **(DS, VF, DR)** = (3, 2, 2)

13. **(DS, VF, DR)** = (3, 3, 1)

14. **(DS, VF, DR)** = (3, 3, 2)

15. **(DS, VF, DR)** = (4, 1, 1)

16. **(DS, VF, DR)** = (4, 1, 2)

17. **(DS, VF, DR)** = (4, 2, 1)

18. **(DS, VF, DR)** = (4, 2, 2)

19. **(DS, VF, DR)** = (4, 3, 1)

20. **(DS, VF, DR)** = (4, 3, 2)

LP-based Heuristic:

21. **(DS, VF, DR) = (1, ∅, DRLP1)** (denoted **LP-(1,∅,1)**)

22. **(DS, VF, DR) = (1, ∅, DRLP2)** (denoted **LP-(1,∅,2)**)

NETOPT-Heuristic:

23. **(DS, VF, DR) = (1, ∅, DRLP2 ≡ DRLP1)** (denoted **NET-(1,∅,1)**). □

Chapter 5

Illustrative Examples and Computational Experience

In this chapter, we begin by presenting an illustrative example that exhibits the salient differences between the models TDSS1 and TDSS2, and the relative performances of the 23 heuristic procedures developed in the foregoing chapter. Following this, we evaluate these heuristic procedures using a test-bed of five realistic data sets that have been synthesized by TTX based on historical information. As we shall see, two Blocking Heuristics ((DS, VF, DR) = (4, 3, 2) and (DS, VF, DR) = (2, 3, 2)) emerge as the two most promising heuristic procedures over all the test instances. Due to the better look-ahead capability of the (4,3,2) heuristic procedure, however, a variant of this procedure that operates within the time limitation constraint is recommended for implementation.

5.1   Illustrative Examples

Consider the 6×4 transportation problem network depicted in Figure 5.1. This problem can be conceived as a special instance of our problem by assuming that all source nodes pertain to period $t = 1$ and have $n_p = 1 \ \forall \ p = 1, \ldots, 6$. Furthermore, all demand nodes $q = 1, \ldots, 4$, pertain to τ sufficiently large so that the expected transit times (days) shown in Figure 5.1 represent the costs on the arcs, and that we obtain an on-time delivery to all the nodes via any of the routes, despite any inherent randomness in the transit times. Hence, regardless of demand node priorities, this cost matrix itself represents the penalized cost coefficients C_{pq} for the real arcs $(p, q) \in A$. Moreover, each demand node is assumed to belong to an individual group for blocking purposes. Consequently, we have $G = 4$ groups in this example. For the arcs incident at the *dummy source* node, assuming a uniform priority of 5.5 for all the demand nodes, we compute a penalty cost

$$\mathcal{M}(DS, q) = 15637 \ \forall \ q = 1, \ldots, 4. \tag{5.1}$$

The respective supplies s_p and demands d_q for $p = 1, \ldots, 6$ and $q = 1, \ldots, 4$ are also shown in Figure 5.1. The lower and upper bounds on each arc (p, q) are taken as $l_{pq} = 0$ and $u_{pq} = \min\{s_p, d_q\}$.

Optimal Solution of TDSS1:

Using NETOPT-CPLEX to solve this transportation problem, we obtain the solution depicted in Figure 5.2. The objective value of this solution is 1095 car-days.

Optimal Solution of TDSS2:

Observe from Figure 5.1 that the source nodes 3, 4, and 6 violate the blocking restriction of requiring no more than one ($n_p = 1$) block to be formed at each such node. Including this stipulation, and using CPLEX-MIP, the corresponding model TDSS2 yields the optimal solution depicted in Figure 5.3. Note that because of this blocking restriction, we now have unmet demands of 1 at each of the two nodes 2 and 4. The total penalized cost of this solution is given by 32536, while the expected car-days pertaining to the real delivered railcars is 1262 car-days.

To examine a slight variation of this problem instance, let us assume that the nodes 2 and 4 at which shortages have occurred in the previous run of model TDSS2 have (scaled) priorities of 10, while the other two nodes 1 and 3 have priorities of 1. Now, in lieu of (5.1), the penalties on the (dummy) unmet demand arcs are computed as

$$\mathcal{M}(DS, q) = 9781 \text{ for } q = 1 \text{ and } 3. \tag{5.2}$$

$$\mathcal{M}(DS, q) = 21493 \text{ for } q = 2 \text{ and } 4. \tag{5.3}$$

Using these penalties in Model TDSS2 and running this revised scenario, yields the optimal solution shown in Figure 5.4. Note that for this variant, we now have a shortage of 4 cars at node 3 alone. The total penalized objective function is given by 40157, while the total expected car-days for the real delivered railcars is 1033 car-days.

Solution of Variants 1 and 2 Using the 23 Heuristic Procedures of Section 4.3.3:

Let us refer to the foregoing two variants of Model TDSS2 under the penalty structures (5.1) and (5.2, 5.3) as **Variant 1** and **Variant 2**, respectively. Tables 5.1 and 5.2 present the results obtained by applying each of the proposed 23 heuristic procedures identified in Section 4.3.3, to Variants 1 and 2, respectively. For Variant 1, the best solution has been obtained by Heuristics (2,2,1), (2,2,2), (2,3,1), (2,3,2), (3,2,1), (3,2,2), (3,3,1), (3,3,2), (4,2,1), (4,2,2), (4,3,1), (4,3,2), and LP-(1,∅,2). Figure 5.5 depicts the corresponding flow distribution for this solution. Note that the total shortage is given by 4, with node 3 incurring a shortage. The resulting total penalized cost is 63581, and the corresponding car-days for the real delivered cars is 1033. This is contrasted with the optimal solution for this variant for which the total shortage was 2 and the total penalized and real car-days were, respectively, 32536 and 1262. Similar results for Variant 2 are displayed in Table 5.2. For this variant, the best solution has been obtained by Heuristics (2,2,1), (2,2,2), (2,3,1), (2,3,2), (4,2,1), (4,2,2), (4,3,1), (4,3,2)
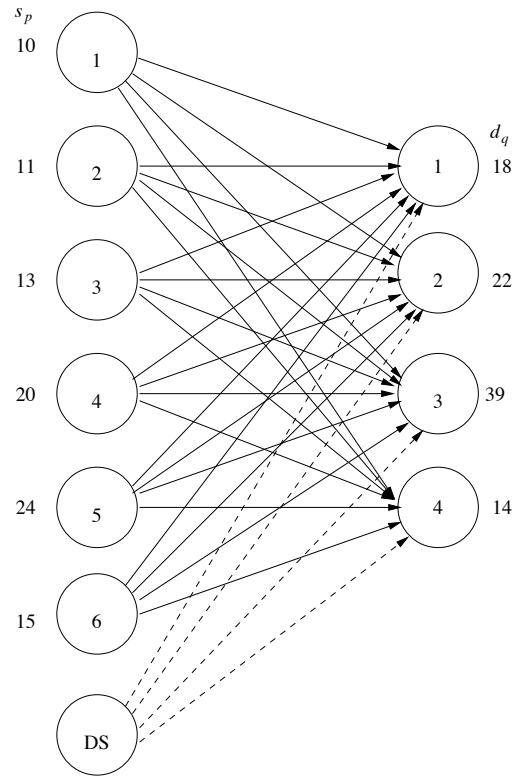
(a)

C_{pq}	1	2	3	4
1	10	15	17	19
2	12	18	16	18
3	13	12	13	20
4	8	16	14	21
5	14	19	10	12
6	19	20	18	13
DS	15637	15637	15637	15637

(b)

Figure 5.1: Transportation Network for the Illustrative Example.
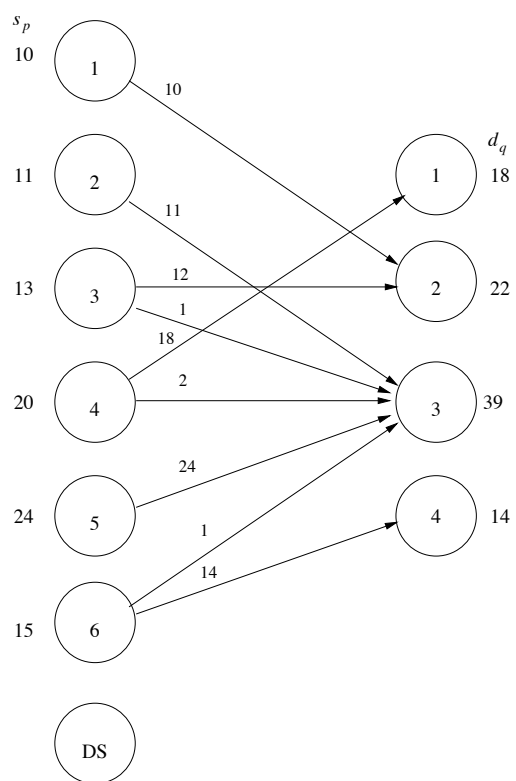
Figure 5.2: Optimal Solution for Model TDSS1.
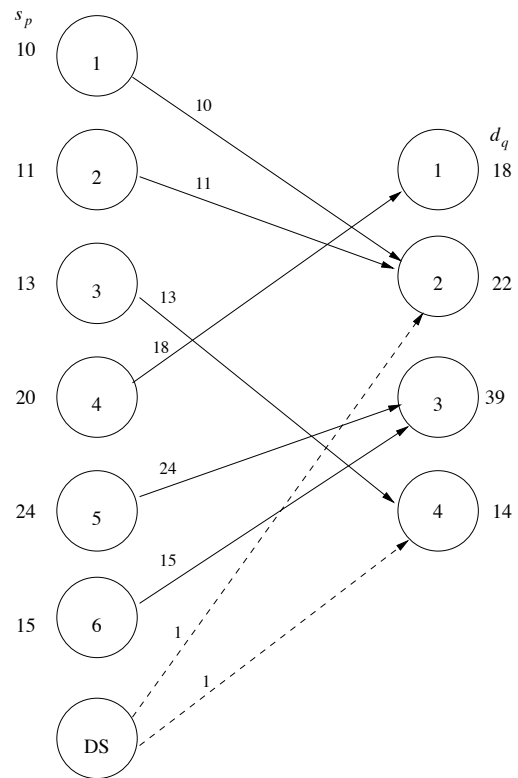
Figure 5.3: Optimal Solution for Model TDSS2.
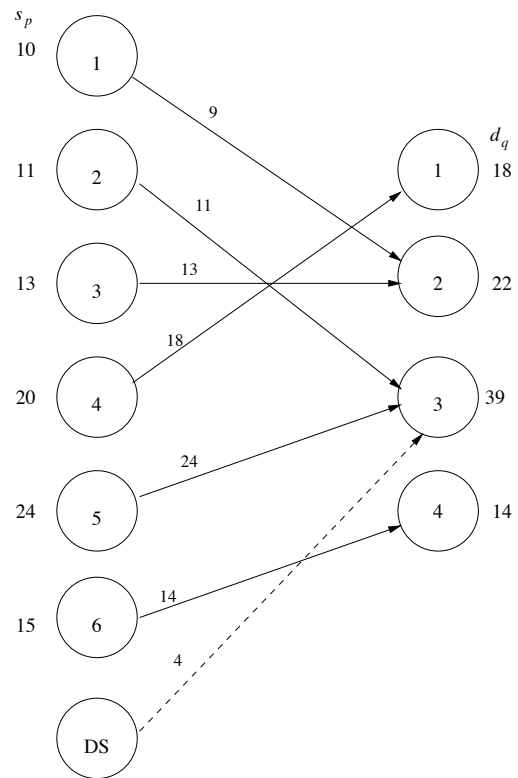
Figure 5.4: Optimal Solution for Second Variant of Model TDSS2.

Table 5.1: Results for the Heuristic Procedures on Variant 1 of the Illustrative Example.

	Heuristic Procedure	Total Penalized Cost	Total Real Car-days	Total Shortage
1	(1,∅,1)	501189	805	32
2	(1,∅,2)	501189	805	32
3	(2,1,1)	110514	1055	7
4	(2,1,2)	110514	1055	7
5	(2,2,1)	63581	1033	4
6	(2,2,2)	63581	1033	4
7	(2,3,1)	63581	1033	4
8	(2,3,2)	63581	1033	4
9	(3,1,1)	110514	1055	7
10	(3,1,2)	110514	1055	7
11	(3,2,1)	63581	1033	4
12	(3,2,2)	63581	1033	4
13	(3,3,1)	63581	1033	4
14	(3,3,2)	63581	1033	4
15	(4,1,1)	110514	1055	7
16	(4,1,2)	110514	1055	7
17	(4,2,1)	63581	1033	4
18	(4,2,2)	63581	1033	4
19	(4,3,1)	63581	1033	4
20	(4,3,2)	63581	1033	4
21	LP-(1,∅,1)	110514	1055	7
22	LP-(1,∅,2)	63581	1033	4
23	NET-(1,∅,1)	344902	888	22
	CPLEX-MIP	32536	1262	2

and LP-(1,∅,2). Figure 5.6 depicts the corresponding flow distribution for this solution. Note that the total shortage is given by 4, with node 3 incurring a shortage. The resulting total penalized cost is 40157, and the corresponding car-days for the real delivered cars is 1033. In this case, the solution obtained matches exactly with the optimal solution for this variant.

5.2   Description of Test Problems

We now consider a test-bed of five realistic problems composed by TTX based on historical data pertaining to different equipment type repositioning scenarios. Table 5.3 provides a summary of the size characteristics of these problems.
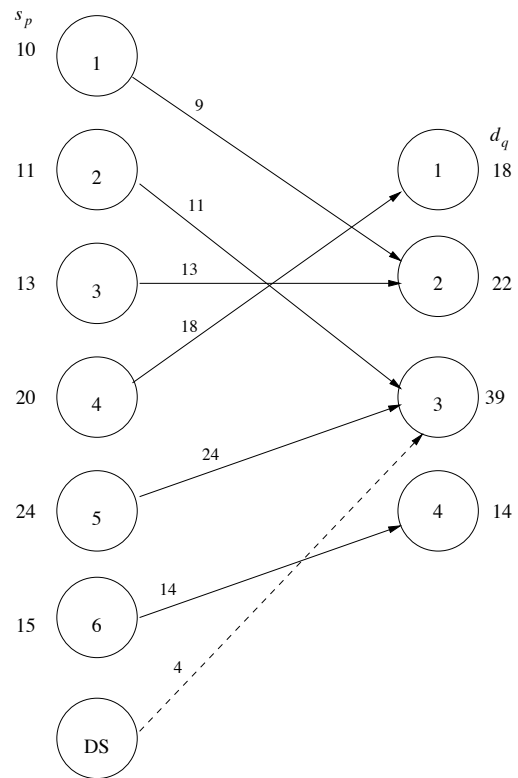
Figure 5.5: Best Heuristic Solution Found for Variant 1 of the Illustrative Example.

Table 5.2: Results for the Heuristic Procedures on Variant 2 of the Illustrative Example.

	Heuristic Procedure	Total Penalized Cost	Total Real Car-days	Total Shortage
1	(1,∅,1)	177117	1059	18
2	(1,∅,2)	177117	1059	18
3	(2,1,1)	69522	1055	7
4	(2,1,2)	69522	1055	7
5	(2,2,1)	40157	1033	4
6	(2,2,2)	40157	1033	4
7	(2,3,1)	40157	1033	4
8	(2,3,2)	40157	1033	4
9	(3,1,1)	128202	1049	13
10	(3,1,2)	128202	1049	13
11	(3,2,1)	108732	1141	11
12	(3,2,2)	108732	1141	11
13	(3,3,1)	108732	1141	11
14	(3,3,2)	108732	1141	11
15	(4,1,1)	69522	1055	7
16	(4,1,2)	69522	1055	7
17	(4,2,1)	40157	1033	4
18	(4,2,2)	40157	1033	4
19	(4,3,1)	40157	1033	4
20	(4,3,2)	40157	1033	4
21	LP-(1,∅,1)	69522	1055	7
22	LP-(1,∅,2)	40157	1033	4
23	NET-(1,∅,1)	177024	966	18
	CPLEX-MIP	40157	1033	4
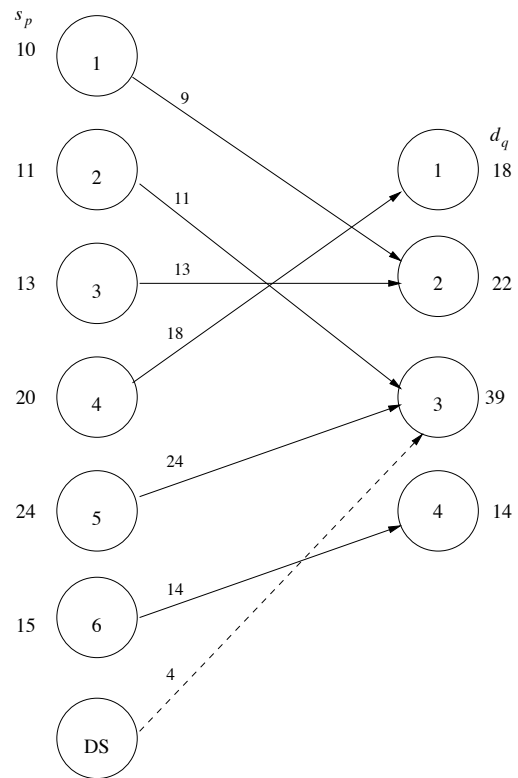
Figure 5.6: Best Heuristic Solution Found for Variant 2 of the Illustrative Example.

Table 5.3: Size Characteristics of the Test Problems.

Test Problem	# of supplies	# of demands	# of arcs	# of groups	# of blocking variables (y)	# of continuous variables (x)
TTX1	424	66	5495	1822	1822	5495
TTX2	1057	417	65876	20290	20290	65876
TTX3	1144	444	75490	22350	22350	75490
TTX4	1200	432	87706	25172	25172	87706
TTX5	1362	530	146970	35626	35626	146970

All runs have been made on a Sun Ultra-1 Sparcstation, with 163 MHz clock speed and 64 MB RAM. A set of parallel runs made on a Pentium-Pro, 200 MHz, 512 RAM PC yields cpu times that are significantly faster, yielding an average fraction of 0.53 with respect to the run-times shown for the Sparcstation runs. Since the 5 minute run-time limit applies to the Pentium-Pro environment, we can permit about a 9 minute run-time on the Sun Ultra-1 Sparcstation. *The times reported for all the runs below pertain to the Pentium-Pro environment in use at TTX.*

5.3 Computational Experience Using Realistic Test Problems

Tables 5.4-5.8 present the results obtained using the 23 heuristic procedures for each of the respective five test problems described in Table 5.3. Also shown therein is the performance of CPLEX-MIP as applied to these test problems. The various columns displayed in Tables 5.4 - 5.8 have the following detailed interpretation.

Cumulative Penalized Cost

This set of columns represents the accumulated penalized cost given by the objective function of Problem TDSS2. This cost is broken down as that pertaining to the flows into demand nodes for the first (imminent) day, the first plus the second day, and finally, for the entire horizon (total cost). The scaling notation of E9 represents 10^9.

Cumulative Real Car-days

This column represents the accumulated real car-days given by the first term $\sum_p \sum_q C_{pq} x_{pq}$ in the objective function of Problem TDSS2. This computation is decomposed into the part of this total that pertains to flows into demand nodes corresponding to the first day, the first plus the second day, in addition to providing the total quantity. Thus, this measure excludes the penalty cost associated with the shipments from the dummy supply node to

the demand nodes. The scaling notation follows a similar convention as that used for the cumulative penalized cost.

Cumulative Shortage

This set of columns represents the accumulated railcar shortages given by the total flow from the dummy source to the demand nodes pertaining to the first day, the first plus the second day, and finally, to all the demand nodes.

CPU time

This column records the performance measure to solve the problem through the specified procedures in terms of the CPU time in minutes on the Pentium-Pro computer.

Observe from the results of Tables 5.4-5.8 that except for the 5,000 arc problem, the use of CPLEX-MIP is untenable with respect to the solution time restriction that has been imposed in practice. Actually, with some experimentation, we have ascertained that many problems having up to 8,000 arcs are CPLEX-MIP-solvable to optimality (see Table 5.9 for the results of 5 instances of problems with 8,000 arcs). For those problems that take more than 3 minutes for CPLEX-MIP to solve, a timer is set to activate the switching from CPLEX-MIP to one of the heuristic procedures once the time has elapsed. For problems having more than 8,000 arcs, we must apply one of the heuristic procedures (except perhaps for certain overnight runs that the user might make). Among the tested heuristic procedures, Blocking Heuristic procedures (DS, VF, DR) $\equiv$ (4,3,2) and (2,3,2) appear to be the two candidates with superior performance with respect to the quality of the solution produced, while containing effort within the prespecified limitation of 5 minutes of cpu time. In fact, an examination of the differences in performances between (2,3,2) and (4,3,2) provides an indication of the benefits of the "look-ahead" feature of the model for some data sets, considering that DS = 2 examines nodes pertaining to each time period individually, while DS = 4 examines demand nodes two periods at a time. Furthermore, note that for all the problems TTX1 - TTX5, the heuristic combination (4,3,2) uses a depth of search set Q equal to the demand nodes comprising the (next) two time periods in the horizon, given the limit of 80 on $|Q|$. However, for the test problem TTX5, for example, the length of the list for the first two time periods is 132. Hence, if the preset limit on $|Q|$ is 40, for example, we would section $\mathcal{D}$ into sets Q of cardinality 40. To illustrate the performance of the Blocking Heuristic (4,3,2) for this test problem when using a limit of 40 on $|Q|$ (call this strategy (**4(40), 3, 2**)) we present the corresponding results obtained in Table 5.10. Also shown in this table is the performance of another variant of this heuristic where for the first loop, we employ Depth of Search Strategy 4(∞) (meaning that the limit on $|Q|$ is ∞), but then for the next selection of Q, we switch over to the Depth of Search Strategy 2. This modified Blocking Heuristic combination is referred to as (**4(∞)-2,3,2**). Observe that (4(40),3,2) yields the best quality solution, followed by the strategies (4(∞)-2,3,2) and (4(80),3,2), in that order, for this test problem. Moreover, the solution time required by the strategies (4(40),3,2), (4(80),3,2) and (4(∞)-2,3,2) are respectively, 2.12, 2.96, and 2.99 cpu minutes. Although the software developed employs the strategy (4(80),3,2) as the default strategy, the other

Table 5.4: Computational Results for Problem TTX1.

	Heuristic Procedure	Cumulative Penalized Cost			Cumulative Real Car-days			Cumulative Shortage			CPU time (mins.)
		1st day (E9)	2nd day (E9)	Total (E9)	1st day	2nd day	Total	1st day	2nd day	Total	
1	(1,∅,1)	0.94	1.09	1.11	34	83	359	3	7	11	0.04
2	(1,∅,2)	0.94	1.09	1.11	34	83	359	3	7	11	0.04
3	(2,1,1)	0.90	1.03	1.04	34	85	370	3	6	9	0.04
4	(2,1,2)	0.90	1.03	1.04	34	85	370	3	6	9	0.04
5	(2,2,1)	0.90	1.02	1.04	34	85	368	3	6	9	0.04
6	(2,2,2)	0.90	1.02	1.04	34	85	368	3	6	9	0.04
7	(2,3,1)	0.90	1.02	1.03	33	85	367	3	6	9	0.04
8	(2,3,2)	0.90	1.02	1.03	33	85	367	3	6	9	0.04
9	(3,1,1)	0.90	1.04	1.05	33	79	357	3	7	11	0.04
10	(3,1,2)	0.90	1.04	1.05	33	79	357	3	7	11	0.04
11	(3,2,1)	0.90	1.04	1.05	33	79	357	3	7	11	0.04
12	(3,2,2)	0.90	1.04	1.05	33	79	357	3	7	11	0.04
13	(3,3,1)	0.90	1.04	1.05	33	79	357	3	7	11	0.04
14	(3,3,2)	0.90	1.04	1.05	33	79	357	3	7	11	0.04
15	(4,1,1)	0.91	1.06	1.07	36	83	358	3	7	11	0.04
16	(4,1,2)	0.91	1.06	1.07	36	83	358	3	7	11	0.04
17	(4,2,1)	0.91	1.06	1.07	36	85	357	3	7	11	0.04
18	(4,2,2)	0.91	1.06	1.07	36	85	357	3	7	11	0.04
19	(4,3,1)	0.90	1.02	1.03	34	84	366	3	6	9	0.05
20	(4,3,2)	0.90	1.02	1.03	34	84	366	3	6	9	0.05
21	LP-(1,∅,1)	0.89	1.01	1.03	32	84	441	3	6	9	0.08
22	LP-(1,∅,2)	0.89	1.01	1.03	32	84	441	3	6	9	0.08
23	NET-(1,∅,1)	0.89	1.01	1.03	32	83	365	3	6	9	0.03
	CPLEX-MIP	0.89	1.01	1.03	33	84	366	3	6	9	0.09

Table 5.5: Computational Results for Problem TTX2.

	Heuristic Procedure	Cumulative Penalized Cost			Cumulative Real Car-days			Cumulative Shortage			CPU time (mins.)
		1st day (E10)	2nd day (E10)	Total (E10)	1st day	2nd day	Total	1st day	2nd day	Total	
1	(1,∅,1)	8.62	9.81	10.17	3187	6528	23447	125	203	881	0.49
2	(1,∅,2)	8.62	9.81	10.17	3185	6525	23583	125	203	851	0.48
3	(2,1,1)	9.24	10.38	10.75	3205	6525	22818	139	203	919	0.54
4	(2,1,2)	9.24	10.37	10.74	3198	6489	22826	139	203	925	0.54
5	(2,2,1)	9.23	10.47	10.84	3239	6155	22880	135	270	925	0.55
6	(2,2,2)	9.23	10.45	10.82	3234	6189	22871	135	258	915	0.55
7	(2,3,1)	7.87	9.21	9.55	3224	6103	23619	84	232	704	1.14
8	(2,3,2)	7.87	9.22	9.55	3224	6104	24045	84	232	654	1.16
9	(3,1,1)	8.60	9.81	10.17	3147	6400	23523	125	222	857	0.49
10	(3,1,2)	8.60	9.81	10.18	3137	6357	23656	125	222	840	0.48
11	(3,2,1)	8.60	9.81	10.17	3147	6400	23523	125	222	857	0.48
12	(3,2,2)	8.60	9.81	10.18	3137	6357	23656	125	222	840	0.48
13	(3,3,1)	8.60	9.81	10.16	3147	6418	23503	125	219	865	0.53
14	(3,3,2)	8.60	9.80	10.17	3138	6383	23623	125	218	836	0.53
15	(4,1,1)	9.24	10.45	10.77	3219	6322	23491	139	251	825	0.54
16	(4,1,2)	9.24	10.45	10.78	3217	6339	23560	139	250	821	0.54
17	(4,2,1)	9.24	10.54	10.86	3239	6087	23255	135	294	847	0.54
18	(4,2,2)	9.24	10.51	10.84	3239	6131	23422	135	282	819	0.54
19	(4,3,1)	7.87	9.17	9.50	3218	6298	24346	84	220	596	1.16
20	(4,3,2)	7.87	9.09	9.43	3219	6424	24338	84	180	604	1.15
21	LP-(1,∅,1)	9.45	10.76	11.15	3520	7235	24463	110	175	926	1.11
22	LP-(1,∅,2)	9.46	10.77	11.16	3506	7157	24175	110	175	963	1.10
23	NET-(1,∅,1)	8.20	9.33	9.68	3392	6813	23420	80	137	865	0.72
	CPLEX-MIP	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	≫ 5.00

Table 5.6: Computational Results for Problem TTX3.

	Heuristic Procedure	Cumulative Penalized Cost			Cumulative Real Car-days			Cumulative Shortage			CPU time (mins.)
		1st day (E9)	2nd day (E9)	Total (E9)	1st day	2nd day	Total	1st day	2nd day	Total	
1	$(1,\emptyset,1)$	5.36	6.48	6.68	315	710	2424	3	6	10	0.53
2	$(1,\emptyset,2)$	5.36	6.48	6.68	315	710	2424	3	6	10	0.53
3	$(2,1,1)$	4.79	5.79	5.95	307	692	2358	3	5	9	0.59
4	$(2,1,2)$	4.79	5.79	5.95	307	692	2358	3	5	9	0.59
5	$(2,2,1)$	4.81	5.81	5.98	301	689	2365	3	5	9	0.59
6	$(2,2,2)$	4.81	5.81	5.98	301	689	2365	3	5	9	0.59
7	$(2,3,1)$	3.77	4.69	4.85	290	666	2321	2	4	8	1.04
8	$(2,3,2)$	3.77	4.69	4.85	290	666	2319	2	4	8	1.04
9	$(3,1,1)$	5.31	6.37	6.56	315	701	2399	3	6	11	0.54
10	$(3,1,2)$	5.31	6.37	6.56	315	701	2399	3	6	11	0.54
11	$(3,2,1)$	5.12	6.19	6.38	310	699	2397	3	6	11	0.54
12	$(3,2,2)$	5.12	6.19	6.38	310	699	2397	3	6	11	0.54
13	$(3,3,1)$	4.94	5.94	6.13	300	679	2370	4	7	13	0.60
14	$(3,3,2)$	4.94	5.94	6.13	300	678	2371	4	7	13	0.60
15	$(4,1,1)$	4.87	5.87	6.05	309	689	2373	3	5	10	0.59
16	$(4,1,2)$	4.87	5.87	6.05	307	684	2368	3	5	10	0.59
17	$(4,2,1)$	5.11	6.13	6.31	303	683	2360	4	7	12	0.59
18	$(4,2,2)$	5.11	6.13	6.31	303	681	2360	4	7	12	0.59
19	$(4,3,1)$	3.77	4.68	4.85	289	675	2320	2	4	10	1.01
20	$(4,3,2)$	3.77	4.68	4.85	289	675	2320	2	4	10	1.01
21	LP-$(1,\emptyset,1)$	6.55	7.91	8.18	370	823	2957	1	3	11	1.09
22	LP-$(1,\emptyset,2)$	6.55	7.94	8.20	364	830	2951	1	3	8	1.03
23	NET-$(1,\emptyset,1)$	3.72	4.66	4.82	290	680	2331	1	3	9	0.96
	CPLEX-MIP	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	$\gg 5.00$

Table 5.7: Computational Results for Problem TTX4.

Heuristic Procedure		Cumulative Penalized Cost			Cumulative Real Car-days			Cumulative Shortage			CPU time (mins.)
		1st day (E9)	2nd day (E9)	Total (E9)	1st day	2nd day	Total	1st day	2nd day	Total	
1	$(1,\emptyset,1)$	3.94	4.80	4.93	298	657	2356	4	10	17	0.62
2	$(1,\emptyset,2)$	3.94	4.79	4.93	298	652	2345	4	10	16	0.68
3	$(2,1,1)$	3.52	4.27	4.38	289	640	2304	4	8	14	0.70
4	$(2,1,2)$	3.52	4.26	4.38	289	640	2304	4	8	16	0.70
5	$(2,2,1)$	3.44	4.18	4.30	287	642	2297	4	8	16	0.69
6	$(2,2,2)$	3.44	4.18	4.30	287	642	2289	4	8	16	0.69
7	$(2,3,1)$	3.08	3.79	3.90	288	634	2285	2	7	14	1.27
8	$(2,3,2)$	3.08	3.79	3.90	289	630	2280	2	7	14	1.29
9	$(3,1,1)$	3.89	4.74	4.88	298	654	2353	4	10	17	0.63
10	$(3,1,2)$	3.89	4.74	4.87	297	649	2344	4	10	17	0.63
11	$(3,2,1)$	3.89	4.74	4.88	298	654	2353	4	10	17	0.66
12	$(3,2,2)$	3.89	4.74	4.87	297	649	2344	4	10	17	0.63
13	$(3,3,1)$	3.89	4.69	4.82	299	652	2356	4	10	17	0.67
14	$(3,3,2)$	3.89	4.69	4.82	298	648	2343	4	10	17	0.67
15	$(4,1,1)$	3.64	4.42	4.54	292	657	2362	4	7	10	0.69
16	$(4,1,2)$	3.64	4.41	4.53	292	658	2344	4	7	10	0.75
17	$(4,2,1)$	3.55	4.34	4.45	289	656	2338	4	7	12	0.73
18	$(4,2,2)$	3.55	4.33	4.45	288	656	2333	4	7	12	0.72
19	$(4,3,1)$	3.21	3.95	4.05	290	665	2326	2	5	10	1.36
20	$(4,3,2)$	3.21	3.95	4.05	291	664	2315	2	5	7	1.28
21	LP-$(1,\emptyset,1)$	7.19	8.66	8.93	391	873	3139	4	10	23	1.31
22	LP-$(1,\emptyset,2)$	7.49	8.89	9.17	406	880	3138	4	10	23	1.18
23	NET-$(1,\emptyset,1)$	3.48	4.21	4.32	302	673	2370	2	6	7	1.16
CPLEX-MIP		N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	$\gg 5.00$

Table 5.8: Computational Results for Problem TTX5.

Heuristic Procedure		Cumulative Penalized Cost			Cumulative Real Car-days			Cumulative Shortage			CPU time (mins.)
		1st day (E11)	2nd day (E11)	Total (E11)	1st day	2nd day	Total	1st day	2nd day	Total	
1	$(1,\emptyset,1)$	1.25	1.62	1.75	4324	8291	35335	80	289	1148	1.14
2	$(1,\emptyset,2)$	1.25	1.62	1.74	4313	8319	36077	80	278	1012	1.21
3	$(2,1,1)$	1.16	1.50	1.62	4236	8259	35722	51	220	869	1.29
4	$(2,1,2)$	1.16	1.50	1.62	4236	8261	36401	51	220	799	1.44
5	$(2,2,1)$	1.16	1.51	1.63	4274	8059	35596	51	261	910	1.28
6	$(2,2,2)$	1.16	1.51	1.63	4255	8074	35925	51	262	864	1.31
7	$(2,3,1)$	1.01	1.34	1.45	4271	8309	36732	19	155	670	3.03
8	$(2,3,2)$	1.01	1.33	1.44	4293	8265	37393	19	155	551	2.83
9	$(3,1,1)$	1.25	1.62	1.75	4323	8223	35475	80	301	1146	1.15
10	$(3,1,2)$	1.25	1.62	1.74	4313	8319	35892	80	278	1021	1.13
11	$(3,2,1)$	1.25	1.62	1.75	4323	8223	35475	80	301	1146	1.28
12	$(3,2,2)$	1.25	1.62	1.75	4313	8319	35892	80	278	1021	1.15
13	$(3,3,1)$	1.23	1.62	1.74	4278	8190	35326	80	307	1136	1.30
14	$(3,3,2)$	1.23	1.61	1.74	4282	8326	36047	80	279	991	1.23
15	$(4,1,1)$	1.17	1.52	1.63	4277	8238	36129	51	241	841	1.29
16	$(4,1,2)$	1.17	1.52	1.63	4276	8238	36337	51	241	832	1.36
17	$(4,2,1)$	1.16	1.51	1.63	4283	8215	35785	51	238	894	1.29
18	$(4,2,2)$	1.16	1.51	1.63	4265	8214	36252	51	238	855	1.28
19	$(4,3,1)$	1.05	1.40	1.51	4317	8256	36997	39	208	700	2.96
20	$(4,3,2)$	1.05	1.39	1.51	4318	8266	36871	39	208	735	2.96
21	LP-$(1,\emptyset,1)$	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	$\gg 5.00$
22	LP-$(1,\emptyset,2)$	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	$\gg 5.00$
23	NET-$(1,\emptyset,1)$	1.07	1.40	1.52	4367	8459	35387	29	172	1079	1.69
CPLEX-MIP		N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	$\gg 5.00$

Table 5.9: CPLEX-MIP Results for 5 instances of Problems with 8,000 arcs.

Test Problem	Cumulative Penalized Cost			Cumulative Real Car-days			Cumulative Shortage			CPU time (mins.)
	1st day (E10)	2nd day (E10)	Total (E10)	1st day	2nd day	Total	1st day	2nd day	Total	
1	4.75	10.55	17.69	379	921	2538	212	639	2044	1.39
2	11.78	17.05	23.24	476	747	1931	512	890	2419	0.58
3	5.40	8.38	15.16	426	864	1992	238	499	1946	1.69
4	8.43	13.36	17.54	744	1002	2595	375	718	1713	1.44
5	10.44	16.98	22.49	483	827	2700	476	458	2324	0.75

Table 5.10: Computational Results for Problem TTX5 Using Variants of Blocking Heuristic (4,3,2).

Heuristic Procedure		Cumulative Penalized Cost			Cumulative Real Car-days			Cumulative Shortage			CPU time (mins.)
		1st day (E11)	2nd day (E11)	Total (E11)	1st day	2nd day	Total	1st day	2nd day	Total	
1	(4(40),3,2)	1.03	1.37	1.50	4212	8173	36697	34	202	769	2.12
2	(4(80),3,2)	1.05	1.39	1.51	4318	8266	36871	39	208	735	2.96
3	(4(∞)-2,3,2)	1.10	1.39	1.50	4325	8328	36745	57	169	659	2.99

foregoing variants are also available for implementation. In summary, we recommend the user to experiment with (2,3,2) and (4,3,2) (with its aforementioned variants) using real data to select a strategy for final implementation.

5.4   Exploration of Some Special System Features

5.4.1   Results for Model TDSS1

In this section, we present results that illustrate certain special features of our model. These include the effects of the blocking restrictions, the effect of the time-based priority structure, and the effect of the penalty parameter θ (see Equation (3.30)).

Table 5.11: Results for Model TDSS1.

Test Problem	Cumulative Penalized Cost			Cumulative Real Car-days			Cumulative Shortage			CPU time (mins.)	# of Blocking Violations
	1st day (E9)	2nd day (E9)	Total (E9)	1st day	2nd day	Total	1st day	2nd day	Total		
1 TTX1	0.89	1.01	1.03	32	83	365	3	6	9	0.03	0
2 TTX2	74.36	84.92	88.06	3305	6668	25681	64	109	368	0.49	112
3 TTX3	3.46	4.31	4.45	288	665	2343	1	3	4	0.71	29
4 TTX4	2.58	3.17	3.25	281	648	2284	2	5	5	0.83	38
5 TTX5	93.93	121.65	131.09	4144	7877	37845	19	147	316	1.18	142

First, let us examine the effect of removing the blocking restrictions, i.e., let us run Model TDSS1 on the five test problems of Table 5.3. The results obtained are presented in Table 5.11. Also shown here is the number of source nodes at which the blocking constraints of Model TDSS2 are violated by the corresponding solution to Model TDSS1. Note that the optimal solutions to TDSS1 are obtained relatively quickly, and are naturally superior in quality to the solutions of Model TDSS2, except for the violation in the blocking constraints. Hence, the user might wish to run Model TDSS1 to assess the acceptability of the violations in the blocking constraints versus the accompanying reductions in latenesses and shortages. Note that the NETOPT-based heuristic that attempts to perturb the solution to TDSS1 in order to satisfy the blocking restrictions provides reasonably good, although not the best, results among the various heuristic procedures.

5.4.2 Effect of the Time-Based Priority Matrix PRMAT

Another feature that we advise the user to reflect on is the time priority hierarchy employed in the model. Recall that the default strategy is to use TIMEPR = 1 which corresponds to the level table depicted in Figure 3.1. Note that by this structure, a preemptively high priority is ascribed to the imminent time period, to the extent that a one day late delivery to an imminent period demand node is considered worse than having a shortage at a demand node corresponding to the next time period. However, the user has the option of selecting other TIMEPR values, where the associated priority structures are described in Section 3.2.1, or to employ a customized priority matrix. For example, with TIMEPR = 2, the highest priority is to minimize shortages occurring over the various time periods, with a greater emphasis on reducing shortages for earlier time periods, before considering the reduction in lateness as per the default hierarchy. To illustrate the effect of PRMAT on the latenesses

Table 5.12: Effect of PRMAT for TTX1 on Latenesses and Shortages.

| TIMEPR | Imminent Period | | | | Second Period | | | |
	1-day late	2-day late	≥ 3-days late	Shortages	1-day late	2-day late	≥ 3-days late	Shortages
1	4	0	0	3	0	0	0	3
2	3	0	0	3	0	0	0	3
3	3	0	0	3	0	0	0	3
4	3	0	0	3	0	0	0	3

Table 5.13: Effect of PRMAT for TTX3 on Latenesses and Shortages.

| TIMEPR | Imminent Period | | | | Second Period | | | |
	1-day late	2-day late	≥ 3-days late	Shortages	1-day late	2-day late	≥ 3-days late	Shortages
1	16	0	0	1	6	3	0	2
2	16	0	0	1	6	3	0	2
3	17	0	0	1	6	3	0	2
4	12	0	0	1	4	3	0	2

and shortages pertaining to the first two time periods, we ran problems TTX1 and TTX3 using different TIMEPR values. For TTX1, we compute the exact solution via CPLEX-MIP, but for TTX3, we have employed the heuristic (4,3,2). The results obtained are presented in Tables 5.12 and 5.13. Notice the compromises made between the various latenesses and shortages over the first two demand time periods by the different time priority structures. Part of this difference is governed by the magnitude of θ used for these runs - see the next section. Notice that since θ is not too large, TIMEPR = 4 makes a greater distinction between relative latenesses within a particular time period, than does TIMEPR = 1. While we prescribe the default strategy TIMEPR = 1, the user might wish to experiment with TIMEPR = 4, in particular, to explore alternative possible railcar distribution schemes. For the sake of illustration, we also provide a comparison between the blocking heuristics (2,3,2) and (4,3,2) for TIMEPR=4, using the test cases TTX1-TTX5. Table 5.14 provides these results. (Note that since the penalized objective functions are different for TIMEPR equal to 1 and 4, it is meaningful only to compare the cumulative real car-days and shortages.)

Table 5.14: Comparison of Blocking Heuristics (2,3,2) and (4,3,2) for TIMEPR=4.

Problem and Procedure		Cumulative Penalized Cost			Cumulative Real Car-days			Cumulative Shortage			CPU time (mins.)
		1st day (E9)	2nd day (E9)	Total (E9)	1st day	2nd day	Total	1st day	2nd day	Total	
TTX1	(2,3,2)	0.63	1.01	1.23	33	85	367	3	6	9	0.04
TTX1	(4,3,2)	0.63	1.01	1.23	34	84	366	3	6	9	0.05
TTX2	(2,3,2)	16.46	27.14	49.09	3367	6782	25411	69	135	484	1.18
TTX2	(4,3,2)	17.22	25.49	47.85	3429	6811	25652	72	124	430	1.89
TTX3	(2,3,2)	0.47	0.81	1.03	294	676	2333	2	4	7	1.05
TTX3	(4,3,2)	0.48	0.81	1.04	308	682	2335	2	4	7	1.59
TTX4	(2,3,2)	0.45	1.20	1.64	294	638	2296	2	7	13	1.28
TTX4	(4,3,2)	0.48	0.64	0.74	303	684	2351	2	3	5	1.92
TTX5	(2,3,2)	5.53	28.23	51.50	4535	8647	38242	19	152	445	2.96
TTX5	(4(40),3,2)	10.16	37.92	69.79	4245	8373	37546	38	202	692	2.12

5.4.3 Effect of the Penalty Parameter θ

Recall from Chapter 3 that to control the hierarchy between the time-based and the demand location-based prioritized objective functions, we defined a parameter θ whose value is determined by Equation (3.30). Note that for all our problems, given the number of levels in the problem, the actual value of θ used was less than 2. Hence, we would obtain identical results for $\theta_{max} \geq 2$. Consequently, in this section, we experiment with using 1.25, 1.50, and 2.00 as the upper bound on θ in Equation (3.30) (call this θ_{max}), for both Models TDSS1 and TDSS2. Table 5.15 presents the results obtained for Model TDSS1 using the first four problems of Table 5.3, while Tables 5.16 and 5.17 present corresponding results obtained for TDSS2 using the two test cases TTX1 and TTX3, along with the exact solution obtained via CPLEX-MIP (for TTX1) and the heuristic solution obtained via the Blocking Heuristic (4,3,2).

Since we would prefer to make a stronger distinction between the objectives at the various levels, and since the main purpose of using an extended horizon is to provide a look-ahead for the imminent period, we suggest that the user also attempt a horizon of 3 or 4 days, unless there exist certain essential routes that require long transit times relative to the horizon. This would reduce the number of levels and provide a greater distinction between the hierarchy of objective functions by permitting a larger value of θ.

Table 5.15: Effect of θ_{max} on TDSS1.

Problem	θ_{max}	θ of Eq. 3.30	Cumulative Penalized Cost			Cumulative Real Car-days			Cumulative Shortage			CPU time (mins.)
			1st day (E7)	2nd day (E7)	Total (E7)	1st day	2nd day	Total	1st day	2nd day	Total	
	1.25	1.250	.1	0.1	0.1	39	90	365	3	6	9	0.03
TTX1	1.50	1.500	4.5	5.6	5.8	34	83	365	3	6	9	0.03
	2.00	1.707	88.6	101.4	102.7	32	83	365	3	6	9	0.03
	1.25	1.250	44.2	56.7	69.2	3401	6665	25116	105	170	466	0.49
TTX2	1.50	1.500	2304.0	2707.4	2860.8	3304	6695	25757	64	109	358	0.49
	2.00	1.582	7435.6	8491.5	8806.3	3305	6668	25681	64	109	368	0.49
	1.25	1.250	2.4	3.5	4.0	306	671	2320	1	4	8	0.60
TTX3	1.50	1.500	115.5	149.8	157.0	291	668	2343	1	3	4	0.69
	2.00	1.578	345.7	431.5	445.2	288	665	2343	1	3	4	0.71
	1.25	1.250	1.7	2.3	2.6	300	678	2309	2	2	2	0.75
TTX4	1.50	1.500	83.5	106.1	110.2	282	647	2292	2	4	4	0.79
	2.00	1.580	258.1	317.0	325.1	281	648	2284	2	5	5	0.81

Table 5.16: Effect of θ_{max} on TDSS2 for TTX1.

Procedure	θ_{max}	θ of Eq. 3.30	Cumulative Penalized Cost			Cumulative Real Car-days			Cumulative Shortage			CPU time (mins.)
			1st day (E6)	2nd day (E6)	Total (E6)	1st day	2nd day	Total	1st day	2nd day	Total	
	1.25	1.250	0.8	1.1	1.2	40	91	366	3	6	9	0.05
CPLEX-MIP	1.50	1.500	45.4	56.1	58.3	34	84	366	3	6	9	0.05
	2.00	1.707	885.9	1013.6	1027.8	33	84	366	3	6	9	0.05
	1.25	1.250	0.8	1.1	1.2	40	88	366	3	6	9	0.05
BH(4,3,2)	1.50	1.500	45.4	56.1	58.3	34	84	366	3	6	9	0.05
	2.00	1.707	895.1	1017.0	1031.3	34	84	366	3	6	9	0.05

Table 5.17: Effect of θ_{max} on TDSS2 for TTX3.

Procedure	θ_{max}	θ of Eq. 3.30	Cumulative Penalized Cost			Cumulative Real Car-days			Cumulative Shortage			CPU time (mins.)
			1st day (E6)	2nd day (E6)	Total (E6)	1st day	2nd day	Total	1st day	2nd day	Total	
	1.25	1.250	25.9	37.1	44.5	311	670	2303	2	5	12	1.10
BH(4,3,2)	1.50	1.500	1251.7	1620.8	1705.4	289	672	2324	2	4	10	1.09
	2.00	1.578	3773.0	4684.0	4847.9	289	675	2320	2	4	10	1.01

Chapter 6

Finale

Besides providing a tool to aid in the decision-making process, a principal utility of the developed system is that it provides the opportunity to conduct various *what-if* analyses. The effects of many of the practical considerations that have been incorporated in TDSS2 can be studied via such various sensitivity analyses. In this chapter, we present a brief discussion of these what-if user interfaces that have been implemented. Within the model itself, however, some strategic selection of a heuristic or exact procedure needs to be made so that the solution approach becomes viable and effective for these studies. Accordingly, based on the performance of the heuristics as observed in Chapter 5, we also prescribe an overall system implementation flowchart. Lastly, after a brief summary and conclusions of our work, some future research directions are presented in this chapter.

6.1 The Implemented Interface

The significance of our research becomes apparent when we consider the scope of the decision support that our model offers. Using the developed tactical tool to aid in the railcar distribution problem in a flexible, interactive fashion, the management team has the opportunity to find an appropriate decision that maximizes the team's short-term utility. In general, there are seven choice-modes that have been incorporated into the user interface of the model. These are addressed in turn below.

6.1.1 What-if on Demands

In this choice-mode, a user is able to view and modify daily forecast demand quantities. The default demand horizon is for seven days. The starting demand day is based on the current day plus two days. This mode allows the user to update daily forecasted demand for either

a specific demand point or for a group of demand locations.

6.1.2   What-if on Supplies

Besides allowing the user to view and modify daily available supplies of empty railcars, this mode also permits the updating of supply by day for a specific supply point or for a group of supply locations.

6.1.3   What-if on Empty Routes

This mode is useful for activating or deactivating selected empty routes by accessing the internal *Route Days* table. If a route is inactivated, however, the system would still ensure that there is at least one alternate active route that is available.

6.1.4   What-if on Diversions

In this mode, the user could view and divert railcars that are assigned to an empty trip. One or more railcars can be diverted by selecting the individual railcar(s), and providing a new destination location, the serving railroad, and the expected time of arrival at the new demand location. Railcar diversion naturally impacts the forecasted demand at both the original and new destination demand locations.

6.1.5   What-if on Storage Issues

The user could release one or more railcars that are in storage or have an inactive status. These railcars can be made available for supply at their current supply location, or at a different supply location, or at some demand location. The last option can be used if the user desires the railcars to be dispositioned to satisfy a specific demand.

6.1.6   What-if on Control Orders

In this mode, the user could change the destination, route, begin-date, end-date and control order quantity, which refers to a requisitioned block of railcars.
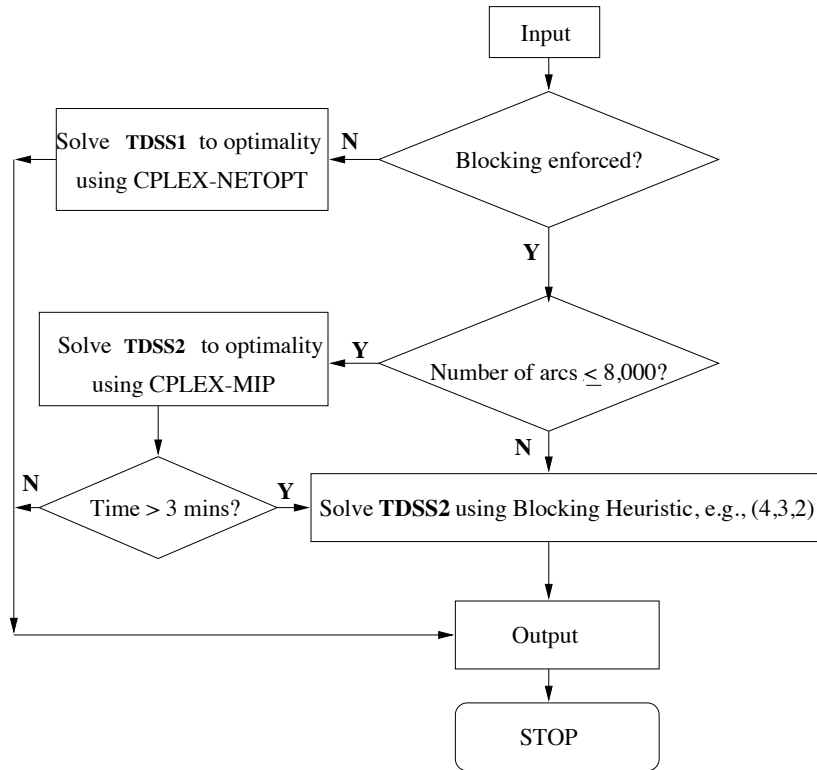
Figure 6.1: Flowchart for the Overall Tactical Decision Support System.

6.1.7 What-if on Temporary Dispositions

The user could view daily available supply amounts and create temporary dispositions in this mode. The latter can be created by selecting a supply location, supply day, demand location, and an associated route. The creation of a temporary disposition removes the railcars from the corresponding supply point and decreases the amount of demand at the corresponding demand location.

6.2 The System Flow Chart

Figure 6.1 depicts the schema of the overall solution procedure employed by the developed Tactical Decision Support System which is based on the results in Chapter 5. This process offers the choice between solving TDSS1 or TDSS2, and allows for an exact solution of TDSS2 using CPLEX-MIP whenever the problem size so permits.

6.3   Summary and Conclusions

The decision support system developed in this research has set a new standard in the railroad and automobile industries. Practical considerations such as a time-based prioritization of flows within a multiple objective framework, demand location-based priorities, variable queue sizes and the treatment of weekends in the demand profile, along with uncertainty and blocking considerations, have not been addressed in an integrated fashion in any existing system up to now.

Our first model, TDSS1, incorporates all of these practical considerations, except for blocking. Moreover, it progressively accommodates each of these features while retaining a network structured formulation that is well suited to the needs of rapid and tactical responses that invariably arise in practice.

Our second model, TDSS2, also incorporates blocking considerations into the formulation. In essence, this more complex model turns out to be a network flow problem with side-constraints and *discrete* side-variables. We have shown how the resulting mixed-integer-programming formulation can be enhanced via some partial convex hull constructions using the Reformulation-Linearization Technique (RLT). Consequently, this tightening process allows us to solve problem scenarios having 5,000 - 8,000 arcs to optimality. However, larger problem scenarios having a size up to 150,000 arcs pose a formidable challenge to solve to optimality, and almost certainly, the computational effort exceeds the imposed 5 minute run-time limit on a Pentium PC.

In order to deal with this obstacle faced in solving such large problem scenarios, we have devised 23 principal heuristics, plus other hybrid variants. Upon testing these heuristic procedures on a set of real problems, the examination of their performance with respect to speed of operation and the quality of solutions leads us to favor two Blocking Heuristics ((DS, VF, DR) = (4,3,2) and (DS, VF, DR) = (2,3,2)). Due to the better look-ahead capability of the (4,3,2) heuristic procedure, however, a variant of this procedure that operates within the time limitation constraint has been recommended for implementation.

Our next step in developing the desired decision support system involved integrating the two models (TDSS1 and TDSS2) and the heuristic procedures into a coherent system that takes advantage of the strengths of each individual model and the recommended heuristic procedure. Figure 6.1 of the previous section summarizes the overall solution procedure of the integrated system.

The resulting system has two-fold benefits in aiding the decision making process for the railroad and automobile industries. The first is to serve as a tool for guiding the actual dispatchment of empty railcars on a daily basis. More importantly, the other benefit of the system is to allow management to conduct various *what-if* analyses based on the incorporation of the many practical considerations into the system. Through a graphical user interface as mentioned in Section 6.1, the railcar distributors are able to investigate the effects of varying

supplies, demands, and routes, retrieving railcars from storage, diverting en-route railcars, and exploring various customer or user-driven fixed dispositions. Therefore, the user has the flexibility to sequentially compose a decision to implement on a daily basis by using business judgment to make suggestions and studying the consequent response prompted by the model.

Below, we describe some further research avenues to enhance the solution of Problem TDSS2, using a Lagrangian Relaxation strategy.

6.4 Future Research: Lagrangian Relaxation

Pursuant to Remark 4.1, in this section, we explore the use of a Lagrangian relaxation strategy to solve Problem TDSS2.

In the definition of problem **TDSS2** as given by ((4.79)-(4.87)), we note that if $G_p^+ = \emptyset$ for some source p and if $n_p = 1$, then (4.85) and (4.82) imply (even in the continuous sense) that

$$\sum_q x_{pq} \equiv \sum_{g \in G_p^f} \sum_{q \in J(g)} x_{pq} \leq \sum_{g \in G_p^f} U_{pg} y_{pg} \leq s_p \sum_{g \in G_p^f} y_{pg} \leq s_p, \qquad (6.1)$$

or that (4.81) holds true. Hence, for such a source, (4.81) is redundant and its Lagrange multiplier can be designated to be zero. Accordingly, let us denote

$$S_0 = \{p : \ G_p^+ = \emptyset \text{ and } n_p = 1\}. \qquad (6.2)$$

Dualizing (4.81) (written as $\geq$ constraints) and (4.80) using respective Lagrange multipliers α and β (where $\alpha_p \equiv 0 \ \forall \ p \in S_0$ as given by (6.2)), we obtain the following Lagrangian dual problem (**LD**).

LD: Maximize $\{\Phi(\alpha, \beta) : (\alpha, \beta) \geq 0 \text{ with } \alpha_p \equiv 0 \ \forall \ p \ \in \ S_0, \text{and } \beta_q \leq \mathcal{M}(DS, q) \ \forall \ q\}$ (6.3)

where

$$\Phi(\alpha, \beta) = \sum_q \beta_q d_q - \sum_p \alpha_p s_p + \text{minimum } \{\sum_p \sum_q (C_{pq} + \alpha_p - \beta_q) x_{pq} : (4.82) - (4.87)\}. \quad (6.4)$$

The function Φ defined by (6.3) is readily seen to be a piecewise linear concave function that can be optimized using a suitable deflected subgradient approach as discussed below. The main effort per iteration in this approach lies in evaluating the function Φ via (6.4). This can be efficiently accomplished as described next.

6.4.1 Evaluation of the Dual function Φ and its Subgradient ξ for a Given (α, β)

Observe that the minimization problem given in (6.4) is separable for each source p and group g. Following Figure 4.4, for each source p, we do the following.

Step 1. For each group g, if $g \in G_p^+$, we solve the problem

$$\text{Minimize} \quad \sum_{q \in J(g)} \overline{C}_{pq} x_{pq} : \quad L_{pg} \leq \sum_{q \in J(g)} x_{pq} \leq U_{pg}$$

$$l_{pq} \leq x_{pq} \leq u_{pq} \ \forall \ q \in J(g)\} \tag{6.5}$$

$$\text{where} \quad \overline{C}_{pq} \equiv C_{pq} + \alpha_p - \beta_q \ \forall \ p, q.$$

Problem (6.5) can be easily solved as follows. Denote

$$C_{pg}^+ = \{q \in J(g) : \overline{C}_{pq} \geq 0\} \text{ and } C_{pg}^- = \{q \in J(g) : \overline{C}_{pq} < 0\}. \tag{6.6}$$

Assume that the indices in each set defined in (6.6) have been arranged in nondecreasing order of the $\overline{C}_{pq}$ values. (Note that in the algorithmic scheme described below, only a partial ordering might be necessary, and C_{pg}^+ may not need to be ordered at all. Hence, for efficiency, this ordering may be done only as necessary.) Also, denote by T_{pg} the sum $\sum_{q \in J(g)} x_{pq}$ of the flow variables.

Step 1a : Initialize $x_{pq} = l_{pq} \ \forall \ q \in J(g)$ and let $T_{pg} = \sum_{q \in J(g)} l_{pq}$.

Step 1b : If $C_{pg}^- = \emptyset$, go to Step 1d.

Step 1c : Increase x_{pq} from its lower bound up to its upper bound in order of the indices in C_{pg}^-, updating the sum T_{pg} accordingly, so long as $T_{pg} \leq U_{pg}$.

Step 1d : If $T_{pg} \geq L_{pg}$, go to Step 1f.

Step 1e : Increase x_{pq} from its lower bound up to its upper bound in order of the indices in C_{pg}^+, updating the sum T_{pg} accordingly, until $T_{pg} = L_{pg}$.

Step 1f : STOP, optimum at hand.

Let

$$x_{pq} = \overline{x}_{pq}, \text{ for } q \in J(g), \ g \in G_p^+ \tag{6.7}$$

be the solution thus obtained.

Step 2. Next, for each group $g \in G_p^f$, suppose that we also solve (6.5) as in Step 1 (noting that $l_{pq} = 0$ by (4.84) in this case) and obtain the solution $\overline{x}_{pq}$ for $q \in J(g)$. However, in this case, observing that the variable y_{pg} scales all the constraint constants/interval-end-points, we obtain that an optimal set of x_{pq} values is given by

$$x_{pq} = \overline{x}_{pq} y_{pq} \ \forall \ q \in J(g), \ g \in G_p^f. \tag{6.8}$$

Step 3. For each p, substituting x_{pq} as given by ((6.7-6.8)) into the objective function of (6.4), reduces this problem to

$$\Phi(\alpha, \beta) = \overline{\nu} + \text{ minimum } \{\sum_p \sum_{g \in G_p^f} \eta_{pg} y_{pg} : \sum_{g \in G_p^f} y_{pg} \leq n_p \ \forall \ p \ni \ G_p^f \neq \emptyset, y \text{ binary}\} \tag{6.9}$$

where ν and η_{pg} $\forall$ p, g are suitably accumulated constants. For each p $\ni$ $G_p^f \neq \emptyset$, Problem (6.9) is now easily solved by setting up to n_p of the variables y_{pg} for $g \in G_p^f$ equal to one in nondecreasing order of η_{pg}, so long as $\eta_{pg} < 0$. Let $\overline{y}$ be the solution thus obtained. This evaluates $\Phi(\alpha, \beta)$ via (6.9). Moreover, denoting the optimal solution for (6.4) by x^* as obtained via ((6.7-6.8)), substituting $y_{pg} = \overline{y}_{pg}$ in (6.8), a subgradient $\xi \equiv (\xi_\alpha, \xi_\beta)$ of Φ at (α, β) is given by

$$(\xi_\alpha)_p = -s_p + \sum_q x_{pq}^* \quad \forall\ p \notin S_0 \tag{6.10}$$

$$(\xi_\beta)_p = d_q - \sum_p x_{pq}^* \quad \forall\ q. \tag{6.11}$$

Note that by virtue of the restriction on β in (6.3), each Lagrangian subproblem solution yields $x_{DS,q} \equiv 0\ \forall\ q$.

6.4.2 Conjugate/Deflected Subgradient Algorithm for Optimizing the Lagrangian Dual

Sherali et al. (1995) have developed a conjugate subgradient variable target value method that could be used for a fixed number of iterations (along with an improvement-based stopping criterion for a possible earlier termination) in order to solve LD. The purpose here is to use this scheme to derive a set of blocking decisions in order to obtain an upper bound on the problem, and to use the Lagrangian dual based lower bound to assess the quality of the former solution.

Toward this end, we can first of all employ the y solution obtained via (6.9) corresponding to the final incumbent dual solution that results from solving LD. Using this set of blocking decisions, we can solve the corresponding transportation problem (using CPLEX or its network flow option) to obtain the associated flow decisions.

Another possible solution can be derived from the estimate of the LP relaxation to TDSS2. As shown by Sherali and Choi (1996), a particular weighted average of the subproblem solutions theoretically converges to an optimum to this LP relaxation. Motivated by this result, for the final 50 or so iterations of the Lagrangian dual optimization scheme, we can construct this average for just the y-part of the subproblem solutions. Let $\hat{y}$ be the resulting average thus obtained. A y solution feasible to ((4.82),(4.87)) can now be obtained as follows.

For each $p \ni G_p^f \neq \emptyset$, in decreasing order of $\hat{y}_{pg} \geq 0.5$ and until n_p of the $y_{p.}$ (variables have not been set at one), continue by putting $y_{pg} = 1$. (6.12)

Then, using the y-solution obtained via (6.12), we can solve the associated transportation problem in the x-variables in order to determine the flow decisions.

The two heuristic solutions thus obtained can be compared with respect to their objective values, and the better solution can be prescribed for implementation. This would afford

an alternative scheme to the heuristics developed in Chapter 4. However, with the current computer technology, the effort involved would probably exceed the set practical limitation for the larger sized problem instances. We hence defer this alternative to future consideration.

Bibliography

ALLMAN, W. P. 1974. An Optimization Approach to Freight Car Allocation under Time-Mileage per Diem Rental Rates. *Management Science.* 18, 567-574.

ASSAD, A. A. 1980. Models for Rail Transportation. *Transportation Research - A.* 14A, 205-220.

ASSOCIATION OF AMERICAN RAILROADS. 1976. Federal Car Utilization Research Demonstration Program (FCUP): Manual of Car Utilization Practices and Procedures. *Task Force 1. NTIS.* Springfield, Virginia.

AVI-ITZHAK, B. B. A. BENN, B. A. POWELL. 1967. Car Pool Systems in Railroad Transportation: Mathematical Models. *Management Science.* 13, 694-711.

BALL, M. O., B. L. GOLDEN, A. A. ASSAD, AND L. D. BODIN. 1983. Planning for Truck Fleet Size in the Presence of a Common-Carrier Option. *Decision Sciences.* 14, 103-120.

BAZARAA, M. S., J. J. JARVIS, AND H. D. SHERALI. 1990. *Linear Programming and Network Flows.* 2nd Ed. John Wiley and Sons.New York.

BEAUJON, G. J. AND M. A. TURNQUIST. 1991. A Model for Fleet Sizing and Vehicle Allocation. *Transportation Science.* 25(1), 19-27.

BODIN, L. D., B. L. GOLDEN, AND A. D. SCHUSTER. 1980. A Model for the Blocking of Trains. *Transportation Research - B.* 14B, 115-120.

BOOLER, J. M. P. 1980. The Solution of a Railway Locomotive Scheduling Problem. *Journal of the Operational Research Society.* 31, 943-948.

CRAINIC, T. G., M. FLORIAN AND J. E. LEAL. 1991. A Model for the Strategic Planning of National Freight Transportation by Rail. *Transportation Science.* 24(1), 1-24.

CRAINIC, T. G., M. GENDREAU, AND P. DEJAX. 1993. Dynamic and Stochastic Models for the Allocation of Empty Containers. *Operations Research.* 41(1), 102-126.

DEJAX, P. J. AND T. G. CRAINIC. 1987. A Review of Empty Flows and Fleet Management Models in Freight Transportation. *Transportation Science.* 21(4), 227-247.

DEODOVIC, D. 1988. *Airline Operations Research.* Gordon & Breach Science Publishers.

FEENEY, G. 1957. Controlling the Distribution of Empty Freight Cars. *Proceedings of the 10th National Meeting, Operations Research Society of America.*

FELTON, J. R. 1978. *The Economics of Freight Car Supply.* University of Nebraska Press. Lincoln.

FLORIAN, M., G. BUSHELL, J. FERLAND, G. GUERIN AND L. NASTASKY. 1976. The Engine Scheduling Problem in a Railway Network. *INFOR.* 14, 121-138.

FORBES, M. A., J. N. HOLT, AND A. M. WATTS. 1991. Exact Solution of Locomotive Scheduling Problems. *Journal of the Operational Research Society.* 42, 825-831.

HARKER, P. T. 1990. The Use of Satellite Tracking in Scheduling and Operating Railroads: Models, Algorithms and Applications. In *Proceeding of the Fifth World Conference on Transport Research-Yokohama.* Western Periodicals, Ventura, California, B299-B312.

JARKE, M. 1982. Developing Decision Support Systems: A Container Management Example. *International Journal Policy Analytical Information System.* 6, 351-372.

JORDAN, W. C. AND M. A. TURNQUIST. 1983. A Stochastic, Dynamic Network Model for Railroad Car Distribution. *Transportation Science.* 17(2), 123-145.

KEELER, T E. 1984. *Railroads, Freight and Public Policy.* The Brookings Institutions. Washington, D.C.

LEDDON, C. D. AND E. WRATHALL. 1968. Scheduling Empty Freight Car Fleets on the Louisville and Nashville Railroad. *Cybernetics Electronic Railways.* 3, 154-158.

MENDIRATTA, V. B. AND M. A. TURNQUIST. 1982. Model for Management of Empty Freight Cars. *Transportation Research Record.* 838, 50 - 55.

MISTRA, S. C. 1972. Linear Programming of Empty Wagon Disposition. *Rail International.* 3, 151-158.

MUEHLKE, R. V. 1984. General Model of Multirailroad Freight Car Management. *Transportation Research Record.* 966, 14-22.

MULVEY, J. M. 1995. Robust Optimization of Large-Scale Systems. *Operations Research.* 43(2), 264-281.

NEWTON, H. N., C. BARNHART, P. H. VANCE. 1997. Finding Near-optimal Railroad Blocking Plans. *Transportation Science.* Manuscript.

PETERSEN, E. R. 1977a. Railyard Modeling – Part I. Prediction of Put-through Time. *Transportation Science.* 11(1), 37-49.

PETERSEN, E. R. 1977b. Railyard Modeling – Part II. The Effect of Yard Facilities on Congestion. *Transportation Science.* 11(1), 50-59.

PHILIP C. E. ND J. M. SUSSMAN. 1977. Inventory Model of the Railroad Empty Car

Distribution Process. *Transportation Research Record.* 656, 52-60.

SHERALI, H. D. 1982. Equivalent Weights for Lexicographic Multi-objective Programs: Characterizations and Computations. *European Journal of Operational Research* 11(4), 367-379.

SHERALI, H. D. 1988. Bounds on Penalties for Dummy Arcs in Transportation Networks. *European Journal of Operational Research* 36(3), 353-359.

SHERALI, H. D. AND W. P. ADAMS. 1990. A Hierarchy of Relaxations between the Continuous and Convex Hull Representations for Zero-One Programming Problems. *SIAM Journal on Discrete Mathematics.* 3, 411-430.

SHERALI, H. D. AND W. P. ADAMS. 1994. A Hierarchy of Relaxations and Convex Hull Characterizations for Mixed-Integer Zero-One Programming Problems. *Discrete Applied Mathematics.* 52, 83-106.

SHERALI, H. D. AND G. CHOI. 1996. Recovery of Primal Solutions When Using Sub-gradient Optimization Methods to Solve Lagrangian Duals of Linear Programs. *Operations Research Letters.* 19(3), 105-113.

SHERALI, H. D. AND A. L. SOYSTER. 1983. Preemptive and Nonpreemptive Multi-Objective Programs: Relationships and Counter Examples. *Journal of Optimization Theory and Applications.* 39(2), 173-186.

SHERALI, H. D. AND C. H. TUNCBILEK. 1997. Static and Dynamic Time-Space Strategic Models and Algorithms for Multi-level Rail-car Fleet Management. *Management Science.* 43(2), 235-250.

SMITH, S. AND Y. SHEFFI. 1989. Locomotive Scheduling under Uncertain Demand. *Transportation Research Record.* 1251, 45-53.

SUZUKI, S. 1973. A Method of Planning Yard Pass Trains on a General Network. *Operational Research '72.* Ed. M. Ross. 353-361. North Holland, Amsterdam.

TURNQUIST, M. A. 1985. Research Opportunities in Transportation System Characteristics and Operations. *Transportation Research.* 19A, 357-366.

TURNQUIST, M. A. AND W. C. JORDAN. 1986. Fleet Sizing under Production Cycles and Uncertain Travel Times. *Transportation Science.* 20(4), 227-236.

WHITE, W. W. 1972. Dynamic Transshipment Networks: An Algorithm and its Application to the Distribution of Empty Containers. *Networks.* 2, 211-236.

WHITE, W. W. AND A. M. BOMBERAULT. 1969. A Network Algorithm for Empty Freight Car Allocation. *IBM Systems Journal.* 8(2), 147-169.

WRIGHT, M. B. 1989. Applying Stochastic Algorithms to a Locomotive Scheduling Problem. *Journal of the Operational Research Society.* 40, 187-192.

Vita

Arief B. Suharko was born on April 3, 1968 in Jakarta, Indonesia. In 1987, he began his B.S degree program in electrical engineering at the University of Kansas with the help of the OFP scholarship program. In his senior year, he took an area of concentration in Radar and microwave communications. He graduated in 1991 with distinction.

He immediately entered the graduate program of the same department under the STMDP scholarship program. In this program, he studied the performance issues in high-speed computer networking. He received his Master of Science degree in 1993.

In the same year, he continued to pursue a Ph.D. degree in Industrial and Systems Engineering under the operations research program option at Virginia Polytechnic Institute and State University (Virginia Tech). He has worked as a teaching assistant, a system administrator as well as a research assistant to support himself in his doctoral study. Upon completing the program, he hopes to return to his home country, Indonesia, and work in the Ministry of Research and Technology.