Optimizing Intermodal Rail Operations

by

Alexandra Mary Newman

B.S. (University of Chicago) 1993
M.S. (University of California, Berkeley) 1994

A dissertation submitted in partial satisfaction of the
requirements for the degree of

Doctor of Philosophy

in

Engineering – Industrial Engineering
and Operations Research

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:
Professor Candace A. Yano, Chair
Professor Ilan Adler
Professor Adib Kanafani

Fall, 1998
The dissertation of Alexandra Mary Newman is approved:

Candace Anai Yano  7/31/98
Chair

T. Adler  7/20/98
Date

6/28/98
Date

University of California, Berkeley

Fall, 1998
Abstract

Optimizing Intermodal Rail Operations

by

Alexandra Mary Newman

Doctor of Philosophy in Industrial Engineering and Operations Research
University of California, Berkeley
Professor Candace A. Yano, Chair

Truck-rail intermodal operations entail the transport of freight containers by truck from the shipper to a nearby intermodal terminal, by train for the long-haul portion of the journey between intermodal terminals, and by truck from the destination intermodal terminal to the consignee. The truck-rail combination provides the fuel and labor efficiency of long-haul trains and the door-to-door service of trucks.

We study rail intermodal operations and address the problem of how to schedule trains and allocate containers to the trains to meet due dates while minimizing the sum of fixed costs for running the trains and per unit costs for transporting containers and holding them in inventory. We consider both direct (origin to destination) and indirect (via a hub) trains, as well as dynamic arrivals of containers over a multi-period planning horizon.

We formulate this optimization problem as a mathematical program with integer decision variables. For problems with this structure, neither commercial optimization software nor classical solution approaches such as Lagrangian relaxation and Bender's decomposition can provide near-optimal solutions. We develop a new decomposition approach in which, broadly speaking, the train scheduling and container allocation decisions are made first at the origins, then outbound from the hub(s). We devise several schemes to implement this approach that differ in the degree of centralization of decision-making and in the information requirements at various decision points. We compare our approach with simple, common-sense methods that were designed to mimic current procedures, and with lower bounds (i.e., valid lower limits on the minimum cost). We develop methods to obtain these lower bounds that are much closer to the minimum cost than those provided by commercial optimization software. Our numerical study suggests that the new approach provides near-optimum solutions that could afford considerable savings from the costs incurred using current procedures.
Our methodology can be extended to other settings in which there are fixed costs to provide increments of transportation capacity at various locations and times, and per unit costs for the transportation of freight using the selected transportation schedule. Examples arise in air freight, trucking, and sea operations.

Candace A. Yano, Chair
# Contents

1 Introduction and Literature Review ........................................ 1
  1.1 Introduction and Background ........................................ 1
  1.2 Literature Review .................................................. 6
    1.2.1 Traditional Rail Operations .................................. 6
    1.2.2 Truck Operations .............................................. 9
    1.2.3 Multimodal Models ............................................ 11
    1.2.4 Intermodal Operations ....................................... 11
    1.2.5 Models Possessing Related Mathematical Structure ....... 15

2 Mathematical Formulation and Problem Structure ...................... 21
  2.1 Mathematical Formulation .......................................... 25
  2.2 Network Description and Formulation ................................ 29
  2.3 Problem Size .................................................................. 34
  2.4 Nature of the Problem ................................................ 35

3 Classical Decomposition Techniques ...................................... 38
  3.1 Lagrangian Relaxation ............................................... 38
    3.1.1 Lagrangian Relaxation Applied to the Original Problem .... 39
  3.2 Bender’s Decomposition .............................................. 41
    3.2.1 Procedure and Bender’s Formulation .......................... 41
    3.2.2 Implementation and Subproblem Feasibility ................. 45
    3.2.3 Adding Constraints to the Master Problem .................. 46
    3.2.4 Sign of the Dual Multipliers .................................. 50
    3.2.5 Other Observations ............................................ 54

4 New Decomposition Technique ............................................. 56
  4.1 Description of the Approach ........................................ 56
  4.2 Suboptimality of the Decomposition Procedure ..................... 65
  4.3 Methods to Handle Many Destinations ............................... 68
  4.4 The Value of Centralized Decision Making ......................... 74
    4.4.1 The Procedures ................................................ 74
    4.4.2 A Note on the Subproblem Structure ........................ 78
  4.5 Numerical Results: Testing the New Decomposition Approach .... 82
    4.5.1 Problem Parameters and Data .................................. 82
    4.5.2 Initial Results ............................................... 84
    4.5.3 Results: The Value of Centralized Decision Making ....... 85
  4.6 Time Performance of the Decomposition Procedure ................. 87
List of Figures

1.1 The Intermodal Journey .................................................. 2
1.2 Relative Performance of Conventional and New Procedures ............. 20

2.1 Representation of the Single Commodity Network .......................... 31

4.1 The Two Parts of the Decomposed Problem ................................ 57
4.2 Network Depiction of the Origin Scheduling Problem ...................... 62
4.3 Network Depiction of the Hub Scheduling Problem .......................... 63
4.4 The Decomposed Problem: Many Destinations ............................... 70

5.1 Super Train Capabilities ................................................... 94
5.2 Arcs Included in Lower Bound Bundle Constraints: Single Origin to Single Destination: 1 ........................................ 95
5.3 Arcs Included in Lower Bound Bundle Constraints: Single Origin to All Destinations ............................................. 95
5.4 Arcs Included in Lower Bound Bundle Constraints: Single Origin to Single Destination: 2 ........................................... 96
5.5 Arcs Included in Lower Bound Bundle Constraints: All Origins to Single Destination ............................................... 97
List of Tables

2.1 Example Problem Data, I ................................................. 36
3.1 Example Problem Data, II ............................................... 46
4.1 Container Arrival Data for Example ................................. 66
4.2 Train Schedule into Hub: Nonoptimal for Hub Scheduling Problem ................................. 66
4.3 Train Schedule into Hub: Optimal for Hub Scheduling Problem ............................................... 66
4.4 Data for Aggregation Procedure ........................................ 73
4.5 Direct Train Schedule under the Aggregation Procedure ................. 73
4.6 Test Problem Characteristics .......................................... 82
4.7 Parameters for Test Problem Instances ................................ 83
4.8 Improvement in the Best Integer Solution and Reduction in Gap ................. 86
4.9 Percent Improvement in Best Integer Solution over Straightforward CPLEX Implementation .......................................... 88
4.10 Time Performance: Initial Implementation vs. Decomposition Procedure ................. 90
5.1 Performance Improvement in Gap via Tightening the Lower Bound ................. 101
5.2 Relative Contributions to Improvement in the Lower Bound ................................. 103
5.3 Comparison of Initial Performance and the Gap after Implementing our Decomposition and Lower Bound Improving Procedures ................. 104
5.4 Gap Estimates for “Hard” Problems ................................... 105
6.1 Heuristics- Percent Above our Best Integer Solutions ......................... 111
Acknowledgements

After having observed a significant number of data points, I have reached the conclusion that an adviser's support has the greatest impact on graduate student life and the immediate afterlife. Indeed, my own data point in this collection fits my conjecture. Therefore, I would like to express my utmost gratitude to my adviser, Professor Candace A. Yano, for all her help, comments, patience, support, and understanding throughout my entire graduate career, which has set me on the right track (no pun intended). She struck an optimal balance between guidance and independence and gave me an exceptional number of opportunities, including "freedom" of research pursuits, teaching responsibilities, and exposure at national conferences. She exhibited an amazing ability to tolerate my Alexandrian behavior and thought patterns, and was constantly available so as to be able to incorporate my somewhat unconventional lifestyle into her schedule: "I think it's about time you went running..."

I would also like to thank the other two members of my committee, Professors Ilan Adler and Adib Kanafani for their comments and support. I would like to acknowledge the significant assistance of my two fellow optimizers at Berkeley, Jennifer Goodhart and Eli Olinick, as well as the input of several professors at the Naval Postgraduate School.

On a more personal level, I would like to thank my family, especially my parents, who taught me the power of hard work, and who unceasingly tolerated stochastic pieces of unsolicited information.

Finally, my gratitude extends to all my teammates and running friends, most notably Michay Brown, Sona Banker, Steve Puller, and Drake Dawson, who indirectly kept me going (running?) and turned my attention to aspects of life other than my work for at least several consecutive nanoseconds.

This research was supported in part by National Science Foundation Grant HRD 93-96288 to the University of California, and by funding from the United States Department of Transportation and the California Department of Transportation, awarded by the University of California Transportation Center.
Chapter 1

Introduction and Literature Review

1.1 Introduction and Background

Intermodal transportation has existed for centuries and consists of combining modes, usually ship, truck, and/or rail to transport freight. The focus of our research is on rail-truck operations that combine the long haul efficiency of rail transportation with the door-to-door service of trucks. The concern for increasing the efficiency of intermodal operations has risen recently due to legislation that has allowed intermodal rates to become competitive with those of trucks and barges. This fortuitously coincides with the growing need for transportation capacity and capabilities due to increased traffic levels, which have resulted in highway congestion, higher accident rates and increasingly stringent environmental regulations. Despite recent advances in the efficiency of intermodal operations, difficulties remain. Some obstacles the railroads face are due to inadequate infrastructure including a shortage of track, and the lack of a fully operational, continuous transcontinental railroad. Other difficulties arise, in part, due to basic management and information deficiencies, which lead to poor operational procedures in developing train routes and schedules, and priority rules for sending shipments (Report to the Committee on Public Works and Transportation, 1992). Because of the extra time incurred and the increased potential for mishandled containers at intermodal terminals, it is important that time and cost considerations be taken seriously if intermodal transportation hopes to continue to compete with long haul trucking.

Our research is motivated by the opportunity we see for rail-truck intermodal transportation to realize its potential more fully. Specifically, we formulate an integer program to address decisions for the rail, or linehaul, portion of the intermodal trip regarding when
and how to route direct and indirect trains and which containers to send on each train to minimize operational costs to the railroad while meeting on-time delivery requirements of the shipments. It should be noted that shippers who do not currently use intermodal transportation claim that were the transit time even one day faster, they would consider switching. Hence, astronomical improvements need not be achieved in order for intermodal transportation to be much more competitive with long haul trucking.

Intermodal rail-truck service is composed of a truck transportation (drayage) portion of the journey during which a container or trailer travels from the shipper to the intermodal terminal, a line haul segment during which the container or trailer is moved via rail to a terminal relatively near its destination, and a second drayage segment during which the container or trailer is transported via truck to the consignee (McKenzie, 1989). Figure 1 depicts the intermodal journey.

Primitive containers were barrels that were transferred from ships to wagons. During the mid-1800s, the barrels were transformed into containers with wheels for easier shipping and handling. Containers remain a primary component of intermodal equipment used today; they are boxes used for shipping goods such that no intermediate reloading of goods is necessary at intermodal transfer terminals. They are durable enough for repeated shipping,
have handles, can be filled and emptied easily, and come in a variety of sizes. The most common containers are 8 to 9.5 feet high and 8 feet wide. Containers used for international shipping have lengths of 20 and 40 feet, but domestic containers can be as long as 53 feet. In order to accommodate the goods being shipped, containers can be insulated, refrigerated, open top, tank (a tank in a box-like framework) or platform (no side walls). Containers are moved on a chassis over the roads. They are transported on double or single stack cars by rail in what is known as a container-on-flatcar (COFC) configuration (McKenzie, 1989).

Another common intermodal unit is the trailer, which is similar to a container except that it possesses its own wheels for road transportation. Trailers usually range from 40 to 53 feet in length and are transported by trains on flat cars in what is termed a trailer-on-flatcar (TOFC) configuration. Although trailers are simply attached to trailer cabs for over the road hauling, and thus eliminate the need for a chassis, they are not as efficient to transport by rail as containers. They cannot be double stacked and because their shape creates more resistance, the fuel economies are not as great (McKenzie, 1989).

Intermodal terminals are locations at which an incoming container or trailer is transferred between modes. In recent years, as the equipment at terminal yards (e.g., cranes for loading and unloading the trailers and containers) has become more sophisticated and correspondingly more expensive, there has been a trend towards a reduction in the number of terminals (McKenzie, 1989).

The interest in increasing the efficiency of intermodal operations has risen recently due to legislation that has allowed intermodal rates to become competitive with those of trucks and barges. This interest is coupled with the growing concern for increased transportation capacity and capabilities. The number of ton-miles of freight transported by truck has continued to rise from about 413 hundred million ton miles in 1970 to 735 hundred million ton miles in 1990 to 921 hundred million ton miles in 1995 (National Transportation Statistics, 1997). This increased traffic has taken its toll on the nation’s highway systems. For example, in 1995 there were 10,183 fatalities resulting from accidents involving trucks. Air pollution and traffic congestion have become major concerns, especially around large cities such as Los Angeles and New York. Highways have also deteriorated due to heavy vehicle loads.

Because of the increased amount of freight traffic, as well as more stringent environmen-
tal regulations, intermodal transportation has been gaining popularity. When used for a
distance of at least 500 miles, the long haul efficiency of trains results in savings in terminal
costs and labor (employees required to move a given amount of freight). The Association of
American Railroads estimates that railroads can haul a given quantity of freight at 1/5 the
fuel expenditure, 1/6 the accidents and 1/10 the land of motor carriers. Furthermore, rail-
roads carry seven times as much freight per employee as motor carriers (McKenzie, 1989).
Finally, because rail transport diverts some freight traffic from the roads, congestion and
wear and tear on the highway system is partially alleviated.

The trend towards increased use of intermodal transportation can be seen clearly. From
1980 to 1992, the number of trailers or containers being shipped intermodally skyrocketed
from 3 million to 6.7 million, and is currently still rising (Railroad Facts, 1993). This trend
has been encouraged by recent improvements in the intermodal system. Intermodal traffic
is often run on a schedule, and has a higher priority over rail lines than all other traffic
except passenger trains. Railroads have acquired more sophisticated equipment for moving
containers, thereby increasing the flexibility in the types of goods that can be shipped.
To decrease damage while the container is en route, slack has been reduced between rail
cars, and ordinary boxcar traffic carrying bulk commodities such as gravel, coal, or grain,
is commonly separated from intermodal traffic. Furthermore, intermodal traffic is not sent
through classification yards called humpyards, in which cars are sorted and reattached to
form a new train by momentum they gain from being sent over a hill, or “hump”; the jarring
motion can damage more delicate freight (McKenzie, 1989).

The intermodal business has acquired dedicated customers such as United Parcel Ser-
tice, which sends many of its trailers on flatcars across the country. Less than truckload
companies also use intermodal service for hauling their trailers, especially for reposition-
ing purposes. Truckload companies are similarly finding intermodal service useful for their
transcontinental shipments as a means of decreasing driver dissatisfaction and fatigue caused
by many long hours on the road and an unreasonable amount of time spent away from home
(Harper, 1993).

Despite these advances, basic infrastructural and operational problems remain. Much
of the nation’s track has been consolidated to reduce maintenance costs under the pretense
that railroads were becoming an archaic means of transportation. This, however, has hurt
the rail industry. In large metropolitan settings with high densities of traffic, the rail lines are completely inadequate to handle the current traffic volumes. For example, around the ports of Los Angeles and Long Beach, the gridlock along four miles of precariously existing rail can lead to delays of up to six hours. Not only is the stretch of track poorly maintained, resulting in derailments, but the volume of traffic is so high that the arrival of a train just a few minutes too late can have catastrophic systemwide effects (Machalaba, 1996).

Intermodal shipping costs can be unreasonably high because drayage may account for up to 40% of the total transportation cost. Train and truck companies that provide intermodal service are often unwilling to share information about their shipments with each other, resulting in many miles of empty container travel. Slow handling at intermediate terminals resulting from inefficient equipment use also leads to disproportionately high drayage costs (Morlok and Spasovic, 1994).

Operational problems, including the timeliness of service, have deterred potential customers. Because railroads have difficulty guaranteeing a given service level, shippers often rely on trucking companies that meet stringent on-time delivery requirements. Inadequate service can result from poor train routes and schedules. For example, trains may be routed through hubs to take advantage of consolidation opportunities, under the pretense that traffic volumes are not high enough to warrant dedicated train service. The delays associated with the indirect routing of traffic lead to this self-fulfilling prophecy. Because traffic is sent indirectly, shippers with time-sensitive deliveries avoid rail transportation. This, in turn, tends to limit traffic levels. Customers are now demanding more frequent service between given origin-destination pairs, and direct service along certain corridors. Along indirect routes, shippers are asking for more coordination between the train schedules, so that shipments do not simply sit for days in the terminal. There are related requests for "seamless transportation" to mimic a trucking company's ability to provide both door-to-door service and information as to the exact location of the shipment at any point en route (Richardson, 1990, Wallace, 1997).

Because the potential for increased intermodal transportation use exists, railways are seeking means to improve the overall quality of their operations to make them more competitive with the long haul trucking industry (Richardson, 1989). Our research is motivated by the opportunity we see for rail-truck intermodal transportation to realize its potential
more fully. Specifically, we formulate an integer program to address decisions for the rail, or linehaul, portion of the intermodal trip regarding when and how to route direct and indirect trains and which containers to send on each train. Given container demands differentiated by origin, destination, arrival date at origin, and due date, the objective is to determine a train schedule and container routing scheme to minimize the overall operational cost to the railroad while meeting on time delivery requirements of the shipments. The cost of the train crew is fairly insensitive to the number of containers on the train. Thus, as the number of containers transported on a train increases, greater economies of scale can be realized. We consider both the fixed cost of sending direct and indirect trains, and the variable cost of transporting, handling, and holding the containers in inventory.

Our problem can be formulated as a multi-commodity, piecewise-concave-cost network design and flow problem. It has a specialized structure which can be divided into two parts. The “easy” part of the problem consists of routing the containers from the origins to the destinations, possibly via a hub (i.e., in a network) given a fixed train schedule. The “hard” part of the problem consists of determining an optimal or near-optimal train schedule given the arrival of containers at the various origins in each time period, their destinations, and their due dates.

1.2 Literature Review

We now present the literature review, which is organized as follows: We first describe models used for the operations of the constituents of intermodal transportation, namely for rail and for truck operations. We then present models for multimode and intermodal operations. Finally, we review literature that treats classes of problems with a mathematical structure similar to that of our formulation.

1.2.1 Traditional Rail Operations

Because our research focuses on the combination of trucking and rail transportation, we give a brief overview of prior research on conventional rail or boxcar operations. This work is, for the most part, not directly applicable in the intermodal setting, because of the different ways in which boxcar and intermodal traffic are handled. For example, many boxcar operations use the concept of blocking, in which groups of cars bound for destinations
in close proximity to each other, or that lie along the same rail line, travel together on a train. At intermediate terminals, or humpyards, these blocks are reclassified onto different trains according to their destination. Because of the jarring motion caused by sending the boxcars over the reclassification humps, intermodal traffic is now separated from boxcar traffic. In general, intermodal traffic incurs fewer stops and reclassifications between its origin and destination than does boxcar traffic. Furthermore, intermodal traffic originates at and is bound for intermodal terminals, as opposed to to boxcar operations, in which boxcars are added along sidings of track adjacent to the main lines, at various points along the journey.

The nature of the goods shipped intermodally also differs from those sent in boxcars. Intermodal goods are not only more fragile (e.g., finished products as opposed to raw materials such as coal, gravel, or sheet metal), but they are also more time-sensitive. Therefore, these goods are treated with a greater sense of urgency, and timetables for running intermodal trains have been established. In contrast, boxcar trains are not run on a schedule. Nozick (1997) gives a thorough description of the nature of intermodal transportation and its differences from classical boxcar operations.

Folk and Bharadwaj (1980) propose a model for the distribution of Conrail’s empty freight cars. The objective is to minimize the total cost of moving freight from the origin to its destination (including the cost of obtaining freight cars, if necessary) subject to supply and demand constraints for the cars. This problem can be translated to an intermodal setting, in which the cars correspond to flatcars or stack cars. Because of the cost of rail cars, it is desirable to maintain a fleet as small as possible, or, equivalently, to minimize empty car miles. There are two types of freight cars, assigned and free-running cars. Assigned cars are dedicated to a specific customer. While the customer can be assured of a reliable car supply, and the rail company can be assured of the customer’s business and the good upkeep of the car, this arrangement results in many empty car miles. Free-running cars are assigned to customers on an as-needed basis. Although these cars average a time-weighted utilization of 70%, about 20 percentage points better than for assigned cars, damage is a problem because there is less control over these cars. The authors consider only free running cars when seeking to reduce the total time cars spend empty.

An implementation of the model results in about a one-third reduction in empty car
mileage by taking advantage of situations that allow empty cars not owned by Conrail to be repositioned to carry loads near their current locations rather than be sent home empty and have a Conrail-owned car brought from a distance to haul the load.

Kikuchi (1985) formulates a linear program to allocate the supply and demand of empty cars in order to improve the car utilization. Cars are either kept in inventory at their current location, or allocated to a point at which they are needed. If a shortage of cars exists, they must be procured, perhaps as free-running cars. Glickman and Sherali (1985) treat a similar problem to which they apply a large scale network algorithm and decomposition procedures to aid in the solution procedure.

Fuel economies can be realized by running trains at less than their maximum speed. For example, it may be more efficient to run a train at a lower speed over hilly terrain in order to maximize the ratio between tractive power and fuel expenditure. To this end, Kraay, Harker, and Chen (1991) develop a nonlinear, mixed integer programming model to determine the optimal pacing of trains subject to meeting due date requirements.

Because much of the rail network in the United States has been reduced to single track in order to decrease maintenance costs, schedule coordination is extremely important to avoid delays incurred by trains meeting head-on. Cai and Goh (1993) formulate an integer program to determine the optimal scheduling of trains on a single track, where trains are allowed to cross only at a single passing loop. They generate a heuristic which quickly produces good solutions.

Bussieck, Kreuzer, and Zimmerman (1996) consider the problem of choosing the optimal set of direct traffic lines on which to service passengers while satisfying a "line frequency requirement." This optimal set of traffic lines is defined as the set along which the maximum number of passengers are serviced, while adhering to capacity constraints for each line. In their model, it is not necessary to service all the demand. They formulate the problem as a mixed integer program and apply relaxation procedures and cutting planes within a branch and bound algorithm in order to obtain solutions that are within about 3% of optimality.

Marín and Salmerón (1996) formulate an integer program within the classical rail setting to determine an optimal train schedule for a rail network, and the optimal assignment of rail cars to these trains such that each train carries cars of a given service class. They employ artificial intelligence methods, specifically, simulated annealing and tabu search. For small
problem instances containing a few hundred variables and constraints, they compare their results with an exact solution obtained from a specialized branch and bound algorithm. They conclude that for these smaller problem instances, their solution procedures provide results that are approximately 5% to 15% from the optimal solution.

1.2.2 Truck Operations

In addition to rail modeling, there has been extensive work done on the modeling of truck routes for the delivery of goods. In contrast to the rail models, these formulations generally address the local delivery portion of a trip, rather than the long distance segment of the journey. This emphasis matches with the nature of rail-truck intermodal operations in which the line haul portion of the trip is performed by rail and the local delivery portion by truck. Although the focus of our research is on the rail portion, we mention several truck models for completeness.

Jordan and Burns (1984) address the problem of truck routing with backhauling. Trucks travel from a depot to a region in which they make multiple local stops, and the concern is to have these trucks return to the depot carrying goods from the region they just visited rather than return to the depot empty. The problem framework is developed on a two-terminal network where the origin of each customer's shipment is known. The continuous version of their model shows how the location of terminals can affect the decision of whether to backhaul, and determines which loads should be included in a backhaul, or return, trip. They show that both the location of customers throughout the region and the metric by which trucks' travel distances are computed will determine how to optimally construct backhaul loops.

The authors also give a discrete formulation of their problem in which each customer is represented explicitly. This model allows for a more general representation of travel distances between customers and depots (i.e., not necessarily on a grid or in a straight line). The objective is to maximize the total backhaul savings from customers served in backhaul loops, or, equivalently, to minimize total miles traveled. Because each backhaul loop is composed of one load from each terminal, each terminal must have the same number of customers in backhaul loops. The problem can be solved optimally with a greedy algorithm wherein one available customer from each terminal with the maximum backhaul savings is added at each iteration.
In subsequent work, Jordan (1986) extends the discrete two-terminal backhauling model to one with many terminals. The goal is to maximize the total reduction in empty miles while requiring that a truck visit only one terminal and balancing constraints be met (ensuring that all trucks will have a load to pick up).

A classic and extensively studied hubbing problem is found in air transportation, where airlines have created hubs to take advantage of the economies of scale of airplane size, offer more frequent service, and more thoroughly dominate the market in a particular geographic area. Hubbing has also been used in less-than-truckload (LTL) shipping. The freight is taken to an intermediate break-bulk terminal where it is consolidated with other loads. While this results in savings in truck miles and better utilization of equipment, there are disadvantages such as increased handling costs, increased delivery time, and the possibility of more circuitous routes.

Hubbing is also used in truckload shipping. Although this setting does not lend itself to consolidation, there is an advantage to hubbing. Truckload drivers are often subject to demanding schedules and low wages. Their schedules are due, in part, to dispatcher’s decisions, which must consider equipment availability and the need to minimize empty miles. Therefore, the time drivers spend away from home can be quite significant. This results in a high workforce turnover rate, in some cases over 100% a year. Because hubbing allows drivers to make shorter trips to and from a hub, rather than to and from a destination, it improves drivers’ conditions, specifically, the amount of time they are able to spend at home, thereby decreasing their turnover rate and the amount of time and money companies must spend on training.

Taha and Taylor (1994) present an integrated software system to evaluate hub-and-spoke operations for truckload operations. The user enters demand distribution data and selects a performance criterion such as directness of routes or driver utilization. Using software that contains both optimization and simulation modules, the user can evaluate strategic and operational decisions including the location of hubs, which hubs should be connected directly, and which hubs should serve which customers. Hub location is strongly influenced by the density of customers in a particular region, the distance between hubs, and the location of existing terminals. The software also determines driver allocation, operational driver rules, and routes.
1.2.3 Multimodal Models

General models addressing concerns in multimodal transportation have been developed. Crainic and Rousseau (1986) treat a multimode, multicommodity freight transportation problem in which they are concerned with determining and integrating decisions regarding the design of a service network, routing traffic over this network, and terminal operating policies. The problem is formulated as a mixed integer program in which the objective is to minimize the total operating and delay costs associated with servicing all demand. The decisions include the frequency of service to provide, and how classes of traffic are to be serviced. They employ a solution technique in which the problem is decomposed into two separate problems: (i) the problem of determining vehicle schedules, and (ii) the problem of servicing the freight given these schedules. The two subproblems are solved via a heuristic procedure and column generation, respectively.

Guélat, Florian, and Crainic (1990) treat a similar problem in which the objective function is given as a nonlinear expression of partial differentials, which represent the marginal cost of transporting a given product along a specific path. The Gauss-Seidel Linear Approximation solution approach is used.

1.2.4 Intermodal Operations

In the previous sections, we have discussed some relevant papers on the constituents of intermodal transportation, namely rail and trucking operations, as well as multimodal models. We now turn to problems that arise when these operations are combined within the framework of rail-truck intermodal transportation.

In the past decade, literature treating intermodal models has grown considerably. One of the earlier works (Morlok and Spasovic, 1990) treats only a part of the intermodal operation. The authors address the question of how substantial a reduction can be made in drayage costs without affecting the timeliness of pickups and deliveries. Currently, drayers are paid for the pick up and delivery of a full trailer or container. In addition, they are also paid, either directly or indirectly, for non-revenue activities such as tractor idling, empty movement (dead-heading), and tractors traveling without a trailer (bobtailing). This implies that the efficiency of tractor and driver use are integral factors in determining drayage rates. To this end, the authors suggest that the drayage operations be centrally planned
with respect to an entire terminal area. While this would require more information, namely all the consignees' and shippers' demands, it would result in increased tractor and trailer utilization. For example, empty trailers could be repositioned where they are demanded rather than being returned directly to the terminal after each use.

The authors treat terminal and rail operations and the cargo demand patterns as given. The relevant time constraints on trailer movements are the latest time a shipper can be provided with an empty trailer, the latest time a loaded trailer can be delivered to a rail terminal, and the latest time a container can arrive at the consignee. All containers are assumed to be identical.

A mathematical model of the drayage operations is used to optimize the efficiency of service for various system designs. It is dynamic in that it considers time-varying demands. It also takes into consideration the fact that trailers and tractors are separable units by modeling them as such. The model is formulated as an integer linear program and includes flows of trailer loads as well as supporting movements (e.g., dead-heading and bobtailling) and trailer repositioning. The goal is to minimize the total cost of tractor and trailer activities, subject to the following constraints: (i) on-time pick ups of loads at shippers, (ii) on-time deliveries of loads to consignees, (iii) tractor and trailer flow conservation, and (iv) nonnegativity and integrality.

The model is tested on data from a Conrail terminal in Pennsylvania. The results indicate that with the use of the model and centralized planning of drayage operations, a substantial savings (44% to 63%) in drayage costs can be realized. Additionally, various alternatives are tested, such as varying the pay structure of the drayers or the scheme for equipment leasing, which result in a further reduction in costs.

Unfortunately, most intermodal planning and coordination today are controlled by multiple retailers, who act as a middlemen between the shippers and the railroad. Because information is not readily shared, it may be difficult to implement a model such as the one described above. Morlok and Spasovic suggest, perhaps unrealistically, a reorganization of the intermodal system to alleviate this problem. Finally, they argue that savings in drayage costs can be passed on to customers, if desired, either as a reduction in rates, expansion into new markets and/or improvements in the quality of service.

Other intermodal models concentrate on the rail, or line haul, segment of the rail-truck
combination. Dial (1994) develops an integer programming model to decrease the trailer-on-flatcar costs incurred by United Parcel Service (UPS). The decision is whether to ship freight with a trailer owned by UPS, or to lease one from the railroad. Depending on the balance of trailers, both of those owned by UPS and those owned by the railroad, it can be more economical to exercise the latter option. The integer programming formulation is transformed to a minimum cost flow problem which is easy to solve.

Barnhart and Ratliff (1993) model intermodal routing by defining paths along which goods are transported from the origin to the destination either by rail and by truck, or solely via truck. They wish to determine how to ship these goods along a minimum cost path to meet demand requirements. The costs on the paths incorporate the transportation and inventory costs between locations. A shortest path algorithm can be used to determine a minimum cost solution. The model is extended to incorporate the case in which costs for over-the-rail transportation are modeled in terms of flatcars, rather than trailers, where each flatcar can accommodate at most two trailers. Another extension incorporates constraints in trailer scheduling, and the allows for ability of flatcars to accommodate various trailer types. Both extensions are solved with the use of matching algorithms.

Nozick and Morlok (1992, 1997) construct a comprehensive model for an intermodal rail-truck system. The objective of their model is to minimize the cost of delivery such that the movements are physically feasible and the goods are delivered on time. They address the movement of freight from the shipper to the intermodal terminal, along the line haul portion of the trip to another intermodal terminal, and then to the consignee. Additionally, the model can be used for determining the optimal fleet size and mix of equipment as well as the costs associated with providing various levels of service.

Their model is based on the following assumptions: First, all durations are expressed as a multiple of a basic time period. Secondly, shippers and consignees are grouped together at their respective terminals to reduce the computational burden of the problem. Individual consignees and shippers are considered at the drayage level but are not taken into account in the comprehensive intermodal model. Thirdly, it is assumed that shipments are available for pick up at the beginning of a time period, and the time by which that load must be delivered to the consignee is known. At most one activity, for example, the movement of a load from a shipper to a consignee or the movement of an empty car from one terminal to
another, can occur in a time period. Additionally, the train schedules are taken as given. Finally, it is assumed that the fleet is homogeneous within a vehicle class.

Rail equipment movements are depicted on a network. Each node represents a location at a particular point in time; equipment becomes available for use by shippers and consignees at the nodes. Flows on the arcs represent the movement of intermodal equipment such as trailers, containers, and flatcars. The objective is to minimize the total cost of a sequence of movements. This cost is composed of the following components: (i) satisfying shippers' demands in moving loads to consignees; (ii) repositioning empty trailers between intermodal terminals, and (iii) repositioning empty flatcars. Constraints include satisfying all demand within a given (prespecified) time frame, and ensuring trailer and flatcar conservation during the planning horizon.

The model is fairly comprehensive in that it treats a number of different decisions including the operation of the rail network, the movement of equipment on this network, and the drayage portion of the trip. The model includes all relevant equipment types and accounts for different levels of service among the operations. The authors mention a number of possible extensions to the model. First of all, it is possible to include more equipment (COFC equipment as well as, or instead of, TOFC equipment) or different sizes of equipment. Another modification would include imposing capacity constraints at the terminal. Changes in the fleet can be incorporated by the addition of new variables representing those trailers that leave and the new ones that are introduced into the system. The authors achieve results within 1% of the optimal solution by solving the linear programming relaxation of their problem, and applying a rounding heuristic.

Gorman (1998a, 1998b) considers the problem of providing train service on a set of predetermined paths while simultaneously routing shipments to meet service requirements, initially in the traditional rail boxcar operational setting. Other constraints, such as yard, line, and train capacity, are included. The formulation does not distinguish direct and indirect train service explicitly. Certain constraints, such as meeting the demand and adhering to train capacity constraints, may be violated with a penalty. Genetic and tabu-enhanced genetic searches are employed to determine a train schedule. Because the genetic search procedure does not necessarily perform well on larger problem instances, an enhanced tabu search procedure is adopted in which tabu search ideas are incorporated in a genetic algo-
rithm. This enhanced solution procedure provides results within 10% of an optimal solution when compared with an exact solution procedure for smaller problem instances, which include at most 18 origin-destination pairs. The exact solution procedure is performed on these problem instances under the assumption of origin-destination independence (i.e., the scenario in which a solution for each origin-destination pair can be obtained independently). The procedure is then compared to current operational procedures of a railroad assuming origin-destination interdependence, and the author concludes that the potential for savings, even within the confines of operating the current number of trains, is great. A case study is performed on Santa Fe’s intermodal operations using this algorithm, and similar conclusions are reached.

1.2.5 Models Possessing Related Mathematical Structure

We model our problem as a large integer program, which cannot be solved with a straightforward application of existing software. We develop specialized procedures to obtain provably good solutions. Much research has been devoted to finding efficient methods of generating good solutions to a variety of classes of integer programming problems. We now review some of the more relevant work. It should be noted that these problems are not derived from our application area. The relationship they share with our problem is one of some similarity in mathematical structure.

Our problem is modeled as a piecewise concave multicommodity flow problem. The objective function consists of a step function term which corresponds to incurring a setup cost for each scheduled train departure. A modest amount of research has been done on the fixed charge problem in which both a fixed cost and a variable cost are imposed on arc traversal. The constraint set consists of conservation of flow at the nodes, and the restriction of flow over arcs to those that are “used” (i.e., those arcs for which the fixed cost of traversal is incurred). We review recent contributions to the literature treating the classical fixed charge problem, keeping in mind that the constraint set in this class of problems differs from our structure because our arc capacities depend on the number of trains sent, and the capacity is shared among multiple commodities.

Palekar, Karwan, and Zionts (1987) evaluate several penalty methods for solving the fixed charge transportation problem. By relaxing integrality requirements on the binary variables, they transform the formulation into a transportation problem in which some of
the values for the binary variables are infeasible (fractional). Driebeek and Cabot-Erenguc
Penalties can be derived to approximate the deterioration in the objective function value
by restricting the relaxed variables to be binary. These penalties can be used to tighten
the bounds on the linear programming relaxation of the original problem. The authors
strengthen these penalties to improve solution times for this class of problems. In particular,
their improved penalty functions reduce the number of transportation problems that need
to be solved. Finally, they report the effect of problem size and cost structure on the ability
to obtain good solutions via these techniques.

Schaffer and O'Leary (1989) treat a special case of the fixed charge problem in which
the fixed charges are associated with the supply point. They use a procedure in which
the problem is separated into a transportation problem (without the binary fixed charge
variables) and a knapsack problem, which contains only the binary variables of the original
formulation. Based on the results from solving these subproblems, they modify the branch-
ing procedure to allow for faster detection of branches which can be pruned. They test this
optimal procedure against a heuristic procedure, which is applied to the non-partitioned
fixed charge problem. The heuristic procedure consists of examining adjacent extreme
points. The procedure is terminated upon finding no better local alternatives. Both the
exact and the heuristic procedures result in faster solution times than those achieved from
a straightforward implementation of software.

Lamar, Sheffi, and Powell (1990) derive an improved lower bound for the fixed charge
transportation problem. This lower bound is based on the linear programming relaxation of
the problem in which the logical constraints, which ensure no flow over “unused” arcs, are
aggregated over all commodities. Furthermore, it is shown that this lower bound converges
to the optimal objective function value for the original problem. Hochbaum and Segev
(1994) develop an efficient solution procedure for the fixed charge problem using an equiva-
 lent formulation augmented with redundant constraints. Lagrangian-based heuristics are
used to generate both a tight lower bound and a good feasible solution. Finally, Herrmann,
Ioannou, Proth, and Minis (1996) propose an iterative dual ascent approach to the fixed
charge problem with finite upper bounds on the arcs. The algorithm employs a labeling
method which has been used for the uncapacitated version of this problem. Improvements
in the integrality gap with this procedure are reported to be on the order of 4% for the
problems tested.

The following two references are examples of work that has been done on more general integer programming problems. The nature of this work is representative of many techniques used to solve “hard” integer programs. Specifically, carefully tailored decomposition and relaxation techniques are used to find good integer solutions, and valid inequalities are added to the constraint set where appropriate to restrict the search space.

Hallefjord and Storøy (1990) consider a binary integer program. They aggregate variables into mutually exclusive and collectively exhaustive sets. Primal and dual linear programming relaxations of the aggregated problem are solved and, subsequently, a heuristic procedure is employed to identify valid inequalities. When imposed on the original disaggregated formulation, this formulation yields a better bound than the linear programming relaxation to the original (unaggregated) problem. The exact design of the procedure, such as generating the valid inequalities, is problem-specific.

Rana (1992) develops a heuristic procedure for obtaining good solutions for mixed integer programming problems for a block-angular constraint matrix with additional coupling constraints. By removing coupling constraints through Lagrangian relaxation, the problem can be decomposed into several smaller subproblems, which are solved independently. A master problem is constructed using information from the solutions of the subproblems and solved via Lagrangian relaxation with subgradient optimization. This procedure iterates between the master problem and the subproblem until there is no change in the Lagrange multipliers of the master problem. The procedure is demonstrated on an airline routing problem.

Our problem fits into the class of fixed charge multicommodity flow problems in which the arc capacities are variable. A vast amount of literature treats the multicommodity flow problem with the arc capacities fixed. We review some recent literature on this type of problem structure here.

Aronson (1986) develops a heuristic shortest path algorithm embedded in a branch and bound algorithm, which results in good solutions to a special case of a multicommodity flow problem, namely the multiperiod assignment problem, i.e., the problem concerned with assigning $n$ people to $n$ jobs over a time horizon of $T$ periods. Dijkstra's algorithm is used to identify a shortest path for each of $n$ problems, where $n$ represents the number of activities.
Each of these problems has an embedded network which is obtained by eliminating the mutual capacity constraint restricting an activity to be assigned to at most one person in a time period.

Aggarwal, Oblak, and Vermuganti (1994) propose a heuristic method for obtaining good solutions to the multicommodity maximum flow and minimum cost flow problems in which the arc capacities are fixed. The heuristic is based on identifying a solution that maximizes flow along the arcs for the single commodity case in which common arc capacities are allocated among individual arcs. The solutions are collectively used as a solution for the multicommodity problem, and flows on the arcs are readjusted using dual information to reallocate arc capacity. The heuristic is shown to perform well on various instances of the equipment replacement problem.

Barnhart, Hane, and Vance (1996) present a solution methodology for integer multicommodity flow problems with fixed capacity constraints. They transform the traditional formulation into a path-based formulation, i.e., one in which paths from the source to the sink are explicitly enumerated. Associated with each path is a variable which takes a value of one if the path is chosen and zero otherwise. They apply a column generation procedure to the linear relaxation of this problem. This procedure constructs an initial set of paths to be included in the formulation, and the problem, termed the restricted master problem, is solved with a customized branch-and-price procedure as follows: Dual variables from the solution are used to solve a pricing problem from which nonbasic paths with the potential for improving the solution are identified. This pricing problem is equivalent to a shortest path problem for each commodity. If such paths are found, the master problem is again solved with these variables included. The procedure is repeated until no paths are found which have the potential to improve the solution to the original problem. An improvement in these computational results can be realized from the use of a more specialized procedure, the branch-and-price-and-cut procedure, in which valid inequalities, termed lifted cover inequalities, are added to the integer program.

We conclude this chapter with a brief overview of the dissertation. In Chapter 2, we present our integer programming formulation for minimizing rail operational costs, the underlying assumptions of our model, and an analysis of the problem structure. We establish that our piecewise-concave-cost multicommodity flow integer program is a large, NP-hard
problem containing thousands of variables and constraints for practical problem instances of moderate size. We also show that simple, intuitive solution procedures may provide poor solutions.

We empirically demonstrate the difficulty of our problem by solving a variety of instances with a straightforward application of conventional software (CPLEX 5.0). In all of these instances, the algorithm terminates prior to finding an optimal solution due to a memory limit. At the point of termination, it yields two pieces of information: (i) the best integer solution obtainable within the memory constraints, and (ii) a lower bound, or "guarantee" on the maximum deviation of the solution from its optimal value. As shown by the numerical results presented in Section 4.5, a gap between the objective function value of the best integer solution and the lower bound of about 25% is common for most problem instances.

In order to obtain improved performance, we explore the possibility of using two classical decomposition techniques, which are commonly applied to difficult integer programming problems. This analysis is presented in Chapter 3. We conclude, through analytical arguments, that Lagrangian relaxation, a constraint-based decomposition procedure, is not an effective solution technique. We then attempt to implement several variations of Bender's decomposition, a variable-based decomposition approach. We demonstrate that our formulation and problem structure are not suitable for this procedure.

Because our attempts at using established techniques are not successful, we develop a new decomposition approach, described in Chapter 4, which is based on partitioning the underlying network structure of our problem. We show that our procedure yields solutions that are about 5%-10% better than those provided from a straightforward implementation of CPLEX. Furthermore, we show that on average, we are able to obtain the solutions as quickly with our decomposition procedure as with a straightforward application of the software.

Although they represent an improvement over the original results, the solutions from the decomposition procedure deviate about 20% from the lower bound. Therefore, we develop another procedure to tighten the lower bounds and demonstrate empirically that the maximum difference between our solutions and the true optimal value is relatively small, about 5% on the average. The construction and validity of these improved lower bounds is discussed in Chapter 5. In Chapter 6, we implement simple heuristics designed to mimic
current operational procedures, and show that our solutions are generally about 10%-20% better than those currently achieved in practice. Figure 1.2 depicts the relative performance of the straightforward implementation of CPLEX, the novel approaches to improving the best integer solution and tightening the lower bound, and the heuristics designed to reflect current operational procedures.

In the next chapter, we present our integer programming formulation for optimizing intermodal rail operations, discuss its mathematical structure, and make some important observations concerning its solvability.
Chapter 2

Mathematical Formulation and Problem Structure

In this chapter, we provide the motivation for and assumptions used in our model, and then present our piecewise-concave-cost multicommodity flow integer program used to optimize the rail segment of intermodal transportation. We then discuss the mathematical structure and the nature of our problem, which will aid in the understanding of the rationale behind our new decomposition approaches, and the procedures for strengthening the bound on the objective function value which we present in Chapters 4 and 5.

Although the intermodal setting combines several transportation modes, e.g., truck and rail, our model concentrates on rail operations, assuming that there is adequate drayage capacity at all origins and destinations. We choose this as our focus, because of the greater potential for increasing timeliness of the entire intermodal journey by improving the on-time performance of the rail segment.

Our research was motivated by the train scheduling and container routing problem that we observed at the intermodal division of a major railroad. The railroad has several intermodal terminals on the west coast of the U.S., a single major hub in the west-central part of the U.S., and several other intermodal terminals east of the Mississippi River. The flow of traffic eastbound is greater than it is westbound, a situation faced by many U.S. railroads.

Much of the eastbound rail transport capacity is dedicated to moving sea cargo for major international shipping lines. The transoceanic transit time is fairly predictable and approximately one week. Remaining train capacity is utilized to service smaller customers within a few hours’ drive of an intermodal terminal. These customers typically use intermodal retail-
ers to coordinate the truck and rail movements for their goods. Intermodal retailers often reserve space on trains in advance, and then "sell" this space to their customers. The fact that intermodal retailers make reservations also contributes to the predictability of demand for the railroad. Overall, demand exhibits weekly patterns due to freighter schedules, and seasonal patterns due to factors such as traditional cycles in retail demand, and agricultural and manufacturing production.

The railroad offers several speeds (or levels) of service and charges a premium for faster (promised) delivery. Trains may be sent directly from an origin intermodal terminal to the destination terminal without stopping at a hub, providing the fastest available service. Alternatively, trains carrying containers bound for several destinations may be sent to a hub, where containers with different origins but bound for a common destination are consolidated onto a train outbound from the hub. This consolidation activity can lead to a delay of up to several days. Part of the delay is due to the need to transfer containers or to reposition rail cars between trains. Further delays also occur when inbound and outbound schedules are not coordinated.

Each train has a limited capacity, where the capacity depends upon the power of the locomotives and the terrain over which the train must travel. Typically, for each transportation segment, locomotive capacity is determined in advance on the basis of demand forecasts, and from this, the train capacity is derived, expressed in terms of number of containers.

Yard storage space for containers waiting to be shipped, awaiting a transfer at the hub, or waiting to be picked up, is limited at all terminals. As the number of containers in storage increases, containers are stacked higher and more densely. This increases the time required to retrieve a container and places a further burden on material handling equipment which may already be a bottleneck.

From our observations, the train schedules and container routing decisions do not appear to be affected strongly, if at all, by what speed of delivery has been promised, or what rate has been charged to the customer. This motivated us to investigate how to schedule trains and route containers to achieve on-time delivery at minimum cost.

We address a short-term, finite-horizon, discrete-time scheduling problem for the linehaul portion of the intermodal trip, that is, when to schedule direct trains, when to schedule
indirect trains (i.e., trains that travel through a hub), and which containers to send on each train. Given container demands differentiated by origin, destination, arrival date at origin, and due date, the objective is to determine a train schedule and container shipment plan to minimize operational costs and inventory holding costs while meeting on-time delivery requirements and adhering to train capacity restrictions.

For convenience, we refer to a time period as a “day”. We assume that transit and hub delay times are deterministic, constant across time, and that both transit times and delays at the hub are expressed as an integral number of time periods. In practice, transit times are rarely predictable, but because time is expressed in terms of days, not minutes or hours, there is implicit slack in the schedule to accommodate unforeseen events.

The operational costs incurred by the rail company consist of both a fixed-charge, or “setup” cost, and a variable, or per unit, component. The fixed-charge component includes operators’ wages and costs of assembling the train. We assume that equipment is available where needed. We assume that the capacity of a train on each transportation segment is known, and that containers are homogeneous in terms of their use of train capacity. These assumptions are reasonably accurate reflections of the situation that motivated our study. We also assume that each train on a segment incurs the same setup cost, and that there is no limit on the number of trains that can be sent each day. Because train operator labor constitutes a major portion of the setup cost, and the labor cost for each additional train is roughly the same provided sufficient labor is available, the first assumption is fairly mild. On the other hand, locomotive capacity may be limited with respect to location and time, so it may not always be possible to schedule as many trains as desired. In the concluding chapter, we discuss how this limitation can be incorporated into our model.

Variable costs consist of three main components: (i) transportation costs, such as fuel, oil, and track maintenance, (ii) handling costs incurred for moving containers on and off the rail cars, or for repositioning the cars at an intermediate terminal, and (iii) yard storage costs associated with holding containers in inventory at the origin, destination, or at an intermediate terminal. We assume that the transportation costs are constant over time and that they depend only on the origin-destination pair. The assumption of constant transportation costs over time is quite reasonable over a short-horizon (e.g., a week or two), and the assumption that costs depend only on the origin-destination pair is consistent
with our assumption regarding homogeneity of containers in terms of their use of transport capacity.

Handling costs for moving containers or repositioning rail cars depend more strongly on the equipment utilized for such operations than they do upon the container itself, or its origin or destination. Thus, the assumption of constant handling costs at the hub is quite mild.

Inventory costs consist primarily of yard storage costs and the opportunity cost of having a container unavailable for use, as the opportunity cost of capital for in-transit goods is borne either by the shipper or by the consignee. Yard storage costs in our model are assumed to be equal for all containers, and these costs do not differ with location or time. We assume that customers will accept early deliveries, so no inventory is held at the destination. Generalizations to consider other linear cost structures and delivery acceptance rules are straightforward.

To summarize, our problem is to choose daily train schedules and container routes over a short horizon, so as to achieve on-time delivery at minimum cost, where the total cost consists of a fixed charge per train with a given capacity, a variable transportation cost per container, both of which are dependent on the origin and destination, handling costs per container dependent upon the location, and inventory holding costs for containers held at any terminal prior to their arrival at the destination.

Note that our model is intended to aid in establishing schedules in the “bottleneck” direction and does not address locomotive or empty container repositioning. (These issues have been studied for the case of a fixed train schedule by Nozick and Morlok, 1997.) In addition, our requirement for on-time delivery is a hard constraint. In Chapter 7, we discuss how this constraint can be relaxed.

Although this problem (or variations of it) have been mentioned in the literature, no solution approach has been proposed to date. Four aspects of the problem that make it especially difficult are: (i) the possibility of sending more than one train on each segment each day, (ii) the option of sending both direct and indirect trains, (iii) the dynamic arrival of containers, and (iv) distinct due dates for different customer orders.

We now turn to a mathematical formulation of the problem.
2.1 Mathematical Formulation

We define below the following indices, parameters, and decision variables. It should be noted that the indexing of variables is specific and consistent throughout the document, e.g., the index $i$ always represents an index on the set of origins.

The subscripts in the model are as follows:

$i = \text{index of origins}$

$j = \text{index of hubs}$

$k = \text{index of destinations}$

$t = \text{index of days in the time horizon}$

$l = \text{index of level of service, i.e., the due date of the container at the destination}$

The parameters in the model are as follows:

$\alpha_{ik} = \text{direct transportation time between origin } i \text{ and destination } k$

$\beta_{ij} = \text{transportation time between origin } i \text{ and hub } j$

$\gamma_{jk} = \text{transportation time between hub } j \text{ and destination } k$

$\delta_j = \text{delay time incurred from passing through hub } j$

$T = \text{length of the planning horizon}$

$C = \text{capacity of a train (number of containers)}$

$h = \text{holding cost of a container ($/container/day$)}$

$c_{ik} = \text{variable unit cost of transporting a container directly from origin } i \text{ to destination } k$

$c_{ijk} = \text{variable unit cost of transporting a container from origin } i \text{ via hub } j \text{ to destination } k$

$Sd_{ik}^o = \text{fixed cost of running a train directly between origin } i \text{ and destination } k$, including labor and train assembly
\[ s_{ij}^c \] = fixed cost of running a train indirectly between origin \( i \) and hub \( j \), including labor and train assembly

\[ s_{jk}^h \] = fixed cost of running a train between hub \( j \) and destination \( k \), including labor and train assembly

\[ l_i^c \] = cost of placing a container on the train at origin \( i \)

\[ l_j^h \] = cost of rearranging a container at hub \( j \)

\[ l_k^d \] = cost of removing a container from the train at destination \( k \)

\[ b_{ikt} \] = the number of containers that arrive at origin \( i \) on day \( t \) bound for destination \( k \) with a due date of time \( l \)

The decision variables are as follows:

\[ I_{ik}^{cl} \] = number of containers held at origin \( i \) at time \( t \), which are required to be at destination \( k \) by time \( l \)

\[ I_{ij}^{ht} \] = number of containers originating at \( i \) and held at hub \( j \) at time \( t \), which are required to be at destination \( k \) by time \( l \)

\[ I_{jk}^{hl} = \sum_i I_{ij}^{ht} \] = number of containers held at hub \( j \) at time \( t \), which are required to be at destination \( k \) by time \( l \)

\[ x_{ik}^{cl} \] = number of containers shipped directly from origin \( i \) at time \( t \), which are required to be at destination \( k \) by time \( l \)

\[ x_{ij}^{kt} \] = number of containers shipped from origin \( i \) at time \( t \) to hub \( j \), which are required to be at destination \( k \) by time \( l \)

\[ u_{ij}^{kt} \] = number of containers which originated at \( i \) and are shipped at time \( t \) from hub \( j \) to destination \( k \), where they are due by time \( l \)

\[ x_{ik} \] = the number of trains sent directly from origin \( i \) to destination \( k \) at time \( t \)

\[ x_{ij} \] = number of trains sent from origin \( i \) to hub \( j \) at time \( t \)

\[ x_{jk} \] = number of trains sent from hub \( j \) to destination \( k \) at time \( t \)
The formulation, then, is as follows:

\[(P): \min \quad Z = \]

\[ \sum_{ikt} h \cdot I^0_{ikt} + \sum_{jkt} h \cdot I^h_{jkt} + \sum_{ijklw} h \cdot x_{ijklw} + \sum_{ikt} c_{ik} \cdot x_{ikt} + \sum_{ikt} c_{ijk} \cdot x_{ijklt} + \sum_{ikt} l^0_i \cdot x_{ikt} + \sum_{ijkt} l^h_j \cdot u_{ijkt} + \sum_{ikt} l^d_k \cdot x_{ikt} + \sum_{ikt} l^d_k \cdot u_{ijk}\]

\[+ \sum_{ikt} s^0_{ik} \cdot z_{ikt} + \sum_{ijt} s^o_{ij} \cdot x_{ijt} + \sum_{jkt} s^h_{jk} \cdot z_{jkt}\]

subject to

\[I^0_{ikt} = 0 \quad \forall i, k, t = 1, t + \alpha_{ik} \leq l < t + \beta_{ij} + \gamma_{jk} + \delta_j\]  
(2.1)

\[b_{ikt} = I^0_{ikt} + x_{ikt} + \sum_j x_{ijkt} \quad \forall i, k, t = 1, l \geq t + \beta_{ij} + \gamma_{jk} + \delta_j\]  
(2.2)

\[b_{ikt} = -I^0_{ikt} + x_{ikt} \quad \forall i, k, t > 1, t + \alpha_{ik} \leq l < t + \beta_{ij} + \gamma_{jk} + \delta_j\]  
(2.3)

\[b_{ikt} = I^0_{ikt} - I^0_{ikt(1)} + x_{ikt} + \sum_j x_{ijkt} \quad \forall i, k, t > 1, l \geq t + \beta_{ij} + \gamma_{jk} + \delta_j\]  
(2.4)

\[I^h_{jk(t+\beta_{ij}+\delta_j)} = 0 \quad \forall j, k, t = 1, l = t + \gamma_{jk}\]  
(2.5)

\[I^h_{jk(t+\beta_{ij}+\delta_j)} = \sum_i x_{ijk} - \sum_i u_{ijk(t+\beta_{ij}+\delta_j)} \quad \forall j, k, t = 1, l > t + \gamma_{jk}\]  
(2.6)

\[-I^h_{jk(t+\beta_{ij}+\delta_j-1)} = \sum_i x_{ijk} - \sum_i u_{ijk(t+\beta_{ij}+\delta_j)} \quad \forall j, k, t > 1, l = t + \gamma_{jk}\]  
(2.7)

\[I^h_{jk(t+\beta_{ij}+\delta_j)} = I^h_{jk(t+\beta_{ij}+\delta_j-1)} + \sum_i x_{ijk} - \sum_i u_{ijk(t+\beta_{ij}+\delta_j)} \quad \forall j, k, t > 1, l > t + \gamma_{jk}\]  
(2.8)

\[b_{ikt} = x_{ikt} \quad \forall i, k, t = 1, t + \alpha_{ik} \leq l < t + \beta_{ij} + \gamma_{jk} + \delta_j\]  
(2.9)

\[x_{ikt} + \sum_j x_{ijkt} \leq b_{ikt} \quad \forall i, k, t = 1, l \geq t + \beta_{ij} + \gamma_{jk} + \delta_j\]  
(2.10)

\[\sum_{w=1}^t b_{ikw} = \sum_{w=1}^t x_{ikw} \quad \forall i, k, t > 1, l \geq t + \beta_{ij} + \gamma_{jk} + \delta_j\]  
(2.11)

\[\sum_{w=1}^t x_{ikw} + \sum_{j=1}^t \sum_{w=1}^t x_{ijkw} \leq \sum_{w=1}^t b_{ikw} \quad \forall i, k, t > 1, l \geq t + \beta_{ij} + \gamma_{jk} + \delta_j\]  
(2.12)
\[ x_{ijkl} = u_{ijk(t+\beta_{ij}+\delta_j)}l \quad \forall i, j, k, t = 1, l = t + \gamma_{ik} \quad (2.13) \]

\[ u_{ijk(t+\beta_{ij}+\delta_j)}l \leq x_{ijkl} \quad \forall i, j, k, t = 1, l > t + \gamma_{ik} \quad (2.14) \]

\[ \sum_{w=1}^{t} x_{ijklw} = \sum_{w=1}^{t} u_{ijk(w+\beta_{ij}+\delta_j)}l \quad \forall i, j, k, t > 1, l = t + \gamma_{ik} \quad (2.15) \]

\[ \sum_{w=1}^{t} u_{ijk(w+\beta_{ij}+\delta_j)}l \leq \sum_{w=1}^{t} x_{ijklw} \quad \forall i, j, k, t > 1, l > t + \gamma_{ik} \quad (2.16) \]

\[ \sum_{l} x_{iktl} \leq C * z_{ikt} \quad \forall i, k, t \quad (2.17) \]

\[ \sum_{kl} x_{ijklt} \leq C * z_{ijt} \quad \forall i, j, t \quad (2.18) \]

\[ \sum_{il} u_{ijklt} \leq C * z_{jkt} \quad \forall j, k, t \quad (2.19) \]

\[ I_{iktl}, \; x_{iktl} \geq 0 \text{ and integer} \quad \forall i, k, t, l \quad (2.20) \]

\[ I_{jkl}^h, \; x_{ijkl} \geq 0 \text{ and integer} \quad \forall j, k, t, l \quad (2.21) \]

\[ x_{ijkl}, \; u_{ijkl} \geq 0 \text{ and integer} \quad \forall i, j, k, t, l \quad (2.22) \]

\[ z_{ikt} \geq 0 \text{ and integer} \quad \forall i, k, t \quad (2.23) \]

\[ z_{ijt} \geq 0 \text{ and integer} \quad \forall i, j, t \quad (2.24) \]

\[ z_{jkt} \geq 0 \text{ and integer} \quad \forall j, k, t \quad (2.25) \]

The objective function contains the following terms: the inventory holding cost at the origin and at the hub; the transportation cost of directly and indirectly shipped goods; the handling cost at the origin for both direct and indirect shipments, the handling cost at the hub for indirect shipments, and the handling cost at the destination for direct and indirect shipments; and finally, the setup cost at the origin for direct trains and trains bound for a hub, and the setup cost at the hub for indirect trains.

In our analysis, we assume only one hub, although the formulation is written for the more general case in which there may be multiple hubs, and each container may pass through at most one hub. We also assume that direct travel time between an origin and a destination is strictly less than the total transit and delay time for a container shipped indirectly, i.e., \( \alpha_{ik} < \beta_{ij} + \gamma_{jk} + \delta_j \).

Constraints (2.1) through (2.4), and (2.5) through (2.8) represent inventory balance for the origins and hubs, respectively. They also ensure that inventory is not held if it must be
sent out to satisfy due date requirements. The four types of constraints in each set result from the need to differentiate between the initial and subsequent time periods, and between schedules with and without slack time.

Constraints (2.9) through (2.12) are level-of-service constraints at the origin. They ensure that all containers arrive at their respective destinations on time. The constraints also restrict the containers sent from the origin to those available to be shipped. If there is no slack time in the schedule, the constraints dictate that the shipment be made immediately. Again, there are four types of constraints in each set.

Constraints (2.13) through (2.16) serve as level of service constraints at the hub. These constraints are analogous to the level of service constraints at the origin.

Constraints (2.17) require that for all origins, destinations, and time periods, the number of containers sent on direct trains must not exceed the total capacity of the trains departing. Likewise, constraints (2.18) and (2.19) ensure that train capacity is not exceeded on trains bound for the hub and trains leaving the hub, respectively.

Finally, nonnegativity and integrality constraints are imposed on all decision variables.

2.2 Network Description and Formulation

We can represent the problem using a network. We describe two equivalent representations of our problem, a single-commodity network in which only one commodity type may flow on each arc, and a multi-commodity network in which multiple commodities may flow on each arc, where a commodity is defined by its origin, destination, arrival date at the origin, and due date at the destination.

The single-commodity network is a set of nodes and arcs, where the nodes represent the physical location of a commodity at a given point in time, and each arc carries the flow of a single commodity through time and space. Figure 2.1 illustrates a single commodity flow network with two origins, one hub, two destinations, two time periods, and two levels service. For simplicity, the travel time on this graph is assumed to be instantaneous so that a level of service of i (ii) can be honored by sending a container out on day 1 (day 2). Furthermore, handling time at the hub is assumed to be instantaneous. In the general case, time indices for the nodes and the arcs connecting the nodes must reflect the true transit and delay times.
Observe that in addition to the supersource and supersink, there are three types of nodes: (i) a set of nodes, one for each (origin, time period, commodity) triple, (ii) another set of nodes, one for each (hub, time period, commodity) triple, and (iii) a third set of nodes, one for each (destination, time period, commodity) triple. Each feasible path in the network passes through one or more nodes in the first and last sets, and passes through the second set of nodes only if the shipment is indirect. If a shipment is never held in inventory, it will pass through exactly one node in each relevant set.

The arcs from the supersource to nodes in the first set have upper and lower capacity limits equal to the number of container arrivals of the commodity at the designated origin and time period. The arcs from nodes in the first set to nodes in the third set represent direct container shipments, and the arcs from the nodes in the first set that pass through one or more nodes in the second set enroute to nodes in the third set represent paths along which indirectly shipped containers flow. From each node in the third set for which the arrival and due dates coincide, containers flow into a supersink. The corresponding arcs have equal upper and lower bounds, which correspond to the demand for containers of each commodity type.

In addition to arcs that represent the movement of containers from one location to another, inventory arcs link one origin, hub, or destination node for a given commodity type with the corresponding node in the following time period. Container flows on these arcs represent storage of containers at a location from one time period to the next.

Per unit handling and transportation costs are assessed for movements between the origins, hubs, and destinations, and inventory holding costs are assessed along arcs between sequential time periods at the same location. There are no costs on the arcs emanating from the supersource or terminating at the supersink.

In addition to the problem attributes that are represented on the single commodity network, train costs and capacity constraints must be included. Note that these details cannot be captured explicitly using a single-commodity network representation. Because more than one commodity may use a single train, we could impose so-called “bundle constraints” on the total flow across all arcs which represent container shipments from one location to the next at a given point in time, if the number of trains associated with the bundle constraint is known.
Figure 2.1: Representation of the Single Commodity Network

ORIGINS: A, B
DESTINATIONS: C, D
LEVELS OF SERVICE: i, ii

--- = direct shipments
-- = indirect shipments
- - - = inventory
The multicommodity network is a simplified version of the single commodity network in that nodes are not distinguished by commodity type. Thus, nodes are differentiated only by location and by time period. Flows on the arcs now correspond to feasible container movements between two locations (i.e., an origin and a destination, an origin and a hub, or a hub and a destination) at a given time period, or the storage of inventory at a given location from one time period to the next. Different commodities that travel between the same two locations at the same time flow on the same arc. Assuming that, for each relevant location pair and time period, the capacity of all trains is the same, the total number of trains traveling on the arc is simply the ceiling of the total number of containers (of any commodity) traveling on that arc divided by the train capacity. Fixed costs associated with using the required number of trains are assessed based on container flows along that single arc. Capacity constraints for a given train are also imposed as an upper bound on the corresponding arc.

Regardless of whether we use the single commodity or multicommodity flow representation, the train costs are assessed on and capacity constraints are imposed on the total related flow for multiple commodities (either over several arcs, each representing flow of a single commodity, or over a single arc, along which multiple commodities flow). Thus, our problem is a multiple-setup, multicommodity flow problem. The difficulty in solving such problems will be discussed subsequently. For a thorough introduction to network flow problems, see Ahuja, Magnanti, and Orlin (1993), and Bazaraa, Jarvis, and Sherali (1990).

From the above discussion, we can construct a new formulation by modeling the container flow constraints as conservation of flow of containers at each node. With this in mind, we can modify our formulation given in Section 2.1 as follows:

Let us define the additional notation:

\[ I_{iktl}^d \] = inventory at destination \( k \) at time \( t \) due at \( l \) which originated from origin \( i \)
\((P_N)\): \(\min Z\)

subject to

\[
x_{ijkl} = u_{ijk}(t + \beta_{ij} + \delta_j) \quad \forall i, j, k, t = 1, l = t + \gamma_{ij} \tag{2.26}
\]

\[
I^h_{ijkl} = x_{ijkl} - u_{ijk}(t + \beta_{ij} + \delta_j) \quad \forall i, j, k, t = 1, l > t + \gamma_{jk} \tag{2.27}
\]

\[
-I^h_{ijkl} = x_{ijkl} - u_{ijk}(t + \beta_{ij} + \delta_j) \quad \forall i, j, k, t > 1, l = t + \gamma_{jk} \tag{2.28}
\]

\[
I^h_{ijkl} = I^h_{ijkl} + x_{ijkl} - u_{ijk}(t + \beta_{ij} + \delta_j) \quad \forall i, j, k, t > 1, l > t + \gamma_{jk} \tag{2.29}
\]

\[
b_{ikl}(t - \alpha_{ik}) = x_{ikt}(t - \alpha_{ik}) \quad \forall i, k, t = 1 + \alpha_{ik}, t \leq l < t + \beta_{ij} + \gamma_{jk} + \delta_j - \alpha_{ik} \tag{2.30}
\]

\[
I^d_{ikt} = x_{ikt}(t - \alpha_{ik}) + \sum_j u_{ijk}(t - \gamma_{jk})l \quad \forall i, k, t = 1 + \alpha_{ik}, l \geq t + \beta_{ij} + \gamma_{jk} + \delta_j - \alpha_{ik} \tag{2.31}
\]

\[
\sum_{w=1}^{l - \alpha_{ik}} b_{ikwl} = I^d_{ikt} + x_{ikt}(t - \alpha_{ik}) \quad \forall i, k, t > 1 + \alpha_{ik}, t \leq l < t + \beta_{ij} + \gamma_{jk} + \delta_j - \alpha_{ik} \tag{2.32}
\]

\[
I^d_{iktl} = I^d_{ikt} + x_{ikt}(t - \alpha_{ik}) + \sum_j u_{ijk}(t - \gamma_{jk})l

\forall i, k, t > 1 + \alpha_{ik}, l \geq t + \beta_{ij} + \gamma_{jk} + \delta_j - \alpha_{ik} \tag{2.33}
\]

\[(2.17), (2.18), (2.19)\]

All variables are restricted to be nonnegative and integer.

The objective function remains the same as given in Section 2.1. Here, constraints (2.1) through (2.4) can be interpreted as conservation of flow at the origin. As in the previous formulation, the various constraints are differentiated depending whether the condition is for the initial or a subsequent time period, and whether there is slack time in the schedule or not. Constraints (2.26) through (2.29) correspond to conservation of flow at the hub nodes. Constraints (2.30) through (2.33) correspond to conservation of flow at the destination nodes. Finally, train capacity (2.17) through (2.19), and nonnegativity and integrality requirements hold.

Mathematically, the two formulations are virtually identical in constraints (2.1) through (2.8), the inventory constraints at the origin and hub in the first formulation, and the conservation of flow at the origin and hub, in the network formulation, respectively. However,
note that we now differentiate inventory at the hub by origin. The level of service constraints at the origin and the hub in the initial formulation, constraints (2.9) through (2.16), correspond to the conservation of flow constraints at the destination with a variable substitution \((u_{ijkl} \text{ for } x_{ijkl})\) and the replacement of the slack in the inequality constraints with a new (slack) variable, \(I'_{ikt}\), the inventory held at the destination. Train capacity, nonnegativity, and integrality requirements hold as before.

Although the two formulations are mathematically equivalent, the network formulation may be preferred because it allows us to take advantage of the structure of the problem as a network flow with side constraints, i.e., train capacity constraints. The special structure of this multicommodity flow problem will be discussed subsequently.

2.3 Problem Size

Depending on whether one considers the initial formulation or the more characteristic “network with side constraints” formulation, the problem has a slightly different dimension — in the latter case, there are more variables and fewer constraints. In this discussion, we refer to the network formulation.

Let us define the following parameters:

- \(i\) = the number of origins
- \(j\) = the number of hubs
- \(k\) = the number of destinations
- \(t\) = the number of days in the time horizon
- \(l\) = the number of speeds of service offered

Note that the value used for \(l\) here will be an approximation, because it may differ depending on the origin-destination pair. Then the number of variables for our problem is given as:

\[
3iktl + 3ijk + ikt + it + jkt
\]

and the number of constraints is approximately:

\[
ikt + ijkt + ikt + it + jkt
\]
A small problem has two origins, two destinations, one hub, seven days in a time horizon, and three levels of service. A medium-sized problem might have four origins and four destinations, one hub, seven days in the time horizon, and five levels of service. Hence, a small problem would have 560 general integer variables and approximately 224 constraints. A medium-sized problem would have 3528 integer variables and approximately 1288 constraints. Larger problems with six origins, one hub, six destinations, seven days in the time horizon, and five levels of service would contain 7896 integer variables and approximately 2856 constraints. The size is one indication that our problem is difficult to solve, as will be discussed below.

2.4 Nature of the Problem

Our problem has an embedded network structure. Specifically, the container flow constraints (2.1) through (2.4), (2.26) through (2.29), and (2.30) through (2.33) constitute a network. For any linear objective function, this reduced constraint set will provide an integral optimal solution given integer data (i.e., integer container arrivals) without the integrality requirements imposed. Hence, this problem is as easy to solve as a linear program. However, the network structure is destroyed with the addition of the train variables. These variables are associated both with a cost for their use (i.e., a fixed cost if there is positive flow on an arc), and with the capacity constraints (2.17), (2.18), and (2.19). For a fixed train schedule our problem is equivalent to the directed multicommodity network flow problem, which is recognized to be NP-complete (Garey and Johnson 1979). Because this network flow problem is a special case of our problem, we conclude that our problem is at least NP-complete, and possibly NP-hard. In either case, there is no known algorithm that will provide a solution in polynomial time. For further discussion of complexity theory and the complexity of integer programming problems, see Nemhauser and Wolsey (1988).

Because the optimal solution may not be consistent with intuition, there is no simple solution procedure. In fact, containers are not necessarily placed on the next outbound train according to their level of service. The different operational costs of direct and indirect trains as well as the different travel times make it difficult to determine the cost and efficiency tradeoffs of direct and indirect shipments ahead of time. This difficulty is accentuated when the cost and travel time differentials become more pronounced between direct and indirect
<table>
<thead>
<tr>
<th>Number of containers</th>
<th>origin</th>
<th>destination</th>
<th>level of service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>A</td>
<td>C</td>
<td>8</td>
</tr>
<tr>
<td>75</td>
<td>A</td>
<td>C</td>
<td>9</td>
</tr>
<tr>
<td>75</td>
<td>A</td>
<td>D</td>
<td>9</td>
</tr>
<tr>
<td>50</td>
<td>A</td>
<td>D</td>
<td>10</td>
</tr>
<tr>
<td>125</td>
<td>B</td>
<td>C</td>
<td>9</td>
</tr>
<tr>
<td>75</td>
<td>B</td>
<td>D</td>
<td>10</td>
</tr>
<tr>
<td>Day 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>A</td>
<td>C</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 2.1: Example Problem Data, I

trains.

Consider the following example with two origins, one hub, two destinations, and two time periods. Suppose the direct travel time between any origin and any destination is four days, and the indirect travel time is seven days. Train capacity is 200 containers. Container arrivals for both days along with the origin, destination, and level of service are given in Table 2.1.

Assume that we have fairly accurate information as to the arrivals of containers over a given time horizon. Let us consider the “intuitive” method of assigning containers to the next outbound train based on their level of service. The 50 containers originating at A and bound for C with level of service 8 would be assigned first, followed by the 75 containers with the same origin and destination and a level of service 9. Because train capacity is 200 and time permits these containers to be sent indirectly, the 75 containers at origin A bound for D by day 9 could be assigned to the same train. In this way, a full indirect train could be sent out on day 1. The remaining 50 containers at A could be sent out on a partially full train or held at the origin for more arrivals. Similar decisions could be made on day 1 at origin B, and on day 2.

If, however, a more counterintuitive approach is taken in which the 50 containers bound for C with the earliest due date are held at A for one day, these containers can be matched with the 150 containers arriving at the same origin on day 2 and sent on a direct train. The remaining day 1 arrivals can be sent on indirect trains from each origin and re-sorted at the hub according to their destination. In this way, all trains are sent full, and all levels of service are met. In fact, the 50 containers that arrived at origin A on day 1 actually arrive at their destination earlier than they would have had they been sent out on day 1 (but on
an indirect train, as initially suggested).

Although this small problem can be solved by inspection, the example illustrates that a simple solution procedure is not possible in general since the tradeoff between direct and indirect trains is not easy to assess. While direct trains travel more quickly and incur less setup and handling cost, lower traffic levels may suggest that they be sent less frequently or not filled to capacity. In this case, it may be more economical to schedule indirect trains. As the number of origins and destinations increases and operational costs change, a good solution becomes hard to recognize.
Chapter 3

Classical Decomposition Techniques

In this chapter, we describe two classical decomposition procedures used to solve "difficult" integer programs. We apply them to our formulation, and demonstrate that our implementations of these classical procedures are ineffective in solving our problem.

3.1 Lagrangian Relaxation

Let \( c \) be a \( 1 \times n_1 \) row vector of objective function coefficients. Let \( x \) be a \( n_1 \times 1 \) vector of decision variables. Let \( A \) and \( D \) be \( m_1 \times n_1 \) and \( m_2 \times n_1 \) matrices of constraint coefficients for \( x \), respectively. Finally, let \( b \) and \( e \) be the vector of right hand side constraint constants of dimension \( m_1 \times 1 \) and \( m_2 \times 1 \), respectively.

Lagrangian relaxation is a procedure used to aid in the solution of "difficult" integer and mixed integer programs of the following form, for example, where we denote any unindexed vectors of decision variables in bold:

\[
(G): \quad \min c x
\]

subject to

\[
A x = b \quad \text{(3.1)}
\]

\[
D x \leq e \quad \text{(3.2)}
\]

\[
x \geq 0, \quad x \text{ integer} \quad \text{(3.3)}
\]

Suppose we consider the \( m_2 \) constraints given by (3.2) to be "troublesome." To apply Lagrangian relaxation to \((G)\), we define a nonnegative vector of multipliers \( \rho \) with dimension
1 \times m_2, and solve the following so-called relaxed problem, which necessarily provides a lower bound on the objective function value for the unrelaxed problem (G):

$$(G_D(\rho)) : \quad \min c x + \rho(D x - e)$$

subject to

$$Ax = b \quad (3.4)$$

$$x \geq 0, \quad x \text{ integer} \quad (3.5)$$

Observe that we have removed constraints (3.2) and brought them into the objective function (i.e., we dualize $m_2$ of the constraints), penalizing their violation. The term $\rho(D x - e)$ represents the penalty to the objective function value derived from violating the dualized (relaxed) constraints.

We would like to solve the following Lagrangian dual problem to find $\rho$ to obtain the best lower bound:

$$(G_D) : \quad \max_{\rho} (G_D(\rho))$$

subject to

$$\rho \geq 0 \quad (3.6)$$

In practice, this problem is not solved directly, but by searching for good multipliers using subgradient optimization (Held, Wolfe, and Crowder, 1974) or specialized multiplier adjustment procedures. If we are able to find $\rho$ such that the corresponding solution to $(G_D(\rho))$, $x^*(\rho)$, is feasible and satisfies complementary slackness conditions:

$$\rho[D x^*(\rho) - e] = 0$$

the solution $x^*(\rho)$ is optimal for the original problem.

3.1.1 Lagrangian Relaxation Applied to the Original Problem

Our formulation has a network structure with side constraints representing the capacity restriction for each train. Let us use notation defined above in addition to the following notation: Let $z$ be an $n_2 \times 1$ column vector of decision variables. Let $f$ be a $1 \times n_2$ row vector
of objective function coefficients. Let $F$ be an $m_2 \times n_2$ matrix of constraint coefficients for $z$. Then a generic version of our problem appears as follows:

$$(G') : \quad \min c x + f z$$

subject to

$$Ax = b \quad (3.7)$$
$$Dx + Fz \leq e \quad (3.8)$$
$$x, z \geq 0, \ z \ integer \quad (3.9)$$

Suppose we consider the $m_2$ constraints given by (3.8) to be "troublesome", i.e., those constraints that correspond to the capacity constraints on our original problem. To apply Lagrangian relaxation to $(G')$, we must solve this relaxed problem:

$$(G'_D(\rho)) : \quad \min c x + f z + \rho(Dx + Fz - e)$$

subject to

$$Ax = b \quad (3.10)$$
$$x, z \geq 0, \ z \ integer \quad (3.11)$$

We did not pursue this approach because we did not expect it to produce improved results over a straightforward implementation of CPLEX, or even a feasible solution. By inspection, we note that for any optimal solution to $(G'_D(\rho))$ for nonnegative objective function coefficients $f$, $z^* = 0$. Alternatively, if any coefficient on $z$ in the objective function is negative, $(G'_D(\rho))$ is unbounded. Moreover, even without cause to suspect that the Lagrangian problem would not yield a feasible solution for the original problem, because the constraints in the relaxed problem constitute a network, for any $\rho$, the solution has the integrality property. Thus, the lower bound, and the best lower bound for our problem (using the value of $\rho$ that maximizes $(G_D(\rho))$ can be no greater than that obtained from a linear programming relaxation of our original problem (Geoffrion (1974), Fisher (1985)). Although this lower bound can be found very quickly (i.e., in a few seconds), it is necessarily the same as the bound from linear programming relaxations found with conventional integer programming software, which we discuss in more detail in a subsequent section.
A second Lagrangian approach for our problem consists of relaxing some or all of the network constraints, rather than the train capacity constraints. Let the constraints given in (3.7) represent the network constraints. Dualizing these constraints with multipliers $\phi$ we obtain:

$$(G_D^\prime(\phi)) : \quad \min cx + fz + \phi(Ax - b)$$

subject to

$$Dx + Fz \leq e \quad (3.12)$$

$$x, z \geq 0, \text{ integer} \quad (3.13)$$

where we are assuming, w.l.o.g., that we relax all $m_1$ constraints of (3.7).

Essentially, this results in a problem for which container routing decisions must be made, (3.7), while adhering to train capacity constraints, (3.8). Even for a fixed set of multipliers, the relaxed problem is almost as difficult as the original problem because of the form of constraints (3.12), which couple all the variables.

### 3.2 Bender's Decomposition

#### 3.2.1 Procedure and Bender's Formulation

We also explored the possibility of implementing another classical decomposition technique, Bender's decomposition. Whereas with Lagrangian relaxation in which troublesome constraints are treated specially, this method mitigates the effect of troublesome (usually integer) variables (Nemhauser and Wolsey, 1988). In our implementation, we divide the problem into an integer program, called the master problem, and a linear program, or the subproblem. The rationale behind this decomposition is that the linear subproblem is easily solved once the values for the integer variables, derived from the smaller and more specialized master problem, are known.

Let us consider the following mixed integer problem:

$$(MIP) : \quad \min cx + fz$$

subject to

$$Ax = b \quad (3.14)$$
\[ Dx - ez \leq 0 \quad (3.15) \]

\[ x, z \geq 0 \text{ and } z \text{ integer} \quad (3.16) \]

The master problem is given as follows:

\[(MIP') : \quad \min \ \eta \]

subject to

\[ \eta \geq f z - e \omega q z \quad q = 1, \cdots, Q \quad (3.17) \]

\[ z \geq 0 \text{ and integer} \quad (3.18) \]

where \( \omega_q \) represents the dual multipliers on constraints (3.15) in (MIP) at iteration \( q \), and \( Q \) represents the current iteration number.

The subproblem has the following structure:

\[(SP') : \quad \min c x \]

subject to

\[ Ax = b \quad (3.19) \]

\[ Dx \leq ex \quad (3.20) \]

\[ x \geq 0 \quad (3.21) \]

Specifically, the structure of the two problems is as follows: The objective function of the master problem is a single scalar which is to be minimized subject to a lower bound set of constraints given by (3.17). This lower bound is defined by expressions involving the difference between the terms containing the integer variables in the original objective function and the product of the right hand side of the constraints with their corresponding dual multipliers in the subproblem. The subproblem has the same form as the original problem except that the troublesome (in our case, integer) variables are specified.

The Bender’s procedure iterates between the master problem and the subproblem. Based on the current values of the integer variables, new dual multipliers are generated from the subproblem at each iteration. These dual multipliers yield information as to how the values of the variables in the master problem can be changed to improve the solution to the
original problem. For example, a negative dual multiplier on a constraint in (3.20) in the
subproblem indicates that a decrease in the objective function value could be realized from
increasing the right hand side of that constraint. Passing the dual multiplier to the master
problem will increase the economic benefit of raising the value of \( z \) corresponding to the
right hand side in the subproblem. In a subsequent iteration, if this constraint in (3.20) has
slack, this indicates that increasing the right hand side will no longer improve the objective
function value, and a zero dual multiplier is then passed to the master problem. Each
cut, or lower bound constraint, generated at each iteration of solving the master problem,
is required to hold at all subsequent iterations. Hence, the number of constraints in the
master problem increases with the number of iterations. The procedure terminates when
the objective function values of the two problems are equal, or when some prespecified
tolerance is reached.

Applying this decomposition to our problem, we obtain the following master problem:

\[
(MP) : \quad \min \eta
\]

subject to

\[
\eta \geq \sum_{ikt} S^{p_k}_{ikt} \cdot z_{ikt} + \sum_{ijt} S^{p_j}_{ijt} \cdot z_{ijt} + \sum_{jkt} S^{h_j}_{jkt} \cdot z_{jkt} - C \sum_{iktq} \omega_{iktq} \cdot z_{ikt} \quad (3.22)
\]

\[
- C \sum_{ijtq} \omega_{ijtq} \cdot z_{ijt} - C \sum_{jktq} \omega_{jktq} \cdot z_{jkt} \quad q = 1, \ldots, Q \quad (3.23)
\]

\[
z \geq 0 \quad \text{and} \quad \text{integer} \quad (3.24)
\]

Here \( \omega_{iktq}, \omega_{ijtq} \) and \( \omega_{jktq} \) represent the dual multipliers obtained at iteration \( q \) from
constraints (3.37), (3.38), and (3.39) of the subproblem given below. We specify the sub-
problem without integrality restrictions on the variables. The intuition behind this result
is given in Section 4.4.2.

\[
\begin{align*}
\min & \quad \sum_{ikt} h \cdot I^{e}_{ikt} + \sum_{ijktl} h \cdot I^{h}_{ijktl} + \sum_{ijktl} \sum_{w=t+\beta_{ij}} h \cdot x_{ijktl} \\
& + \sum_{i} c_{ik} \cdot x_{ik} + \sum_{i} c_{ijk} \cdot x_{ijk} \\
& + \sum_{ikt} \sum_{l} l^{p}_{ikt} \cdot x_{ikt} + \sum_{ij} l^{p}_{ij} \cdot z_{ijt} + \sum_{ijk} l^{h}_{ijk} \cdot u_{ijk} + \sum_{ikl} l^{d}_{ikl} \cdot x_{ikl} + \sum_{ijk} l^{d}_{ijk} \cdot u_{ijk} \\
& + \sum_{ikt} S^{d}_{ikt} \cdot z_{ikt} + \sum_{ij} S^{p}_{ij} \cdot z_{ijt} + \sum_{jkt} S^{h}_{jkt} \cdot z_{jkt}
\end{align*}
\]
subject to

\[ I_{iktl}^0 = 0 \quad \forall i, k, t = 1, t + \alpha_{ik} \leq l < t + \beta_{ij} + \gamma_{jk} + \delta_j \quad (3.25) \]

\[ b_{iktl} = I_{iktl}^0 + z_{iktl} + \sum_j x_{ijkl} \quad \forall i, k, t = 1, l \geq t + \beta_{ij} + \gamma_{jk} + \delta_j \quad (3.26) \]

\[ b_{iktl} = -I_{ik(t-1)l}^0 + z_{iktl} \quad \forall i, k, t > 1, t + \alpha_{ik} \leq l < t + \beta_{ij} + \gamma_{jk} + \delta_j \quad (3.27) \]

\[ b_{iktl} = I_{iktl}^0 - I_{ik(t-1)l}^0 + z_{iktl} + \sum_j x_{ijkl} \quad \forall i, k, t > 1, l \geq t + \beta_{ij} + \gamma_{jk} + \delta_j \quad (3.28) \]

\[ x_{ijkl} = u_{ijk(t+\beta_{ij}+\delta_j)} \quad \forall i, j, k, t = 1, l = t + \gamma_{jk} \quad (3.29) \]

\[ I_{ijk(t+\beta_{ij}+\delta_j)}^h = x_{ijkl} - u_{ijk(t+\beta_{ij}+\delta_j)} \quad \forall i, j, k, t = 1, l > t + \gamma_{jk} \quad (3.30) \]

\[ -I_{ijk(t+\beta_{ij}+\delta_j-1)}^h = x_{ijkl} - u_{ijk(t+\beta_{ij}+\delta_j)} \quad \forall i, j, k, t > 1, l = t + \gamma_{jk} \quad (3.31) \]

\[ I_{ijk(t+\beta_{ij}+\delta_j)}^h = I_{ijk(t+\beta_{ij}+\delta_j-1)}^h + x_{ijkl} - u_{ijk(t+\beta_{ij}+\delta_j)} \quad \forall i, j, k, t > 1, l > t + \gamma_{jk} \quad (3.32) \]

\[ b_{ik(t-\alpha_{ik})} = x_{ik(t-\alpha_{ik})} \quad \forall i, k, t = 1 + \alpha_{ik}, t \leq l < t + \beta_{ij} + \gamma_{jk} + \delta_j - \alpha_{ik} \quad (3.33) \]

\[ I_{iktl}^d = x_{ik(t-\alpha_{ik})} + \sum_j u_{ijk(t-\gamma_{jk})} \quad \forall i, k, t = 1 + \alpha_{ik}, l \geq t + \beta_{ij} + \gamma_{jk} + \delta_j - \alpha_{ik} \quad (3.34) \]

\[ \sum_{w=1}^{t-\alpha_{ik}} b_{ikwl} = I_{ik(t-1)l}^d + x_{ik(t-\alpha_{ik})} \quad \forall i, k, t > 1 + \alpha_{ik}, t \leq l < t + \beta_{ij} + \gamma_{jk} + \delta_j - \alpha_{ik} \quad (3.35) \]

\[ I_{iktl}^d = I_{ik(t-1)l}^d + x_{ik(t-\alpha_{ik})l} + \sum_j u_{ijk(t-\gamma_{jk})l} \quad \forall i, k, t > 1 + \alpha_{ik}, l \geq t + \beta_{ij} + \gamma_{jk} + \delta_j - \alpha_{ik} \quad (3.36) \]

\[ \sum_l x_{ikt} \leq C \cdot z_{ikt} \quad \forall i, k, t \quad (3.37) \]

\[ \sum_{kl} x_{ijkl} \leq C \cdot z_{ijt} \quad \forall i, j, t \quad (3.38) \]

\[ \sum_{il} u_{ijkl} \leq C \cdot z_{jkt} \quad \forall j, k, t \quad (3.39) \]

\[ I_{iktl}^0, I_{iktl}^d, x_{iktl} \geq 0 \quad \forall i, k, t, l \quad (3.40) \]

\[ I_{ijktl}^h, x_{ijkl}, u_{ijkl} \geq 0 \quad \forall i, j, k, t, l \quad (3.41) \]

Values of the train variables \( z_{ikt}, z_{ijt}, \) and \( z_{jkt} \) (i.e., a train schedule) are obtained from the master problem. Initially, these values can be set equal to zero. Alternatively, an initial feasible train schedule can be constructed. Based on these values, the subproblem generates
dual multipliers for constraints (3.37) through (3.39). Let us consider a train traveling between two points, either between an origin and a hub, an origin and a destination, or between a hub and a destination at a given time period. For ease of exposition, we will refer to this as a train operating within a particular $ikt$ slice, and it is understood that this also encompasses indirect train travel either to or from a hub.

If there is slack in the constraint, i.e., if a train corresponding to a particular $ikt$ slice is not fully utilized, the subproblem will generate a dual multiplier value of 0 indicating to the master problem that the value for the corresponding train variable does not need to be increased, in other words, no additional capacity needs to be added. Conversely, if a train corresponding to a particular $ikt$ slice is full, it may be beneficial to add another train. Mathematically, this corresponds to a tight capacity constraint for this particular $ikt$ slice. This constraint will have a dual multiplier whose value corresponds to the improvement in the objective function value per unit increase in the right hand (constant) side. Hence, adding another train in the $ikt$ slices corresponding to the most negative dual multipliers will result in the maximum improvement to our subproblem whose objective is to minimize variable container costs. The procedure continues to iterate between the master problem and the subproblem until trains have been added to optimally accommodate all containers.

3.2.2 Implementation and Subproblem Feasibility

Given an initial feasible train schedule, an optimal solution for the container shipments can be determined from the subproblem. Dual multipliers from the capacity constraints in the subproblem are passed to the master problem. Because it is seldom the case that the trains are exactly full (i.e., they generally have unused capacity), the values for these dual multipliers are zero. This results in a train schedule from the master problem in which no trains are sent, i.e., the master problem "overshoots" and creates a train schedule which is too conservative. As a consequence, the subproblem is infeasible at the next iteration and the process terminates. In the following sections we explain why the structure of the master and sub-problems does not lend itself to a straightforward Bender's decomposition procedure. We also describe methods to enhance the Bender's procedure by reformulating the problem.
<table>
<thead>
<tr>
<th>arrival time</th>
<th>level of service</th>
<th>Oak → Chi</th>
<th>Oak → Memp</th>
<th>LA → Chi</th>
<th>LA → Memp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>90</td>
<td>15</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>110</td>
<td>180</td>
<td>90</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>40</td>
<td>25</td>
<td>80</td>
<td>90</td>
</tr>
</tbody>
</table>

Table 3.1: Example Problem Data, II

3.2.3 Adding Constraints to the Master Problem

Our first attempt at correcting the premature termination of the algorithm resulting from an infeasible solution for the subproblem is to place lower bounds in the master problem on the number of trains that must be sent to satisfy level of service constraints. Specifically, for each time period, we require that the cumulative number of direct and indirect trains be sufficient to ensure on-time delivery for the cumulative demand. We illustrate this idea with an example. Let us assume a train has a capacity of 200 containers and consider the following arrival data given in Table 3.1.

We assume that direct travel time between any origin and any destination is four days, and indirect travel time is five days, which includes two days' travel time to the hub, two days' travel time from the hub, and one day of delay at the hub. Hence, containers requiring a level of service of 7 (i.e., are due at their destination by time 7) can be serviced by an indirect train leaving as late as time period 2 or by a direct train leaving as late as time period 3. We list below the constraints imposed outbound from the origin for time period 3, and the corresponding constraints outbound from the hub. The first set of constraints is imposed in time period 3 for containers requiring a level of service of 7:

\[
\begin{align*}
    z_{oc1} + z_{oc2} + z_{oc3} + z_{oh1} + z_{oh2} & \geq \left[ \frac{90+5}{200} \right] \\
    z_{om1} + z_{om2} + z_{om3} + z_{oh1} + z_{oh2} & \geq \left[ \frac{15+10}{200} \right] \\
    z_{lc1} + z_{lc2} + z_{lc3} + z_{lh1} + z_{lh2} & \geq \left[ \frac{20+5}{200} \right] \\
    z_{lm1} + z_{lm2} + z_{lm3} + z_{lh1} + z_{lh2} & \geq \left[ \frac{10+10}{200} \right] \\
    z_{oc1} + z_{oc2} + z_{oc3} + z_{om1} + z_{om2} + z_{om3} + z_{oh1} + z_{oh2} & \geq \left[ \frac{90+5+15+10}{200} \right] \\
    z_{lc1} + z_{lc2} + z_{lc3} + z_{lm1} + z_{lm2} + z_{lm3} + z_{lh1} + z_{lh2} & \geq \left[ \frac{20+5+10+10}{200} \right] \\
    z_{hc5} & \geq \left[ \frac{90+5+10+5}{200} \right] - z_{oc1} - z_{oc2} - z_{oc3} - z_{lc1} - z_{lc2} - z_{lc3} - z_{hc4} \\
    z_{hm5} & \geq \left[ \frac{15+10+10+10}{200} \right] - z_{om1} - z_{om2} - z_{om3} - z_{lm1} - z_{lm2} - z_{lm3} - z_{hm4}
\end{align*}
\]
For containers requiring a level of service of 8, the following constraints are imposed in time period 4:

\[
\begin{align*}
\sum_{w} z_{oc1} + z_{oc2} + z_{oc3} + z_{oc4} + z_{oh1} + z_{oh2} + z_{oh3} \geq \left[ \frac{90+5+110+40}{200} \right] \\
\vdots \\
\sum_{w} z_{oc1} + z_{oc2} + z_{oc3} + z_{oc4} + z_{om1} + z_{om2} + z_{om3} + z_{om4} + z_{oh1} + z_{oh2} + z_{oh3} \geq \left[ \frac{90+5+110+40+15+10+180+25}{200} \right] \\
\vdots \\
\sum_{w} z_{hc6} \geq \left[ \frac{90+5+110+40+20+5+90+80}{200} \right] - z_{oc1} - z_{oc2} - z_{oc3} - z_{oc4} - z_{lc1} - z_{lc2} - z_{lc3} - z_{lc4} - z_{hc4} - z_{hc5}
\end{align*}
\]

where the indices \(o, l, c, m,\) and \(h\) represent the origins Oakland and LA, the destinations Chicago and Memphis, and the hub, respectively, and the third index represents the time period in which the train is sent out.

In general, using the previously defined notation, we can express these three sets of constraints as follows:

\[
\begin{align*}
\sum_{w=1}^{t} z_{ikw} + \sum_{j} \sum_{w=1}^{t'} z_{ijw} \geq \left[ \frac{\sum_{w=1}^{t} \sum_{p=1}^{l} b_{ikwp}}{C} \right] \\
\forall i, k, l, 1 \leq t \leq l - \alpha_{ik}, 1 \leq t' \leq l - \beta_{ij} - \gamma_{jk} - \delta_j
\end{align*}
\]

This first set of constraints ensures that the cumulative number of direct and indirect trains scheduled for each origin and destination must be sufficient in each time period to meet on-time delivery requirements for the cumulative number of containers that have arrived, for all levels of service.

\[
\begin{align*}
\sum_{k} \sum_{w=1}^{t} z_{ikw} + \sum_{j} \sum_{w=1}^{t'} z_{ijw} \geq \left[ \frac{\sum_{k} \sum_{w=1}^{t} \sum_{p=1}^{l} b_{ikwp}}{C} \right] \\
\forall i, l, 1 \leq t \leq l - \alpha_{ik}, 1 \leq t' \leq l - \beta_{ij} - \gamma_{jk} - \delta_j
\end{align*}
\]

This second set of constraints ensures that the cumulative number of direct and indirect trains scheduled for each origin bound for all destinations must be sufficient in each time...
period to meet on-time delivery requirements for the cumulative number of containers that have arrived, for all levels of service.

\[
z_{jk}(t+\beta_j+\delta_j) \geq \left[ \sum_{w=1}^{t} \sum_{l=1+\alpha_{ik}}^{t+\beta_j+\delta_j-1} \frac{\hat{b}_{ikw}}{C} \right] - \sum_{w=1}^{t} \sum_{i=1}^{t} \sum_{w=1}^{t+\beta_j+\delta_j-1} z_{ikw} - \sum_{w=1+\beta_j+\delta_j}^{t} z_{jkw} \forall j, k, l, 1 \leq t \leq l - \alpha_{ik}
\]

This final set of constraints ensures that for each hub, destination, time period and level of service, the cumulative number of indirect trains must be able to accommodate the cumulative number of containers that have arrived at the hub, and have not yet been shipped out.

When we impose these constraints on the master problem, we find that its structure is such that the smallest number of trains possible (i.e., the lower bound implied by the constraints above) are sent. In fact, the larger in magnitude (more negative) the dual multipliers are, the more unattractive it is for the master problem to generate train variables with positive values because of the sign of the dual multipliers and the structure of the objective function. Hence, the lower bound constraints serve only to concentrate or “clump” train departures at time periods in which the dual multipliers are small in magnitude compared with the multipliers for the other time periods.

Another problem with this approach is the structure of the constraints, which may cause trains to be sent out before some containers traveling on these trains even arrive! Because we only consider the ceiling of the cumulative arrivals, if few containers arrive in a given period (say, period 3), and trains are sent in a previous period (period 2) with slack, the constraints for period 3 would be satisfied without adding trains, and the period 3 arrivals would remain unserviced. This result would be obtained if the dual multipliers were larger in period 3 than in period 2.

To correct for this, we could impose constraints in the master problem to serve as an upper bound on the cumulative number of trains that can be sent out by any given time period based on the cumulative number of container arrivals by that time period. However, tight bounds are hard to obtain because we cannot say definitively how full direct and indirect trains should be in an optimal solution.
Another way in which we tried to correct for the infeasibility of the subproblem is to allow containers to be shipped late or not at all while incurring a modest to large penalty. In this way, we would not require an extra train to be sent to service only a few containers in order to maintain feasibility of the subproblem.

This modified network portion of our problem, i.e., the portion of the problem addressing only the container routing decisions, is similar to the depiction in Figure 2.1. We add uncapacitated backorder arcs to allow for delays. We impose penalties of three levels of severity (i.e., for the arrival of a container one day late, two days late, or if a container is left unshipped). The penalty need not increase linearly with the tardiness of the arrival. More general penalty structures could also be imposed. The formulation for this subproblem is given below.

\[
\begin{align*}
\min & \quad \sum_{i, k, t} h \cdot I_{i, k, t}^0 + \sum_{i, j, k, t} h \cdot I_{i, j, k, t}^h + \sum_{i, j, k, t} c_{i, k} \cdot x_{i, j, k, t} + \sum_{i, j, k, t} c_{i, j, k} \cdot x_{i, j, k, t} + \\
& \quad \sum_{i, j, k, t} l_i^0 \cdot x_{i, k, t} + \sum_{i, j, k, t} l_i^1 \cdot x_{i, j, k, t} + \sum_{i, j, k, t} l_i^2 \cdot x_{i, j, k} + \sum_{i, j, k, t} l_i^3 \cdot x_{i, j, k} + \\
& \quad \sum_{i, j, k, t} P_1 \cdot I_{i, k, t}^d + \sum_{i, j, k, t} P_2 \cdot I_{i, k, t}^d + \sum_{i, j, k, t} P_3 \cdot I_{i, k, t}^d
\end{align*}
\]

subject to

\[
\begin{align*}
I_{i, k, t}^0 &= 0 \quad \forall i, k, t = 1, t + \alpha_{ik} \leq l < t + \beta_{ij} + \gamma_{jk} + \delta_j \\
b_{i, k, t} &= I_{i, k, t}^0 + x_{i, k, t} + \sum_j x_{i, j, k, t} \quad \forall i, k, t = 1, l \geq t + \beta_{ij} + \gamma_{jk} + \delta_j \\
b_{i, k, t} &= -I_{i, k, t}^0 + x_{i, k, t} \quad \forall i, k, t > 1, t + \alpha_{ik} \leq l < t + \beta_{ij} + \gamma_{jk} + \delta_j \\
b_{i, k, t} &= I_{i, k, t}^d - I_{i, k, t}^0 + x_{i, k, t} + \sum_j x_{i, j, k, t} \quad \forall i, k, t > 1, l \geq t + \beta_{ij} + \gamma_{jk} + \delta_j \\
x_{i, j, k, t} &= u_{i, j, k, t} + \alpha_{ik} \quad \forall i, j, k, t = 1, l = t + \gamma_{jk} \\
I_{i, j, k, t}^h &= x_{i, j, k, t} - u_{i, j, k, t} \quad \forall i, j, k, t = 1, l > t + \gamma_{jk} \\
-I_{i, j, k, t}^h &= x_{i, j, k, t} - u_{i, j, k, t} \quad \forall i, j, k, t > 1, l = t + \gamma_{jk} \\
I_{i, j, k, t}^h &= I_{i, j, k, t}^h + x_{i, j, k, t} - u_{i, j, k, t} \quad \forall i, j, k, t > 1, l > t + \gamma_{jk} \\
I_{i, k, t}^d &= x_{i, k, t} + \sum_j u_{i, j, k, t} \quad \forall i, k, t = 1 + \alpha_{ik}, l \\
I_{i, k, t}^d &= I_{i, k, t}^d + x_{i, k, t} + \sum_j u_{i, j, k, t} \quad \forall i, k, 1 + \alpha_{ik} < t \leq T + 1, l
\end{align*}
\]
(2.17), (2.18), (2.19)

All variables are restricted to be nonnegative and integer.

where

\[ P_1 = \text{the penalty incurred for a container arriving at the destination one day late} \]
\[ P_2 = \text{the penalty incurred for a container arriving at the destination two days late} \]
\[ P_e = \text{the penalty incurred for an unshipped container} \]

Although this formulation maintains feasibility of the subproblem, the same implementation difficulties (e.g., "overshooting" and "clumping") remain.

3.2.4 Sign of the Dual Multipliers

The master problem and subproblem given in Section 3.2.1 can be stated as follows:

\[ (\text{MIP}^{''}) : \quad \min \eta \]
subject to

\[ \eta \geq \sum (S - Cw_q) \cdot z \quad q = 1 \cdots Q \quad (3.52) \]
\[ z \geq 0 \text{ and integer} \quad (3.53) \]

\[ (\text{SP}') : \quad \min c \cdot x \]
subject to

\[ Ax = b \quad (3.54) \]
\[ Dx \leq ez \quad (3.55) \]
\[ x \geq 0 \quad (3.56) \]

Dual multipliers on less than or equal to constraints in a minimization problem are nonpositive. Negative dual multipliers from the capacity constraints in subproblem indicate to the master problem that the number of trains in the schedule should be increased.
However, given a negative dual multiplier, the master problem sets the corresponding train variable equal to zero. Therefore, we consider two modifications to change the sign on the dual multipliers or the form of the lower bound in the master problem to coordinate the objectives of the master problem and the subproblem.

A tight capacity constraint for a particular $ikt$ slice indicates that another train should be added. This constraint will, in general, have a nonzero dual multiplier, with a value corresponding to the improvement in the objective function per unit increase in the right hand (constant) side. Hence, adding another train in the $ikt$ slices corresponding to the most negative dual multipliers will result in the maximum improvement to our subproblem whose objective is to minimize variable container costs.

However, because the Bender's master problem is a minimization problem in which we successively add lower bound constraints on the objective function value, the optimal solution will be such that this lower bound is as small as possible. Hence, the terms with positive coefficients on the train variables will be set to zero in the optimal solution, and the terms with negative coefficients will have corresponding optimal train values that are as large as possible. Unfortunately, this is inconsistent with the way in which the master and sub-problems should interact. The subproblem has large negative dual multipliers in instances where adding another train at a specific $ikt$ slice will effect a large decrease in the objective function. However, it is specifically in these instances that adding the train is least attractive for the master problem, since the coefficient on the train variable for that $ikt$ slice is a large positive value.

Let us consider a reformulation of the subproblem such that the sign of the dual multipliers is changed through a change in the objective function in the subproblem, specifically, one in which we maximize profit rather than minimize costs:

\[
\max \sum_{ikt} -h \cdot I^2_{ikt} + \sum_{ijkt} -h \cdot I^1_{ijkt} + \sum_{ijkl} \sum_{w=t+\beta_j}^{t+\beta_j+\delta_j} -h \cdot x_{ijkwl} + \\
\sum_{ikt} r_{ikt} \cdot x_{ikt} + \sum_{ijkt} r_{ijkt} \cdot x_{ijkt} + \sum_{ijkl} r_{ijkl} \cdot u_{ijkl}
\]
subject to

(2.1), (2.2), (2.3), (2.4)

(2.26), (2.27), (2.28), (2.29) \hspace{1cm} \text{(network)}

(2.30), (2.31), (2.32), (2.33)

(2.17), (2.18), (2.19)

\[ I^a_{iklt}, x_{iklt} \geq 0 \quad \forall i, k, t, l \quad \text{(3.57)} \]

\[ I^b_{ijkl}, z_{ijkl}, u_{ijkl} \geq 0 \quad \forall i, j, k, t, l \quad \text{(3.58)} \]

where

\[ r_{ikl} = \text{revenue derived from transporting a container directly from origin } i \text{ to destination } k \text{ at a level of service } l \]

\[ r_{ijkl} = \text{revenue derived from transporting a container from origin } i \text{ via hub } j \text{ to destination } k \text{ at a level of service } l \]

where the revenue reflects the variable profit earned from sending the containers less the transportation and handling costs incurred from these movements.

This is consistent with the Bender's framework because tight capacity constraints necessitate additional capacity. An increase in the capacity results in an increase in the profit represented by the objective function. Moreover, the optimal solution for the master problem now generates a schedule with trains for those \( ikt \) slices for which the dual multipliers are sufficiently large (i.e., there is sufficient profit to be gained from the addition of a train).

This change is equivalent to changing the structure of the lower bound constraint in the master problem as follows:

\[ (MIP^m): \quad \min \eta \]

subject to

\[ \eta \geq \sum (S + C_{q}) * z \quad q = 1, \ldots, Q \quad \text{(3.59)} \]

\[ z \geq 0 \text{ and integer} \quad \text{(3.60)} \]

52
In other words, our subproblem can still be viewed as a cost minimization problem if a sign change is made to the lower bound constraint in the master problem.

Although these two methods correct for the logical deficiency, neither was implemented because in the absence of additional constraints in the master problem, they result in an unbounded solution. Because the coefficients on the train variables are now negative for sufficiently large dual multipliers, allowing an unlimited number of trains to be scheduled results in an infinite lower bound on the function we wish to minimize. To rectify this, an upper bound can be imposed on the number of trains sent for any given \( ikt \) slice. This is not an effective correction because a negative coefficient will induce the total number of trains for the corresponding \( ikt \) slice to be set at this upper limit (to drive the lower bound on the objective function as low as possible).

In order to dampen this oscillatory behavior in assigning trains within each \( ikt \) slice, we can create binary variables with an index \( v \) to represent different trains traveling at the same time between the same locations:

\[
z_{ikt} = \begin{cases} 
1 & \text{if train } v \text{ is sent directly from origin } i \text{ to destination } k \text{ at time } t \\
0 & \text{otherwise}
\end{cases}
\]

\[
z_{ijt} = \begin{cases} 
1 & \text{if train } v \text{ is sent from origin } i \text{ to hub } j \text{ at time } t \\
0 & \text{otherwise}
\end{cases}
\]

\[
z_{jkt} = \begin{cases} 
1 & \text{if train } v \text{ is sent from hub } j \text{ to destination } k \text{ at time } t \\
0 & \text{otherwise}
\end{cases}
\]

where

\[
\sum_{v=1}^{m} z_{ikt} = z_{ikt} \quad \forall i, k, t
\]

\[
\sum_{v=1}^{m} z_{ijt} = z_{ijt} \quad \forall i, j, t
\]

\[
\sum_{v=1}^{m} z_{jkt} = z_{jkt} \quad \forall j, k, t
\]

assuming, w.l.o.g., an upper bound of \( m \) trains for each \( ikt \) slice.

We could assess different costs to these variables in the master problem. Such a modification may also have practical relevance because sending multiple trains to service demand
on the same day may be proportionately more costly. Note that we have indirectly placed bounds on the total number of trains in an \( ikt \) slice by imposing an upper bound on the train index \( v \). This will constrain the objective function value of the master problem, but will not dampen the oscillatory behavior. If one train is sent in a given \( ikt \) slice, then there is a tendency to send all of these trains given similar fixed train costs.

### 3.2.5 Other Observations

The master problem should provide us with information both as to the time periods in which trains should be sent and whether these trains are direct or indirect. However, in the formulations we have examined so far, these decisions are not considered in tandem. The dual multipliers are taken from constraints in which both successive time periods and direct and indirect trains are considered separately. In essence, our formulation is too decoupled. In addition to the original capacity constraints, we could relate direct and indirect shipments and the corresponding trains through the following constraints:

\[
\sum_{kl} x_{ijkl} + \sum_l x_{ikl} \leq C \ast z_{ikt} + C \ast z_{ijt} \quad \forall i, j, k, t
\]

A similar constraint would hold for shipments outbound from the hub. By including constraints such as these, the feasibility of certain shipments to be routed both directly and indirectly would be accounted for.

Because the decisions in one period influence the policy in the next, dual multipliers from constraints in which time periods are considered independently do not reflect interactions between time periods. For example, if the due date of a container does not require its immediate departure from the origin, it can be held in inventory for one or more periods. Sending a container in one eligible time period affects whether or not it is sent in a subsequent period. Therefore, another related change would be to derive dual information from constraints in which time periods are considered interdependently.

In summary, we have found that the master problem converges to a suboptimal solution because the lower bound constraint is required to be satisfied for dual multipliers from each iteration. Hence, when certain unfavorable dual multipliers are generated, there is no incentive for the solution to change since the bad set of dual multipliers maintain a high lower bound. Imposing upper bounds on the number of trains scheduled does not alleviate
this problem. Even if the lower bound constraint on the objective function of the master problem is required to hold only for dual multipliers at the current iteration number, a "bang-bang" effect results. That is, not enough trains are sent out, which causes the dual multipliers to be very large. Consequently, the master problem generates a schedule with an excess of trains, which, in turn, causes the subproblem to generate dual multipliers with value zero. This results in the master problem sending no trains at the next iteration.

Other variations in implementation to try to eliminate this effect included changing the initial train schedule, changing the relative magnitudes of the costs (or revenue) in the subproblem, and perturbing the dual multipliers. Because of the uncompromising nature between the way in which the master problem and the subproblem interact, none of these modifications changes the fundamental behavior of the problem.

Even with the final modification described above, we confront the fundamental difficulty that the dual multipliers represent the change in the objective function value per unit change in the right hand side. However, when the values of the train variables are changed by one unit, the change is reflected in a right hand side change equal to the capacity of the train. Hence, the dual information is not consistent with the ability of the master problem to adjust the values of the integer variables accordingly.

We conclude that our problem does not lend itself well to the use of Bender's decomposition via our implementation techniques. We would like to emphasize, however, that there may be an implementation of this methodology that will yield good results. For example, there may be cuts we could add to the master problem, or penalties we could add to dampen the oscillatory behavior. Alternatively, we might reformulate our problem by defining our decision variables differently, e.g., increasing the number of indices on the variables representing container movements in order to include the train on which the container is sent. However, because of the difficulties we experienced in applying this approach, we decided to develop our own original decomposition procedure described in the subsequent chapters to produce good solutions for our problem.
Chapter 4

New Decomposition Technique

4.1 Description of the Approach

As was shown in Chapter 3, our formulation does not lend itself well to two classical decomposition procedures, a constraint-based procedure and a variable-based procedure. Therefore, we introduce an unconventional method of decomposition based on a physical, rather than a mathematical, decomposition of the problem.

Our decomposition approach is motivated, in part, by the observation that if the optimal pattern of container arrivals at the hub (by origin, destination, arrival date at origin, and due date) were known, we could infer which containers were to be sent on direct trains. The remainder of the problem would decompose into two parts: (i) Scheduling trains and containers to produce the optimal shipment of containers either directly bound for their destinations or bound for the hub, and (ii) optimally scheduling trains and containers outbound from the hub. These two subproblems are easier to solve than the entire problem, as we explain in more detail later. The difficulty lies in finding the optimal, or sometimes even a good, pattern of container arrivals at the hub. Not only are there many feasible patterns, but the optimum objective of each of the aforementioned subproblems is not a smooth, well-behaved function of the arrival patterns, because of the fixed-charge component of the objective function. In fact, the arrival patterns themselves are four-dimensional matrices, where the dimensions correspond to the indices on the parameter denoting container arrivals at the origin. Our hope is that there are sufficiently many good arrival patterns that will produce near-optimal solutions.

We consider a heuristic decomposition procedure in which we separate the graph into two
distinct collections of nodes and the associated decisions: (i) The set of nodes corresponding to the origins and the arcs emanating from them, and (ii) the set of nodes corresponding to the hub and the arcs emanating from it. We then solve these two problems in succession. We first consider the containers by arrival date, due date, origin, and destination. We formulate and solve the problem of how to schedule the trains and containers outbound from the origins, both directly to the destinations, and indirectly to the hub. Given the decisions made at the origins, the problem then becomes one of determining a train schedule and container shipment scheme outbound from the hub. We call these two problems the "origin scheduling" and "hub scheduling" problem, respectively. Figure 4.1 shows how we partition the network.

Without modification, the origin scheduling problem ignores certain effects of its solution on the costs incurred in the hub scheduling problem. Some of these effects can be incorporated into the origin scheduling problems, resulting in solutions superior to those from a naive decomposition.

Let us consider the consequences of making train scheduling decisions at the origin without considering downstream costs. A direct train sent from the origin incurs a higher fixed cost than an indirect train from the origin to the hub because the fixed cost for the
indirect train is only assessed for the distance between the origin and the hub, rather than for the complete distance from the origin to the destination. Similarly, the handling and holding costs for indirect containers in the origin scheduling problem are accounted for at the origin, but not at the hub. Because of the shortsightedness of this approach (i.e., costs incurred at the hub are not taken into account at the origin), it is necessary to provide the origin scheduling problem with more information as to the true costs incurred as a result of decisions made at the origin. These solutions for origin scheduling problem will yield better, more informed, solutions for the original problem.

Let us define the following additional cost parameters, $\overline{S_{ij}}$ and $\overline{H_{ij}}$, representing estimates of the sum of the setup costs at the origin and those at the hub, and the sum of the handling costs at the origin and the hub, respectively. In estimating the setup cost at the hub, we assume that each indirect train bound for the hub is associated with an unspecified indirect train sent from the hub. Under this assumption, the revised costs represent the entire setup and handling costs incurred on the intermodal journey until the container reaches its destination. Note that the assumption implies a one-to-one ratio between the number of trains arriving at a hub, and the number of trains departing from that hub over the time horizon. The resulting cost assessment may either overestimate or underestimate the true costs of the indirect train schedule. If fewer indirect trains are sent from the origin to the hub than from the hub to the destination, the indirect setup costs are underestimated while the opposite is true if more indirect trains are sent from the origin to the hub than from the hub to the destination. It would be extremely difficult to more accurately predict the balance of indirect train traffic into and out of the hub. From an operational standpoint, it can be argued that the number of indirect trains sent into and out of the hub should be approximately equal. If the setup cost is high enough to encourage “batching” at the origin with relatively infrequent trains, it is more likely to induce a one-to-one ratio between trains inbound to and outbound from the hub due to container arrival patterns there. However, instances in which the traffic levels are extremely irregular could result in an imbalance of trains. For example, if traffic bound for a certain destination in a given time period is particularly heavy, greater consolidation of shipments at the hub might result in fewer trains being sent outbound from the hub than from the origin.

Let $S^h_j$ be the setup cost at hub $j$, obtained by using an average or a weighted average
of $S^{h}_{jk}$ across destinations. The new cost parameters can be expressed as follows:

$$\tilde{S}_{ij} = S_{ij} + S^{h}_{j}$$

$$\tilde{I}_{ij} = l_{i} + l^{h}_{j}$$

Then, using these additional parameters in conjunction with the parameters and variables defined earlier, we formulate the origin scheduling and hub scheduling problems. The origin scheduling problem is formulated as follows:

$$(P_o): \min \quad \tilde{Z}_1 =$$

$$\sum_{i,k,t} h \cdot I_{ikt} + \sum_{i,k,t} c_{ik} \cdot x_{iktl} + \sum_{i,j,k,t} c_{ijk} \cdot x_{ijkl} +$$

$$\sum_{i,k,t} l_{i} \cdot x_{iktl} + \sum_{i,j,k,t} l^{h}_{ij} \cdot x_{ijkl} + \sum_{i,k,t} S^{0}_{ik} \cdot z_{ikt} + \sum_{i,j,t} \tilde{S}_{ij} \cdot z_{ijt}$$

subject to

$$b_{iktl} = x_{iktl} \quad \forall i, k, t = 1, t + \alpha_{ik} \leq l < t + \beta_{ij} + \gamma_{jk} + \delta_j \quad (4.1)$$

$$b_{iktl} = I_{ikt} + x_{iktl} + \sum_{j} x_{ijkt} \quad \forall i, k, t = 1, l \geq t + \beta_{ij} + \gamma_{jk} + \delta_j \quad (4.2)$$

$$b_{iktl} = -I_{ik(t-1)l} + x_{iktl} \quad \forall i, k, t > 1, t + \alpha_{ik} \leq l < t + \beta_{ij} + \gamma_{jk} + \delta_j \quad (4.3)$$

$$b_{iktl} = I_{iktl} - I_{ik(t-1)l} + x_{iktl} + \sum_{j} x_{ijkt} \quad \forall i, k, t > 1, l \geq t + \beta_{ij} + \gamma_{jk} + \delta_j \quad (4.4)$$

$$\sum_{l} x_{iktl} \leq C \cdot z_{ikt} \quad \forall i, k, t \quad (4.5)$$

$$\sum_{kl} x_{ijkl} \leq C \cdot z_{ijt} \quad \forall i, j, t \quad (4.6)$$

$$I_{ikt}, x_{ikt} \geq 0 \text{ and integer} \quad \forall i, k, t, l \quad (4.7)$$

$$x_{ijkl} \geq 0 \text{ and integer} \quad \forall i, j, k, t, l \quad (4.8)$$

$$z_{ikt} \geq 0 \text{ and integer} \quad \forall i, k, t \quad (4.9)$$

$$z_{ijt} \geq 0 \text{ and integer} \quad \forall i, j, t \quad (4.10)$$
We formulate the hub scheduling problem as follows:

\[(P_h): \min \quad Z_2 = \]
\[
\sum_{jkt} h * I^h_{jkt} + \sum_{ijkl} I^h_{ijkl} * u_{ijkl} + \sum_{ijkl} c_{ijkl} * u_{ijkl} + \sum_{jkt} S^h_{jk} * z_{jkt}
\]

subject to

\[I^h_{jk(t+\beta_{ij}+\delta_j)t} = 0 \quad \forall j, k, t = 1, l = t + \gamma_{jk}\]

\[-I^h_{jk(t+\beta_{ij}+\delta_j-1)t} = \sum_{i} x_{ijkl} - \sum_{i} u_{ijkl(t+\beta_{ij}+\delta_j)t} \quad \forall j, k, t > 1, l = t + \gamma_{jk}\]

\[I^h_{jk(t+\beta_{ij}+\delta_j)t} = I^h_{jk(t+\beta_{ij}+\delta_j-1)t} + \sum_{i} x_{ijkl} - \sum_{i} u_{ijkl(t+\beta_{ij}+\delta_j)t} \quad \forall j, k, t > 1, l > t + \gamma_{jk}\]

\[x_{ijkl} = u_{ijkl(t+\beta_{ij}+\delta_j)t} \quad \forall i, j, k, t = 1, l = t + \gamma_{ik}\]

\[u_{ijkl(t+\beta_{ij}+\delta_j)t} \leq x_{ijkl} \quad \forall i, j, k, t = 1, l > t + \gamma_{ik}\]

\[\sum_{w=1}^{t} x_{ijklw} = \sum_{w=1}^{t} u_{ijkl(w+\beta_{ij}+\delta_j)t} \quad \forall i, j, k, t > 1, l = t + \gamma_{ik}\]

\[\sum_{w=1}^{t} u_{ijkl(w+\beta_{ij}+\delta_j)t} \leq \sum_{w=1}^{t} x_{ijklw} \quad \forall i, j, k, t > 1, l > t + \gamma_{ik}\]

\[\sum_{i} x_{ijkl} \leq C * Z_{jkt} \quad \forall j, k, t\]

\[x_{ijkl}, u_{ijkl} \geq 0 \text{ and integer} \quad \forall i, j, k, t, l\]

\[I^h_{ijkl} \geq 0 \text{ and integer} \quad \forall j, k, t, l\]

\[z_{jkt} \geq 0 \text{ and integer} \quad \forall j, k, t\]

Both the origin scheduling and hub scheduling problems have fewer variables and constraints than the original problem. Hence, we expect these two subproblems to be easier to solve. In fact, the problem size is reduced even further when we take advantage of our ability to decompose the origin scheduling problem. Because the origin scheduling problem determines only train schedules and the corresponding container movements outbound from each origin without considering any interaction between containers from different origins.
bound for the same destination, we can decouple the origin scheduling problem by origin. This observation is extremely important for two reasons. First of all, we have found that even small origin scheduling problems with three or four origins cannot be solved to optimality when decisions for all of the origins must be considered simultaneously. Secondly, decoupling the origin scheduling problems by origin causes the solvability of this subproblem to be completely independent of the number of origins in the original problem. Therefore, a problem with more origins would simply consist of solving more, but not necessarily larger, subproblems. Additionally, the hub scheduling problem decomposes by destination, because we can make train scheduling and container routing decisions outbound from the hub to a specific destination independently of any other destination. This also results in fewer decision variables per hub scheduling subproblem. However, because the problem structure is less complicated than for the origin scheduling problem (i.e., only an indirect train schedule, rather than both a direct and an indirect train schedule needs to be determined), we have been able to solve our hub scheduling problem instances for all destinations simultaneously. Empirically, our undecoupled hub scheduling problem instances take a matter of minutes for CPLEX to solve. The nature of the decisions for the hub scheduling problem is closely related to that of a discrete time production and inventory problem with multiple setup costs, where there is an upper limit on production for each setup. Our problem differs from those in the literature in that we have an added constraint on “production” in each period corresponding to the availability of containers that can be sent out. Yano and Newman (1998) present a theoretical analysis, including a polynomial time algorithm for solving the problem.

Another important observation is that given a fixed train schedule, both the origin scheduling and the hub scheduling problem are minimum cost network flow problems. Because of this, we can relax the integrality requirements on the variables representing the container routing decisions and ensure the existence of a solution with integer container shipments. Because one of these feasible solutions will yield the optimal solution, there exists an optimal solution with integral container shipments without imposing the integrality restriction on these variables. The reduction in the number of integer variables results in a substantial reduction in the computational effort needed to solve each of these subproblems to optimality.
Figure 4.2: Network Depiction of the Origin Scheduling Problem

We now describe representations of the two subproblems using single commodity networks. Because the origin scheduling problem can be decoupled by origin, we consider the subproblem for a fixed origin. Figure 4.2 depicts the network of an origin scheduling problem for a single origin given a fixed train schedule. The example has one hub, two destinations, two time periods, and two due dates. Note that travel time between all locations is instantaneous.

In addition to the supersource and supersink, there are two types of nodes: (i) a set of nodes, one for each (time period, commodity) pair, and (ii) a set of nodes, one for each time period in which trains travel between the same two locations. Each feasible path in the network passes through one or more nodes in the first set, and exactly one node in the second set. If the shipment is never held in inventory at the origin, it will pass through exactly one node in each set.

Arcs from the supersource to the first set of nodes have upper and lower capacity limits equal to the number of container arrivals of the commodity at the designated time period. Arcs from nodes in the first set to nodes in the second set represent container shipments on a given train, either directly to the destination, or into the hub. From each node in the second set, containers flow into a sink along an arc with a lower bound of zero and an upper
bound equal to the capacity of the train(s) scheduled between the specific origin-destination or origin-hub pair at the relevant time period. Direct and indirect transportation costs are assessed on the arcs between the origin and the corresponding train node. No costs are assessed on the arcs originating at the supersource or terminating at the supersink.

In addition to arcs that represent the movement of containers from one location to another, inventory arcs at the origin link each commodity type with the corresponding type in the following time period. Container flows on these arcs represent storage of containers at a location from one time period to the next. Inventory holding costs are assessed along these arcs.

Figure 4.3 depicts the network for the hub scheduling problem given a fixed train schedule. The hub scheduling subproblems can be decoupled by destination and require information as to a container's arrival time at the hub and its due date, but not its origin. The figure illustrates the network for a single hub-destination pair, two time periods, and two due dates. Note again that the travel time is instantaneous.

The hub scheduling problem has a source node, a supersink node and two additional types of nodes: (i) a set of nodes, one for each (time period, due date) pair, and (ii) a set of nodes, one for each time at which one or more trains is scheduled. Each feasible path
in the network passes through one or more nodes in the first set, and exactly one node in
the second set. If the shipment is never held in inventory at the hub, it will pass through
exactly one node in each set.

Arcs from the source to the first set of nodes have upper and lower capacity limits equal
to the number of container arrivals, differentiated only by time of arrival at the hub, and
due date. Arcs from nodes in the first set to nodes in the second set represent container
shipments on a given train from the hub to the destination. From each node in the second
set, containers flow into a sink along an arc with a lower bound of zero and an upper
bound equal to the capacity of the train(s) scheduled at the relevant time period. Indirect
transportation costs are assessed on the arcs between the (time period, due date) node
and the corresponding train node. No costs are assessed on the arcs originating at the
supersource or terminating at the supersink.

In addition to arcs that represent the movement of containers from the hub to the
destination, inventory arcs at the hub link one category of containers, differentiated by due
date with the corresponding type in the following time period. Container flows on these arcs
represent storage of containers at a location from one time period to the next. Inventory
holding costs are assessed along these arcs.

One of our proposed methods for solving the original problem, which we call the **fixed
train and container scheduling problem**, takes full advantage of both of the following char-
acteristics: (i) the ability to decompose the whole problem into the origin scheduling and
the hub scheduling problem, and to decouple the origin scheduling problem by origin, and
(ii) to relax the integrality requirement on all container variables while preserving an inte-
ger solution. We determine a train schedule and the corresponding container movements
outbound from the origin, and fix the direct train schedules to all destinations, the indirect
train schedules into the hub, and the corresponding direct and indirect container routing
decisions. Of these container shipments, we consider the indirect container arrivals into the
hub. We use this as input information in the hub scheduling problem to determine the train
schedule and corresponding indirect container shipments to each destination. We then fix
the indirect train schedule outbound from the hub, and the corresponding indirectly routed
containers outbound from the hub.

This and other variations on the methodology for obtaining solutions to these two suc-
cessive parts of the original problem involve a tradeoff between the degrees of freedom we allow, and the availability of information necessary for solving each subproblem. We discuss this tradeoff in more detail in Section 4.4.

4.2 Suboptimality of the Decomposition Procedure

As explained in the previous section, we take advantage of the fact that we can define additional cost parameters for the origin scheduling problem to more accurately reflect the consequences of making decisions at the origin which influence the total cost of the hub scheduling problem. However, because the subproblems are decoupled by origin, opportunities to coordinate shipments outbound from the hub are partially sacrificed. The assignment of containers to trains in the origin scheduling problem will affect their slack time at the hub. This, in turn, will affect the scheduling of trains outbound from the hub. A greater amount of slack time available in an indirect container's schedule when it arrives at the hub allows for more flexibility in the indirect train schedule at the hub and more consolidation of shipments. This results in fewer indirect trains being sent from the hub, and a lower cost solution to the hub scheduling problem. Because the origin scheduling problem cannot plan for maximum scheduling flexibility at the hub (i.e., it is indifferent among many decisions that may be significant in the overall cost of the hub scheduling problem), our decomposition procedure will not necessarily yield an optimal solution to the original problem.

As a small example, let us consider the following problem: Assume that direct travel time between any origin and any destination is four days, and indirect travel time is five days, which includes two days' travel time to the hub, one day spent at the hub, and two days traveling from the hub to the destination. Suppose train capacity is 200 containers, and we have arrival data for containers at two origins bound for a single destination as given in Table 4.1. Tables 4.2 and 4.3 present two equally good train schedules for the origin scheduling problem.

In each of the schedules given in Tables 4.2 and 4.3, one direct and two indirect trains are scheduled. Roughly the same cost is assessed for each of these scenarios in the origin scheduling problem. Note that in the first scenario, a full direct train is sent at time period 1 between Seattle and Chicago carrying containers with due dates of 6 and 8. However,
<table>
<thead>
<tr>
<th></th>
<th>origin</th>
<th>destination</th>
<th>number of containers</th>
<th>due date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>Sea</td>
<td>Chi</td>
<td>100</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Sea</td>
<td>Chi</td>
<td>100</td>
<td>8</td>
</tr>
<tr>
<td>Day 2</td>
<td>Sea</td>
<td>Chi</td>
<td>100</td>
<td>7</td>
</tr>
<tr>
<td>Day 3</td>
<td>Oak</td>
<td>Chi</td>
<td>100</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 4.1: Container Arrival Data for Example

<table>
<thead>
<tr>
<th>train type</th>
<th>origin</th>
<th>&quot;destination&quot;</th>
<th>number of containers</th>
<th>due date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>Direct</td>
<td>Sea</td>
<td>Chi</td>
<td>100-6;100-8</td>
</tr>
<tr>
<td></td>
<td>Indirect</td>
<td>Sea</td>
<td>Salt Lake</td>
<td>100-7 (Chi)</td>
</tr>
<tr>
<td></td>
<td>Indirect</td>
<td>Oak</td>
<td>Salt Lake</td>
<td>100-8 (Chi)</td>
</tr>
</tbody>
</table>

Table 4.2: Train Schedule into Hub: Nonoptimal for Hub Scheduling Problem

<table>
<thead>
<tr>
<th>train type</th>
<th>origin</th>
<th>&quot;destination&quot;</th>
<th>number of containers</th>
<th>due date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>Indirect</td>
<td>Sea</td>
<td>Salt Lake</td>
<td>100-8 (Chi)</td>
</tr>
<tr>
<td></td>
<td>Direct</td>
<td>Sea</td>
<td>Chi</td>
<td>100-6; 100-7</td>
</tr>
<tr>
<td></td>
<td>Indirect</td>
<td>Oak</td>
<td>Salt Lake</td>
<td>100-8 (Chi)</td>
</tr>
</tbody>
</table>

Table 4.3: Train Schedule into Hub: Optimal for Hub Scheduling Problem
when the indirect containers arrive at the hub, 100 containers bound for Chicago with due
date 7 have no slack time in their schedule. Hence, they must be shipped immediately at
time 5 on an indirect train bound for Chicago. When the 100 containers from Oakland
bound for Chicago arrive at the hub a day later, another indirect train must be sent to
the same destination, resulting in two indirect trains being sent from Salt Lake to Chicago.
Now let us consider the second scenario in which the same number of direct and indirect
trains are scheduled. The direct train (still full) is sent between Seattle and Chicago at
time period 2 rather than time period 1, carrying containers with due dates of 6 and 7.
An indirect train from Seattle at time period 1 with 100 containers due at Chicago at time
period 8, and an indirect train from Oakland at time period 3 with 100 containers, likewise
due at Chicago at time period 8 are both sent. Under this scenario, when containers arrive
at the hub, they can be sent via a single indirect train arriving at Chicago by time period
8. In this case, fewer indirect trains are sent from the hub, even though the solutions to
the origin scheduling problems produce the roughly same objective function value for the
origin scheduling problem.

Assessing a higher cost to hold a container at the origin than at the hub in the origin
scheduling problem may encourage this problem to yield solutions in which containers are
shipped from the origin as quickly as possible to allow for maximum scheduling flexibility
at the hub. However, this modification is not very effective, as it does not guarantee that
available containers with earlier due dates would all be placed on a direct train before
containers with more slack in their schedule are placed on the same direct train.

Alternatively, we could add logical constraints and valid inequalities to our origin schedul-
ing subproblems, requiring that containers with earlier due dates be sent on direct trains, if
such trains are sent. Similarly, we could add constraints that force containers with earlier
due dates to be shipped on indirect trains before those with later due dates, even if there
is enough slack time available in the schedule to allow the more urgent containers to be
shipped later. In this way, containers with the least slack time in their schedule would
avoid the hub, if possible, or, at a minimum, would arrive at the hub as early as possible.
We can model this by imposing the following constraint set for every valid \((t,l)\) combina-
tion, i.e., for each value of \(t\) over the time horizon and each relevant due date such that
\(l \geq t + \alpha_{ik}\).
\[ \begin{align*}
    b_{iklt} - x_{iklt} & \leq M \ast (1 - \zeta_{ikl}) & \forall i, k, t, l \geq t + \alpha_{ik} \\
    b_{iklt} - x_{iklt} & \geq 1 - \zeta_{ikl} & \forall i, k, t, l \geq t + \alpha_{ik} \\
    x_{iklt} & \leq M \ast \zeta_{ikl} & \forall i, k, t, l \geq t + \alpha_{ik} + 1
\end{align*} \] (4.23) (4.24) (4.25)

where \( M \) is a sufficiently large number and \( \zeta_{ikl} \) is a binary variable, with one such variable for each origin-destination-level of service combination, and each set of constraints. When the value of \( \zeta_{ikl} \) is 0, for a given origin-destination pair, the number of containers due in period \( l \) but not yet shipped in period \( t \) is positive, as indicated by (4.23) and (4.24). Hence, for that origin-destination pair, constraints (4.25) force the number of containers due later than period \( l \) and shipped in period \( t \) to be 0. Conversely, if \( \zeta_{ikl} \) is 1, indicating that for a given origin-destination pair, all containers due at time period \( l \) have been shipped (implied by (4.23) and (4.24)), then constraints (4.25) allow the number of containers due later than time \( l \) and shipped in period \( t \) to be positive. The constraints on the indirect shipments have a similar structure.

However, when we add these constraints to our origin scheduling problems, it substantially reduces their tractability because we are imposing constraints on the container variables. Because the values for the container variables remain integral in a basic solution without adding this restriction to the formulation, constraining the container movements has the same effect as adding constraints to a linear program, i.e., it makes it harder. This is in contrast to the consequences of placing constraints on the train variables in the origin scheduling problem, which, as will be discussed in the following chapter, makes the problem easier to solve (i.e., improves the lower bound) by restricting the solution space.

Hence, a clearly observable cause for the nonoptimality of our decomposition procedure results from the inefficiencies we incur in scheduling indirect trains outbound from the hub due to the difficulty in regulating the order in which containers are sent out from the origin in our origin scheduling subproblems.

### 4.3 Methods to Handle Many Destinations

Because our decomposition approach decouples the problem by origin, for problem instances containing more origins, we simply solve more, but not, \textit{a priori}, larger subproblems. Hence,
the quality of solutions we obtain from our subproblems does not deteriorate as the number of origins increases.

Unfortunately, our decomposition procedure does not allow for decoupling of the origin scheduling problem by destination. Hence, the size of the subproblems grows as the number of destinations increases. In fact, for some subproblems containing as few as four destinations, CPLEX yields solutions that are guaranteed to be no better than about 20% to 30% from the optimal solution. In these cases, using the best integer solution that we can obtain to the origin scheduling problem along with the solution to the hub scheduling problem results in solutions to the original problem that are 2%-3% worse than could be obtained via a straightforward implementation of CPLEX. Because of this poor performance, we develop a procedure to obtain solutions to the origin scheduling problem for an arbitrarily large number of destinations.

The rationale behind our procedure is as follows: When establishing a direct train schedule between an origin and a destination, we are specifically concerned with that particular origin-destination pair. Hence, aggregating the other \( n - 1 \) destinations does not result in a substantial loss of information in terms of establishing a direct train schedule, because the containers that are to be sent to the other \( n - 1 \) destinations are of secondary concern when constructing the direct train schedule between the origin and the \( n^{th} \) destination. Hence, these \( n - 1 \) destinations can be thought of collectively as a "super-destination." After direct train schedules are established for all origins and destinations, we simultaneously determine the indirect train schedules and the routing of both the direct and the indirect container traffic outbound from the origin, with the direct train schedule fixed. Making these decisions simultaneously allows us to incorporate the dependence between direct and indirect trains outbound from the same origin. However, because we establish direct train schedules for each origin separately, we lose information regarding economies of scale in sending containers indirectly. The loss of information resulting from this procedure is discussed after we explain our method.

For each origin scheduling problem, decoupled by origin, we consider \( n \) different subproblems, where \( n \) corresponds to the number of destinations in the original problem. In each of these sub-subproblems, we partition the set of destinations into two groups: (i) a single destination, and (ii) the remaining \( n - 1 \) destinations which we aggregate into a
"super-destination." Figure 4.4 represents the aggregation of multiple destinations for one origin and aggregated-destination pair.

Demands are aggregated and weighted average fixed and variable costs are assessed for the direct and indirect routes between the origin and the aggregated destination. To accurately reflect travel time differences, the due dates should be adjusted on those origin-destination and hub-destination routes with longer travel times to reflect the need for the container to depart the origin sooner. This can be done as follows: Decrease indirect travel time on all hub-destination routes to the shortest travel time. Decrease container due dates to reflect this travel time change for all routes on which the travel time was decreased. Finally, shorten travel time commensurately on all direct origin-destination routes to accurately reflect the original time differential between sending a shipment directly vs. indirectly.

The aggregated problem is now treated as a two-destination problem. For each aggregated two-destination subproblem, we consider the original question of how to optimally schedule both direct and indirect trains and the corresponding containers on these trains. For each origin, we solve such an origin-scheduling problem for each destination, aggregating the other \( n - 1 \) destinations.
Let us define the following notation for each origin-destination pair:

Let \( \iota \) denote the origin of concern, i.e., one of \( \{i = 1...m\} \)

Let \( \kappa \) denote the destination of concern, i.e., one of \( \{k = 1...n\} \)

Let \( \Upsilon \) denote the set of \( n - 1 \) destinations, i.e., not including \( \kappa \)

Then, using these additional parameters in conjunction with the parameters and variables defined earlier, we formulate the origin scheduling problem for given \( \iota \) as follows:

\[
(P_{agg}^o):\quad \min \quad Z_{agg}^o = \sum_{ktl} h \cdot P_{ikt}^o + \sum_{ktl} c_{tkl} \cdot x_{ikt} + \sum_{jktl} c_{jkl} \cdot x_{jktl} + \sum_{ktl} P_{ikt}^o \cdot x_{ikt} + \sum_{jktl} P_{jkl}^o \cdot x_{jklt} + \sum_{kt} S^o_{tk} \cdot x_{ktl} + \sum_{jt} S^o_{tj} \cdot x_{jlt}
\]

subject to

\[b_{iktl} = x_{ikt} \quad \forall k \in \{\kappa, \Upsilon\}, t = 1, t + \alpha_{ik} \leq l < t + \beta_{ik} + \gamma_{ik} + \delta_{ik} (4.26)\]

\[b_{iktl} = P_{ikt}^o + x_{ikt} + \sum_{j} x_{ijklt} \quad \forall k \in \{\kappa, \Upsilon\}, t = 1, l \geq t + \beta_{ik} + \gamma_{ik} + \delta_{ik} (4.27)\]

\[b_{iktl} = -P_{ikt(t-1)}^o + x_{ikt} \quad \forall k \in \{\kappa, \Upsilon\}, t > 1, t + \alpha_{ik} \leq l < t + \beta_{ik} + \gamma_{ik} + \delta_{ik} (4.28)\]

\[b_{iktl} = P_{ikt}^o - P_{ikt(t-1)}^o + x_{ikt} + \sum_{j} x_{ijklt} \quad \forall k \in \{\kappa, \Upsilon\}, t > 1, l \geq t + \beta_{ik} + \gamma_{ik} + \delta_{ik} (4.29)\]

\[\sum_{l} x_{ikt} \leq C \cdot x_{ikt} \quad \forall k \in \{\kappa, \Upsilon\}, t \quad (4.30)\]

\[\sum_{kl} x_{ikt} \leq C \cdot x_{ikt} \quad \forall j, t \quad (4.31)\]

\[P_{ikt}, x_{ikt} \geq 0 \text{ and integer} \quad \forall k \in \{\kappa, \Upsilon\}, t, l \quad (4.32)\]

\[x_{ijklt} \geq 0 \text{ and integer} \quad \forall j, k \in \{\kappa, \Upsilon\}, t, l \quad (4.33)\]

\[x_{ikt} \geq 0 \text{ and integer} \quad \forall k \in \{\kappa, \Upsilon\}, t \quad (4.34)\]

\[x_{ijt} \geq 0 \text{ and integer} \quad \forall j, t \quad (4.35)\]

From the solution, we retain only the direct train schedule between the origin and the single (unaggregated) destination for each origin-destination pair. Note that the train schedule we obtain for the aggregated destination is not necessarily even feasible for the
original problem. Then, with these direct train schedules fixed, we solve an origin scheduling problem to determine the indirect train schedule. In order to consider all available information, this indirect train schedule must be established considering the hub and all destinations simultaneously. This allows container traffic to be allocated to the fixed direct trains, which will, in turn, affect the indirect train schedule. As such, the procedure is not independent of the number of destinations in the problem. However, the problem of determining only indirect trains and the corresponding container movements is much easier than that of determining both direct and indirect train movements simultaneously with all container movements. Empirically, we have been able to solve indirect train origin scheduling problems with six destinations in a matter of seconds. Therefore, we anticipate being able to solve origin scheduling problems of a practical size with the direct trains fixed.

If desired, we can modify our procedure to be independent of the number of destinations. When we solve the aggregated version of the problem and establish the direct train schedules for each origin-destination pair, we can also fix the containers being sent on these trains. The remaining problem reduces to one of determining schedules for trains and containers bound solely for a common hub. Thus, the number of destinations is irrelevant. As will be discussed in Section 4.4, however, fixing variables locally causes some deterioration in the solution quality.

Note that this procedure generally will not provide an optimal solution to the unaggregated subproblem because of the loss of information when destinations are consolidated, and because of the way in which direct train schedules are first determined and then fixed. Economies of scale and the benefits of consolidation at the hubs are not fully taken into account because interactions with other origins are ignored. This often leads to the construction of a train schedule consisting of more than the optimal number of direct trains. As an example, consider the following origin subproblem with one period and three destinations and container arrival data given in Table 4.4. Assume a train has a capacity of 200 containers. Using the procedure described above and assuming that 180 containers constitute a sufficiently full train to send directly, we establish the schedule shown in Table 4.5. With the direct train schedule fixed, solving for the indirect train schedule, we obtain a complete train schedule for the origin subproblem consisting of: (i) a direct train between Seattle and Chicago, (ii) a direct train between Seattle and Memphis, and (iii) an indirect
### Table 4.4: Data for Aggregation Procedure

<table>
<thead>
<tr>
<th>origin</th>
<th>destination</th>
<th>number of containers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seattle</td>
<td>Chicago</td>
<td>180</td>
</tr>
<tr>
<td>Seattle</td>
<td>Memphis</td>
<td>180</td>
</tr>
<tr>
<td>Seattle</td>
<td>New Orleans</td>
<td>20</td>
</tr>
</tbody>
</table>

### Table 4.5: Direct Train Schedule under the Aggregation Procedure

<table>
<thead>
<tr>
<th>origin</th>
<th>destination</th>
<th>aggregated destination</th>
<th>train schedule between origin and destination</th>
<th>train schedule between origin and aggregated destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seattle</td>
<td>Chicago</td>
<td>Memphis-New Orleans</td>
<td>1 direct</td>
<td>1 indirect</td>
</tr>
<tr>
<td>Seattle</td>
<td>Memphis</td>
<td>Chicago-New Orleans</td>
<td>1 direct</td>
<td>1 indirect</td>
</tr>
<tr>
<td>Seattle</td>
<td>New Orleans</td>
<td>Chicago-Memphis</td>
<td>1 indirect</td>
<td>1 direct or indirect</td>
</tr>
</tbody>
</table>

train between Seattle and New Orleans.

However, we can identify a schedule for the unaggregated version of the problem which is clearly better than the above schedule, namely to send one direct train (either between Seattle and Chicago, or between Seattle and Memphis) and one indirect train (carrying the shipments not routed directly). The latter schedule may still be suboptimal for the entire problem, but serves to illustrate that a better schedule can be found to the origin subproblem when we consider all destinations and both direct and indirect train scheduling simultaneously. In fact, there is a loss of information associated with decoupling the problem into the origin scheduling problem and the hub scheduling problem (cf. Section 4.2). We incur this loss regardless of whether we use this aggregation technique or consider all \( n \) destinations simultaneously when establishing a direct and indirect train schedule for the origin scheduling problem. We subsequently quantify the effects of this loss of information.

When the origin scheduling problems are large enough (i.e., they contain sufficiently many destinations) that CPLEX terminates prior to finding an optimal solution, the use of this aggregation procedure may be required to obtain a good solution to the original problem. Our numerical results in Section 4.5 show that the solutions from this procedure are better than the ones obtained via a straightforward implementation of CPLEX.
4.4 The Value of Centralized Decision Making

4.4.1 The Procedures

The solutions to subproblems found via the decomposition procedure described above can be used to construct a solution to our original problem through a heuristic nested optimization procedure.

Recall that $Z$ represents the objective function value for the original problem, and let $Z^*$ denote the optimal objective function value. Additionally:

$Z_1 =$ the objective function value for the origin scheduling problem with the original cost parameters $S_{ij}^o$ and $l_i^o$

$\overline{Z}_1 =$ the objective function value for the origin scheduling problem with the adjusted cost parameters, $\overline{S}_{ij}$ and $\overline{l}_{ij}$

$Z_2 =$ the objective function value for the hub scheduling problem

Let $x_i$ and $u_j$ be shorthand notation for all container flows from origin $i$ and hub $j$, respectively:

$x_i = \{z_{ikt}, x_{ijkl}\} \ \forall i, j, k, t, l$

$u_j = \{u_{ijkl}\} \ \forall i, j, k, t, l$

Finally, let $z_i$ and $z_j$ denote the set of trains outbound from origin $i$ and hub $j$, respectively:

$z_i = \{z_{ikt}, z_{ikt}\} \ \forall i, j, k, t$

$z_j = \{z_{jk}\} \ \forall j, k, t$

In solving any subproblem in isolation, all relevant constraints apply. For example, in solving the origin scheduling problem, we require conservation of flow of containers at the origin, and train capacity constraints on both direct and indirect trains leaving the origin. For the hub scheduling problem, we require conservation of flow of the containers at the hub, and adherence to capacity constraints for all trains outbound from the hub. In both parts of the problem, containers must be shipped far enough in advance to arrive at the destination on time.
The original problem can be posed as a nested optimization problem:

\[
Z^* = \min_{x_1} \left( \min_{x_1} \{ Z_1(x_1|x_1) \} + \min_{u_j, z_j} Z_2(u_j, z_j|x_1) \right)
\]

The inner problem is the hub scheduling problem, given container flows into the hub. The middle-level problem involves optimizing container flows outbound from the origin, taking into account both the costs incurred at the origin and the resulting (optimal) cost for the hub scheduling problem. The outer optimization problem establishes the train schedules outbound from the origins. The problem could be solved, in principle, using this approach, but would pose two formidable challenges. First, although the hub scheduling problem is easy to solve, the optimum cost for each destination is not a smooth function of the container arrivals. Thus, choosing the container flows in the middle-level problem is not easy. Second, choosing the train schedules outbound form the origins is a difficult combinatorial problem because of the high degree of substitutability of direct and indirect trains, and of trains scheduled at different times. It is also difficult because the impact of the train schedule at the origins on the middle (and inner) problem is indirect: it only defines constraints on certain container flows.

We develop a decomposition approach that was motivated by the nested optimization problem above, but avoids some of the difficulties by approximating the impact of the train schedule at the origin on the cost of the hub scheduling problem. Using our decomposition procedure, we can find a solution for the original problem in one of several ways. These variations are based on the tradeoff between the degrees of freedom we allow in solving various optimization problems, and the information we must possess in order to solve each of the problems. The more degrees of freedom we allow, the better is the quality of our solution, but the more coordination is required between various parts of our system.

Let us consider the following three scenarios in decreasing order of degrees of freedom:

1. From our decomposition procedure, we can determine a solution as follows: First, solve the origin scheduling problem separately for each origin. Then, fixing the resulting schedule of trains outbound from the origin, solve the problem of scheduling the trains outbound from the hub and all container movements (i.e., both those outbound from the origin and those outbound from the hub). We refer to making these decisions in their entirety as the fixed origin train scheduling problem. Here, we allow the most degrees of freedom since we only fix the schedule of trains outbound from the origin. However, for practical purposes, this
allows each origin to determine only its train schedule independently of the other origins. The indirect train schedule outbound from the hub, and the container scheduling problem both at the origins and at the hub are determined for the system simultaneously. Hence, information about the train schedules at all origins is needed to obtain a solution for all container movements and the train schedule outbound from the hub.

Mathematically, this corresponds to the following optimization problems:

1. \( \text{Solve: } \min_{x_i, z_i} \bar{Z}_1(x_i, z_i) \)

2. Then, given \( z_i \) from step 1, solve: \( Z^{(1)} = \min_{x_i} \{ Z_1(x_i|z_i) + \min_{u_j, z_j} Z_2(u_j, z_j|x_i) \} \)

A second variation is as follows: Again, solve the origin scheduling problem separately for each origin. In order to solve the hub scheduling problem for each destination, use the matrix of container arrivals at the hub, as determined by the origin scheduling problem, and solve the hub scheduling problem. Fix the resulting train schedules outbound from the origin and outbound from the hub as determined from the origin and hub scheduling problems, respectively, to determine all container movements. We term this method the fixed train scheduling problem. Although now all train schedules (i.e., both those at the origin and those at the hub) are determined with only container arrival data from the origin scheduling problem, the systemwide container scheduling problem (for all origins and the hub) again requires information as to the train schedules from all origins and the hub.

Mathematically, this corresponds to the following optimization problems:

1. \( \text{Solve: } \min_{x_i, z_i} \bar{Z}_1(x_i, z_i) \)

2. Then, given \( x_i \) from step 1, solve: \( \min_{u_j, z_j} Z_2(u_j, z_j|x_i) \)

3. Finally, given \( x_i \) from step 1 and \( z_j \) from step 2, solve:

\[
Z^{(2)} = \min_{x_i, z_j} \{ Z_1(x_i|z_i) + Z_2(u_j|z_j) \}
\]

76
Finally, we can solve the origin scheduling problem and fix the solution for both the train schedule and the associated container movements. Then, using the resulting container arrivals at the hub, we determine the train schedule and associated container movements outbound from the hub by solving the hub scheduling problem. This is the method described in Section 4.1, which we call the fixed train and container scheduling problem. In this scenario, each location determines its decisions either independently (in the case of the origin), or given previous decisions (in the case of the hub). As a result, no systemwide information is necessary to solve any part of the problem.

Mathematically, this corresponds to the following optimization problems:

1. Solve: \( \min_{x_1, x_1} Z_1(x_1, x_1) \)

2. Then, given \( x_1 \) from step 1, solve: \( \min_{u_j, z_j} Z_2(u_j, z_j | x_1) \)

3. Finally, set \( Z^{(3)} = Z_1(x_1, x_1) + Z_2(u_j, z_j) \)

We expect to obtain better solutions when we allow for more degrees of freedom in solving our entire problem. On the other hand, such an approach can have several drawbacks. In large problem instances, it may be impossible to obtain good solutions when many decisions must be made simultaneously. Using the fixed train scheduling problem or the fixed train and container scheduling problem may be necessary, and also allows us to obtain an indirect train schedule outbound from the hub by decoupling the hub scheduling problem by destination.

As another consideration, more freedom in determining the systemwide “optimum” requires more information. For example, using either the fixed origin train scheduling problem or the fixed train scheduling problem, requires information about the train schedule at each node before determining container shipments from any node. Additionally, it may be impossible to determine the schedule of trains outbound from the hub far enough in advance to adequately plan the container schedule outbound from the origin.

We initially report the best objective function value found with the most degrees of freedom allowed using our decomposition procedure. Subsequently, we analyze the tradeoffs between the availability of information and its effect on the objective function value.
4.4.2 A Note on the Subproblem Structure

We have shown in Section 4.1 that for a capacity-feasible, integer-valued solution for the train schedule, each origin scheduling and hub scheduling problem can be formulated as a minimum cost network flow problem. As such, with integer arrivals to the relevant subsystems and integer train capacities, all basic solutions to the linear programming relaxations of the subproblems are integral. Because the solution methodology for the fixed train and container scheduling problem entails solving exclusively these network flow subproblems, we can relax integrality constraints on the container flows without loss of generality.

On the other hand, the fixed origin train scheduling problem and the fixed train scheduling problem require the solution of subproblems in which the entire train schedule is fixed and the optimal container flows must be identified for this train schedule. For these cases, to legitimately relax the integrality constraints on the container variables, it is necessary to prove that an integer optimal solution to the container flow problem exists; otherwise the container flows may be fractional.

In solving an array of test problems (and many container flow subproblems for various fixed train schedules for each of these test problems), we observed that CPLEX always produces solutions in which the container flows are integral even when the integrality constraints on these variables are relaxed. This motivated us to investigate the phenomenon further. Although we have not been able to prove the result formally, we state our conjecture and then explain why we believe that either (i) the conjecture is true or, at a minimum, that (ii) the number of fractional containers in the optimal solution is small enough that a simple rounding procedure would be adequate in practice, because the train capacity constraints are not rigid.

Conjecture 1 For any capacity-feasible train schedule, the minimum cost container flow problem has integral basic feasible solutions. Hence, the linear programming relaxation always produces integral solutions.

Recall that in our origin scheduling problem, we use modified setup and variable transportation costs for trains routed to the hub. The purpose of these cost modifications is to capture the estimated effect of costs incurred outbound from the hub that cannot be represented accurately in the origin scheduling problem. Consider another version of the
origin scheduling problem in which the setup and variable transportation costs are not modified; instead, they are captured in the hub scheduling problem. In effect, all costs are now reflected accurately in each of the relevant subproblems. This new version of the origin scheduling problem retains the network structure for a given train schedule.

We now have several origin scheduling subproblems, one for each origin, and several hub scheduling subproblems, one for each hub. We observe that the only interdependence among these subproblems that is ignored in the decomposition is that the “demands” in the origin scheduling subproblems correspond to container arrivals at the hub in the hub scheduling problem. For any feasible solution, these conservation of flow constraints must hold as equalities, and require that the sum of container arrivals at the hub at a given time with a common due date and destination be equal to the corresponding demands from the origin scheduling subproblems. These containers may arrive from all origins and may have various arrival dates at the respective origins. If we knew the optimal matrix of such intermediate demands, we could find an optimal solution for the entire problem by solving the network flow subproblems. Moreover, if we can identify an integral optimal matrix of intermediate demands, then the solution for the network flow subproblems, and thus for the entire problem, would have integral container flows. Indeed, the solution will be non-integral only if all optimal matrices of intermediate demands contain some fractional values.

In a more general setting where commodities consume different amounts of capacity on the arcs, fractional intermediate demands could readily occur. On the other hand, when all commodities require one unit of capacity on the arcs and all arc capacities are integral, the capacity constraints do not inherently lead to fractional flows. Rather, if fractional solutions arise, they are a consequence of interactions between the capacity constraints and the flow costs. Under our assumption that the variable transportation cost between two locations is constant over time, the total variable transportation cost is the same for all routings from an origin to a destination through the hub. Thus, there is no economic motivation to split a container between routes. The only difference in cost among routings lies in the total holding cost incurred by the container; however, these costs are assessed on uncapacitated inventory arcs, are the same for all containers, and are constant over time. Thus, there is no economic incentive for holding fractional portions of two or more types of containers in
inventory rather than one unit of a single container type. We have been unable to identify any effects of the capacity constraints or the economics that would necessitate or motivate fractional flows. A formal proof remains a subject for further research.

Note that fractional solutions can occur only when some train capacity constraints are tight. If all capacity constraints with slack are non-binding, an integer optimal solution necessarily exists, because the constraints can be removed without affecting the optimality of the integer program. The remaining (container routing) constraints form a network, which guarantees the existence of an integer optimal solution. Moreover, the pattern of container arrivals will typically create situations in which a partially full train must be sent to meet due date requirements. The corresponding train capacity constraint will therefore be non-binding. Observe also that a greater number of capacity constraints will be binding when economic incentive exists to ship containers as late as possible. If this is the case, containers with alternate feasible routes are only shipped early to take advantage of transportation economies of scale. Our model assumes that transportation costs are constant over time, and that inventory costs are only assessed at the origin and the hub. This cost structure provides incentive to ship containers early, which results in fewer full trains, and therefore fewer binding constraints.

To demonstrate that if a solution of the linear programming relaxation contains fractional values, the maximum number of these values is small, we introduce a path-based formulation of the container routing problem for a given train schedule. In order to simplify the analysis, we do not include inventory arcs and their associated costs, which appear in the original formulation. Because these arcs are uncapacitated, they can be ignored without loss of generality for the purpose of the current analysis. Note that for ease of exposition, the notation we introduce here is specific to this section, although similar notation is used elsewhere in the dissertation to represent different parameters and variables.

Let:

\[ i = \text{the index of a commodity, where commodities are differentiated by origin, destination, arrival date at the origin, and due date} \]

\[ k = \text{the index of a path, defined as a feasible route between an origin and an destination, i.e., a pair of trains, one traveling to the hub at a specified time, and another from the hub at a feasible time following the arrival of the inbound train} \]
\( t \) = the index of a train, distinguished by origin, destination, and time period (where either the "origin" or the "destination" may be the hub)

\( C_t \) = the capacity of train \( t \) (number of containers)

\( D_i \) = the demand for commodity \( i \)

\( c_{ik} \) = the cost for one unit of commodity \( i \) to traverse path \( k \)

\( v_{kt} = 1 \), if path \( k \) uses train \( t \), and 0 otherwise

\( \psi_{ik} = 1 \), if commodity \( i \) can use path \( k \), and 0 otherwise

\( x_{ik} \) = the number of units of commodity \( i \) which use path \( k \)

We define a "path" as a pair of trains, one inbound to and another outbound from the hub, without loss of generality. We can model direct trains as indirect trains that pass through a hub without incurring the extra costs or delay in time, and without the ability to arbitrarily connect to any indirect train outbound from the hub.

Then a path-based formulation of our problem for a fixed train schedule is:

\[(P_P): \quad \min \sum_{ik} c_{ik} x_{ik}\]

subject to

\[\sum_{ik} v_{kt} x_{ik} \leq C_t \quad \forall \ t \quad (4.36)\]

\[\sum_k \psi_{ik} x_{ik} = D_i \quad \forall \ i \quad (4.37)\]

\[x_{ik} \geq 0 \quad \forall \ i, k \quad (4.38)\]

Because the cost of using a path is the same for all commodities, we can suppress the commodity index \( i \) in the cost coefficient \( c_{ik} \). Let \( I \), \( K \), and \( T \) be the number of commodities, paths, and trains, respectively. If the number of origin-destination pairs is large, the number of commodities will be large, and the number of variables, \( IK \), will be much greater than the number of constraints, \( I + T \). Even if all constraints in (4.36) are binding, any basic feasible solution to the problem would contain relatively few fractional trains because there can be at most \( I + T \) basic variables divided among \( I \) commodities, where \( I >> T \). Therefore, even if most of the constraints in (4.36) are binding, given a fractional optimal solution, the number of these fractional values will probably be low. Practically speaking,
<table>
<thead>
<tr>
<th>problem instance group</th>
<th>number of origins-hubs destinations</th>
<th>relative proportion expedited service demanded</th>
<th>ratio of origin to hub indirect setup cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>3-1-3</td>
<td>$\sim \leq 5%$</td>
<td>$\sim 4:1$</td>
</tr>
<tr>
<td>6-10</td>
<td>3-1-3</td>
<td>$\sim 20%$</td>
<td>$\sim 1:1$</td>
</tr>
<tr>
<td>11-15</td>
<td>3-1-3</td>
<td>$\sim 10%$</td>
<td>$\sim 1:1$</td>
</tr>
<tr>
<td>16-18</td>
<td>3-1-4</td>
<td>$\sim 20%$</td>
<td>$\sim 1:1$</td>
</tr>
<tr>
<td>19-20</td>
<td>3-1-4</td>
<td>$\sim 10%$</td>
<td>$\sim 1:1$</td>
</tr>
<tr>
<td>21-23</td>
<td>4-1-3</td>
<td>$\sim 20%$</td>
<td>$\sim 1:1$</td>
</tr>
<tr>
<td>24-25</td>
<td>4-1-3</td>
<td>$\sim 10%$</td>
<td>$\sim 1:1$</td>
</tr>
<tr>
<td>26-30</td>
<td>6-1-6</td>
<td>$\sim 20%$</td>
<td>$\sim 1:1$</td>
</tr>
<tr>
<td>31-35</td>
<td>6-1-6</td>
<td>$\sim 10%$</td>
<td>$\sim 1:1$</td>
</tr>
</tbody>
</table>

Table 4.6: Test Problem Characteristics

because the capacity restrictions are not absolutely rigid (i.e., a deviation of a few containers would be allowed), solutions containing small numbers of fractional shipments could be easily rounded to provide a nearly equivalent integral solution.

4.5 Numerical Results: Testing the New Decomposition Approach

4.5.1 Problem Parameters and Data

We solved thirty-five test problems with differing characteristics in order to reflect different rail network topologies, different container demand patterns, and different cost structures. We summarize problem characteristics in Table 4.6. Although the number of origins and destinations differs between problem types, each network contains only one hub. Container demand was generated randomly for each origin-destination-time-due date combination with a probability of 0.55 of being randomly generated from a discrete uniform distribution between 10 and 65, and a probability of 0.45 of being 0. Scenarios in which less expedited service was demanded were generated by setting demands to zero for randomly chosen origin-destination-time-due date combinations such that $t + \alpha_{ik} \leq l < \beta_{ij} + \gamma_{jk} + \delta_j$.

Table 4.7 gives ranges of values for container arrival rates and the cost parameters used in the thirty-five test problems. Rough calculations using industry data suggest that fixed and variable transportation costs for shipping a full load are approximately equal. That is, fuel and other transit related costs are approximately equal to the fixed cost of labor,
<table>
<thead>
<tr>
<th>parameter</th>
<th>range used in test problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>container arrival rate per day</td>
<td>0-65</td>
</tr>
<tr>
<td>setup cost at origin (direct train) ($/train)</td>
<td>11000-17000</td>
</tr>
<tr>
<td>setup cost at origin (indirect train) ($/train)</td>
<td>4000-8000</td>
</tr>
<tr>
<td>setup cost at hub ($/train)</td>
<td>2000-8000</td>
</tr>
<tr>
<td>transportation cost ($/container)</td>
<td>40-100</td>
</tr>
<tr>
<td>handling cost (all nodes) ($/container)</td>
<td>1-2</td>
</tr>
<tr>
<td>inventory holding cost ($/container/day)</td>
<td>1.5-2</td>
</tr>
</tbody>
</table>

Table 4.7: Parameters for Test Problem Instances

locomotive repositioning, and other activities related to running a train. Correspondingly, we assign the setup cost for a train to be approximately equal to the transportation cost per container multiplied by the capacity of a train (i.e., the entire variable “transportation cost” for a full train). We estimate the setup costs at the origin and the hub to be commensurate to the distance traveled between locations. Handling costs per container are based on an hourly wage of yard operators and the approximate time needed to load a container onto a stack car. Holding costs per container per day are assigned a low value as a slight deterrent to occupying yard space with containers. In actuality, the opportunity cost of capital, which constitutes most of the holding cost, is incurred by the shipper or consignee, and not the railroad.

The relative values of the parameters change between problem instances in order to reflect particular physical settings. For example, handling costs that are lower at the hub than at the origin and destination reflect greater handling requirements at the initial and terminal nodes. This could result from the need to move containers on and off the rail cars at the origin and destination, respectively, and only to re-order rail cars at the hub. Conversely, higher handling costs at the hub represent instances in which containers must be transferred between cars. Similarly, if the setup cost of an indirect train is higher at the origin than at the hub, it reflects a longer travel time between the origin and the hub, than between the hub and the destination. A reverse relationship in the relative magnitude of these setup costs would reflect the scenario in which the origin and the hub were closer than the hub and the destination. Although the indirect setup costs at the origin and the hub are each less than the direct setup cost at the origin, the total indirect setup cost per
train (i.e., the sum for the segments inbound to and outbound from the hub) is always more than the direct setup cost. Finally, capacity for each train is assumed to be 200 containers, which is consistent with the value used by a major national railroad.

It should be noted that the ability to obtain solutions from CPLEX in which the gap between the best feasible solution and the tightest lower bound is small, is very dependent upon the nature of the data. For example, solutions to problems with large arrival rates of containers tend to consist of more direct trains, making the schedule easier to determine. When setup costs are an order of magnitude lower, solution gaps of less than 10% are obtainable, even on problems commensurate in size to the sample problems discussed above. Our randomly generated data, which are consistent with actual data for our application, result in problem instances with larger initial gaps, as reported.

4.5.2 Initial Results

Our computational results suggest that it is often possible to solve both the origin scheduling subproblems and the hub scheduling problem to optimality for small problem instances with three origins, a hub, and three destinations. Nonetheless, even for the instances in which we are able to solve these subproblems to optimality, the resulting solution does not constitute an optimal solution for the original problem. However, in nearly all problem instances we have tested, our decomposition procedure yields a modest to significant improvement in the value of the objective function over that obtained directly via CPLEX. (Please see Appendix A for a complete description of the hardware and software specifications.) When computing the objective function value, \( Z^{(1)} \), we relax the integrality requirement on the container variables, as discussed in Section 4.4.2. This substantially reduces computation time and memory usage. We compare our objective function values for the fixed origin train scheduling problem to those found by CPLEX at the point at which the computer runs out of memory i.e., after the computer has been running for up to several hours. Table 4.8 illustrates the improvement in the objective function value between the best integer solution obtainable by CPLEX before the computer reaches its memory limit, and the best integer solution we obtain via our decomposition procedure. Abbreviating “best integer” and “lower bound” as “b.i.” and “l.b.”, respectively, this improvement is calculated as follows:
We then report the initial CPLEX gap, calculated as:

\[
\frac{\text{CPLEX}_{b,i} - \text{our}_{b,i}}{\text{CPLEX}_{b,i}} - 1
\]

Finally, we also give the percent reduction in the gap as a result of using our solution as the best integer solution, rather than the one initially found with CPLEX, calculated as:

\[
\frac{\text{CPLEX}_{b,i} - \text{our}_{b,i}}{\text{CPLEX}_{b,i} - \text{CPLEX}_{l,b}}
\]

In all but one instance, we obtain an improvement in the objective function value of about 3% to 7% over that found with a straightforward implementation of CPLEX. This equates to about a 15%-25% reduction in the initial CPLEX gap. Moreover, these results are insensitive to problem size.

4.5.3 Results: The Value of Centralized Decision Making

We next analyze the difference in objective function value for the three scenarios described in Section 4.4.1 used to obtain a solution to the original problem. The quality of the objective function obtained with the decomposition procedure deteriorates as the number of degrees of freedom we allow decreases. That is, when we fix the train schedule outbound from the origin and solve for the remaining decisions (the fixed origin train scheduling problem), the search space is sufficiently restricted to enable the software to yield a relatively good solution, even if the solution found under this scenario is not optimal. The next best objective value can be obtained from fixing the train schedule and coordinating the container movements throughout the system (the fixed train scheduling problem). When computing the objective function values, \(Z^{(1)}\) and \(Z^{(2)}\), to decrease computation time and memory usage, we relax the integrality requirement on the container variables, as discussed in Section 4.4.2. Finally, coordinating neither train scheduling nor container routing decisions results in the lowest objective function value (the fixed train and container scheduling problem). Again, when computing \(Z^{(3)}\), we relax the integrality requirement on the container variables, as discussed in Section 4.1. Although we did not implement it here, a postprocessing step should be used for the fixed train and container scheduling problem to improve the quality of the
<table>
<thead>
<tr>
<th>Problem</th>
<th>improvement in solution (%)</th>
<th>initial CPLEX gap (%)</th>
<th>reduction in gap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.7</td>
<td>34.8</td>
<td>18.1</td>
</tr>
<tr>
<td>2</td>
<td>3.1</td>
<td>28.4</td>
<td>14.1</td>
</tr>
<tr>
<td>3</td>
<td>6.1</td>
<td>38.0</td>
<td>22.1</td>
</tr>
<tr>
<td>4</td>
<td>3.3</td>
<td>14.2</td>
<td>26.1</td>
</tr>
<tr>
<td>5</td>
<td>4.0</td>
<td>31.2</td>
<td>16.7</td>
</tr>
<tr>
<td>6</td>
<td>1.4</td>
<td>27.6</td>
<td>6.9</td>
</tr>
<tr>
<td>7</td>
<td>2.2</td>
<td>28.5</td>
<td>9.8</td>
</tr>
<tr>
<td>8</td>
<td>4.0</td>
<td>24.9</td>
<td>26.9</td>
</tr>
<tr>
<td>9</td>
<td>0.5</td>
<td>24.7</td>
<td>2.8</td>
</tr>
<tr>
<td>10</td>
<td>5.6</td>
<td>23.0</td>
<td>30.0</td>
</tr>
<tr>
<td>11</td>
<td>2.8</td>
<td>20.5</td>
<td>16.6</td>
</tr>
<tr>
<td>12</td>
<td>-</td>
<td>22.9</td>
<td>-</td>
</tr>
<tr>
<td>13</td>
<td>4.3</td>
<td>20.4</td>
<td>25.5</td>
</tr>
<tr>
<td>14</td>
<td>2.5</td>
<td>20.5</td>
<td>14.6</td>
</tr>
<tr>
<td>15</td>
<td>5.3</td>
<td>26.3</td>
<td>25.5</td>
</tr>
<tr>
<td>16</td>
<td>3.3</td>
<td>25.1</td>
<td>15.9</td>
</tr>
<tr>
<td>17</td>
<td>2.3</td>
<td>28.3</td>
<td>9.2</td>
</tr>
<tr>
<td>18</td>
<td>2.9</td>
<td>24.0</td>
<td>15.0</td>
</tr>
<tr>
<td>19</td>
<td>4.4</td>
<td>30.4</td>
<td>18.8</td>
</tr>
<tr>
<td>20</td>
<td>4.3</td>
<td>24.2</td>
<td>22.3</td>
</tr>
<tr>
<td>21</td>
<td>1.3</td>
<td>27.0</td>
<td>5.9</td>
</tr>
<tr>
<td>22</td>
<td>4.1</td>
<td>28.7</td>
<td>19.5</td>
</tr>
<tr>
<td>23</td>
<td>4.2</td>
<td>24.3</td>
<td>21.4</td>
</tr>
<tr>
<td>24</td>
<td>3.3</td>
<td>25.6</td>
<td>16.4</td>
</tr>
<tr>
<td>25</td>
<td>4.9</td>
<td>24.9</td>
<td>24.9</td>
</tr>
<tr>
<td>26</td>
<td>3.7</td>
<td>29.4</td>
<td>15.0</td>
</tr>
<tr>
<td>27</td>
<td>5.1</td>
<td>31.0</td>
<td>21.3</td>
</tr>
<tr>
<td>28</td>
<td>3.8</td>
<td>27.0</td>
<td>17.8</td>
</tr>
<tr>
<td>29</td>
<td>3.5</td>
<td>28.3</td>
<td>16.2</td>
</tr>
<tr>
<td>30</td>
<td>4.0</td>
<td>30.7</td>
<td>17.3</td>
</tr>
<tr>
<td>31</td>
<td>7.6</td>
<td>29.0</td>
<td>33.8</td>
</tr>
<tr>
<td>32</td>
<td>6.6</td>
<td>27.7</td>
<td>30.3</td>
</tr>
<tr>
<td>33</td>
<td>5.3</td>
<td>29.5</td>
<td>23.1</td>
</tr>
<tr>
<td>34</td>
<td>6.6</td>
<td>25.7</td>
<td>32.3</td>
</tr>
<tr>
<td>35</td>
<td>5.0</td>
<td>25.9</td>
<td>24.3</td>
</tr>
</tbody>
</table>

Table 4.8: Improvement in the Best Integer Solution and Reduction in Gap
solution. Reassigning the containers to trains outbound from the origin in increasing order of due date before the hub scheduling problem is solved will allow for more flexibility at this intermediate node and, potentially, an improved solution for the system as a whole.

We find that in many instances, especially those in which the number of destinations is small enough not to require using the aggregated procedure (see Section 4.3), the three variations perform similarly. Hence, we conclude that even if all decisions are made locally, the quality of the solution for the system is not sacrificed. However, for the larger problem instances in which the aggregated procedure is used to obtain the best integer solution, we see a deterioration in the quality of the solution when implementing the fixed train scheduling problem over the fixed origin train scheduling problem, and (usually) a further deterioration in the quality of the solution when implementing the fixed train and container scheduling problem over the fixed train scheduling problem. This deterioration tends to be on the order of about 1% to 2%, but because the average improvement of the fixed origin scheduling problem over the solution obtained directly from CPLEX is, on average, about 5%, this deterioration can be significant. Table 4.9 gives the improvements over the solution obtained directly via CPLEX for the three variations. These improvements are calculated as the difference between the initial CPLEX solution and the the best integer solution found with one of the three variations divided by the initial CPLEX solution. Note that the quality of the solution deteriorates as the degrees of freedom decrease.

4.6 Time Performance of the Decomposition Procedure

We were able to substantially reduce computation time both for problems and for subproblems by relaxing the integrality requirements on the container variables. The validity of this relaxation can be established from the discussion in Section 4.4.2. The solution times reported here are those for obtaining a solution to the fixed origin train scheduling problem, which provides us with the best objective function values. Averaging across problem instances, we are able to obtain our (better!) solutions approximately as quickly (when computed in parallel) with this decomposition procedure compared to a straightforward implementation of CPLEX. Results for our test problems appear in Table 4.10. The first column lists the problem instance, and the next column reports the time at which the computer ran out of memory while solving the original problem using CPLEX. We then
<table>
<thead>
<tr>
<th>Problem</th>
<th>fixed origin train scheduling problem</th>
<th>fixed train scheduling problem</th>
<th>fixed train and container scheduling problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.7</td>
<td>4.7</td>
<td>4.7</td>
</tr>
<tr>
<td>2</td>
<td>3.1</td>
<td>3.1</td>
<td>3.1</td>
</tr>
<tr>
<td>3</td>
<td>6.1</td>
<td>6.1</td>
<td>6.1</td>
</tr>
<tr>
<td>4</td>
<td>3.3</td>
<td>2.9</td>
<td>2.9</td>
</tr>
<tr>
<td>5</td>
<td>4.0</td>
<td>4.0</td>
<td>4.0</td>
</tr>
<tr>
<td>6</td>
<td>1.4</td>
<td>1.4</td>
<td>1.4</td>
</tr>
<tr>
<td>7</td>
<td>2.2</td>
<td>2.2</td>
<td>2.2</td>
</tr>
<tr>
<td>8</td>
<td>4.0</td>
<td>4.0</td>
<td>4.0</td>
</tr>
<tr>
<td>9</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>10</td>
<td>5.6</td>
<td>5.3</td>
<td>5.3</td>
</tr>
<tr>
<td>11</td>
<td>2.8</td>
<td>2.8</td>
<td>2.8</td>
</tr>
<tr>
<td>12</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>13</td>
<td>4.3</td>
<td>2.9</td>
<td>2.9</td>
</tr>
<tr>
<td>14</td>
<td>2.5</td>
<td>2.5</td>
<td>2.5</td>
</tr>
<tr>
<td>15</td>
<td>5.3</td>
<td>5.3</td>
<td>5.3</td>
</tr>
<tr>
<td>16</td>
<td>3.3</td>
<td>2.3</td>
<td>2.2</td>
</tr>
<tr>
<td>17</td>
<td>2.3</td>
<td>0.8</td>
<td>0.7</td>
</tr>
<tr>
<td>18</td>
<td>2.9</td>
<td>1.7</td>
<td>1.7</td>
</tr>
<tr>
<td>19</td>
<td>4.4</td>
<td>4.0</td>
<td>3.9</td>
</tr>
<tr>
<td>20</td>
<td>4.3</td>
<td>3.2</td>
<td>3.2</td>
</tr>
<tr>
<td>21</td>
<td>1.3</td>
<td>0.4</td>
<td>0.3</td>
</tr>
<tr>
<td>22</td>
<td>4.1</td>
<td>4.1</td>
<td>4.1</td>
</tr>
<tr>
<td>23</td>
<td>4.2</td>
<td>4.2</td>
<td>4.2</td>
</tr>
<tr>
<td>24</td>
<td>3.3</td>
<td>3.3</td>
<td>3.3</td>
</tr>
<tr>
<td>25</td>
<td>4.9</td>
<td>4.9</td>
<td>4.8</td>
</tr>
<tr>
<td>26</td>
<td>3.7</td>
<td>2.3</td>
<td>2.2</td>
</tr>
<tr>
<td>27</td>
<td>5.1</td>
<td>3.3</td>
<td>3.3</td>
</tr>
<tr>
<td>28</td>
<td>3.8</td>
<td>3.0</td>
<td>2.9</td>
</tr>
<tr>
<td>29</td>
<td>3.5</td>
<td>2.4</td>
<td>2.3</td>
</tr>
<tr>
<td>30</td>
<td>4.0</td>
<td>3.0</td>
<td>2.9</td>
</tr>
<tr>
<td>31</td>
<td>7.6</td>
<td>6.1</td>
<td>6.0</td>
</tr>
<tr>
<td>32</td>
<td>6.6</td>
<td>5.1</td>
<td>5.0</td>
</tr>
<tr>
<td>33</td>
<td>5.3</td>
<td>4.1</td>
<td>4.0</td>
</tr>
<tr>
<td>34</td>
<td>6.6</td>
<td>3.9</td>
<td>3.8</td>
</tr>
<tr>
<td>35</td>
<td>5.0</td>
<td>4.0</td>
<td>3.9</td>
</tr>
</tbody>
</table>

Table 4.9: Percent Improvement in Best Integer Solution over Straightforward CPLEX Implementation
report the run time associated with finding the solution obtained from the decomposition procedure. This run time is computed as the maximum run time of all the origin scheduling subproblems decoupled by origin (because these problems can be run in parallel) added to the computation time for the problem of determining the train schedule outbound from the hub together with all container routing decisions once the train schedule outbound from the origin is fixed. Finally, we report the total run time computed by adding the run time for all the origin scheduling problems (i.e., corresponding to serial computation time of the various subproblems) to that of determining the train schedule outbound form the hub and all container routing decisions. It should be noted that in most problem instances, the maximum solution time for an origin scheduling subproblem far exceeds the mean solution time across all origins.

Many of the decoupled origin scheduling problems take a few minutes to solve to optimality. However, even for subproblem instances in which the computation time is higher (perhaps an hour or two), a solution which deviates only 2%-5% from the optimal is found in a matter of ten to thirty minutes. Hence, although it may require an hour or two of CPU time to prove optimality of a solution to some of the more difficult instances of the decoupled subproblems, a near-optimal solution is obtainable in a much shorter amount of time. Using a straightforward implementation of CPLEX on the original problem, either the solution improves incrementally throughout the run, or a feasible solution is found near the beginning of the run which remains unchanged until the computer reaches its memory limit. In either case, without the decomposition procedure, the quality of the solutions found by the end of the run are poor, i.e., they yield gaps between the best integer solution and the lower bound of about 20%-30%, as shown in Table 4.8.

Although there is a substantial reduction in solution time using the decomposition technique for the problem instances with three, and even, in some cases, four destinations, the solution time increases for the larger problem instances (i.e., those containing six destinations). For these instances, the aggregation technique described in Section 4.3 is used. There is a wide deviation in the amount of time necessary to find the optimal direct train schedules, which results in both a high parallel and a high serial processing time. As the number of origins increases, the number of origin scheduling subproblems increases, and hence the serial run time. Recall that a direct train schedule must be obtained for
<table>
<thead>
<tr>
<th>Problem</th>
<th>memory failure (min.) for original problem</th>
<th>decomposition run time: parallel (min.)</th>
<th>decomposition run time: serial (min.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>132</td>
<td>244</td>
<td>607</td>
</tr>
<tr>
<td>2</td>
<td>157</td>
<td>205</td>
<td>442</td>
</tr>
<tr>
<td>3</td>
<td>130</td>
<td>214</td>
<td>360</td>
</tr>
<tr>
<td>4</td>
<td>113</td>
<td>220</td>
<td>415</td>
</tr>
<tr>
<td>5</td>
<td>151</td>
<td>451</td>
<td>722</td>
</tr>
<tr>
<td>6</td>
<td>73</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>7</td>
<td>72</td>
<td>57</td>
<td>64</td>
</tr>
<tr>
<td>8</td>
<td>78</td>
<td>26</td>
<td>39</td>
</tr>
<tr>
<td>9</td>
<td>79</td>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>92</td>
<td>35</td>
<td>41</td>
</tr>
<tr>
<td>11</td>
<td>114</td>
<td>153</td>
<td>196</td>
</tr>
<tr>
<td>12</td>
<td>84</td>
<td>49</td>
<td>78</td>
</tr>
<tr>
<td>13</td>
<td>81</td>
<td>15</td>
<td>24</td>
</tr>
<tr>
<td>14</td>
<td>103</td>
<td>113</td>
<td>128</td>
</tr>
<tr>
<td>15</td>
<td>74</td>
<td>15</td>
<td>20</td>
</tr>
<tr>
<td>16</td>
<td>91</td>
<td>155</td>
<td>191</td>
</tr>
<tr>
<td>17</td>
<td>85</td>
<td>84</td>
<td>180</td>
</tr>
<tr>
<td>18</td>
<td>101</td>
<td>51</td>
<td>63</td>
</tr>
<tr>
<td>19</td>
<td>104</td>
<td>111</td>
<td>250</td>
</tr>
<tr>
<td>20</td>
<td>98</td>
<td>79</td>
<td>162</td>
</tr>
<tr>
<td>21</td>
<td>91</td>
<td>38</td>
<td>92</td>
</tr>
<tr>
<td>22</td>
<td>83</td>
<td>8</td>
<td>15</td>
</tr>
<tr>
<td>23</td>
<td>91</td>
<td>31</td>
<td>66</td>
</tr>
<tr>
<td>24</td>
<td>88</td>
<td>11</td>
<td>19</td>
</tr>
<tr>
<td>25</td>
<td>102</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>26</td>
<td>109</td>
<td>234</td>
<td>613</td>
</tr>
<tr>
<td>27</td>
<td>103</td>
<td>214</td>
<td>350</td>
</tr>
<tr>
<td>28</td>
<td>104</td>
<td>149</td>
<td>227</td>
</tr>
<tr>
<td>29</td>
<td>101</td>
<td>214</td>
<td>430</td>
</tr>
<tr>
<td>30</td>
<td>102</td>
<td>245</td>
<td>585</td>
</tr>
<tr>
<td>31</td>
<td>101</td>
<td>163</td>
<td>492</td>
</tr>
<tr>
<td>32</td>
<td>102</td>
<td>219</td>
<td>399</td>
</tr>
<tr>
<td>33</td>
<td>95</td>
<td>253</td>
<td>490</td>
</tr>
<tr>
<td>34</td>
<td>107</td>
<td>160</td>
<td>196</td>
</tr>
<tr>
<td>35</td>
<td>100</td>
<td>201</td>
<td>337</td>
</tr>
</tbody>
</table>

Table 4.10: Time Performance: Initial Implementation vs. Decomposition Procedure
each origin-destination pair before the indirect train schedule is determined for each origin scheduling problem. Once the direct train schedule is fixed, the problem of determining an indirect train schedule is solved relatively quickly (i.e., within a matter of seconds). For the problem instances with a larger number of destinations, the other significant contributor to the solution time is determining the train schedule outbound from the hub simultaneously with all of the container movements once the train schedule outbound from the origin is fixed. For these problems, the solution times are often on the order of two hours, at which point the computer runs out of memory without reaching a provably optimal solution. However, the deviation from optimality is less than 2% in all of these subproblems. Moreover, the best solution obtained throughout the run is often found within the first two or three minutes. Hence, these long solution times generally inflate the practical run time by one to two hours.

Because of these considerations, the solution times cannot be accurately portrayed nor compared with those of simply running the original problem with CPLEX until a memory limit is reached. A machine with less memory would obtain solutions as good as most of the ones we obtain, but in less time, simply because its memory limit would be reached sooner. Conversely, a machine with more memory may yield longer runs for the subproblems that cannot be solved to optimality with our hardware, but would certainly also result in a longer run time on the original problem. Practically speaking, solution times on the order of several hours are not unreasonable, though. The purpose of our model is to establish train schedules on a weekly basis, which could be done overnight.

We conclude that we are able to obtain better solutions than those obtained with a straightforward implementation of CPLEX through several variations of a decomposition technique based on a physical partitioning of the network. These variations take problem size, specifically, the number of destinations, into account, as well as the centralization of decision-making and availability of information.
Chapter 5

Lower Bounds

The previous chapter describes a method for finding better solutions using a new decomposition approach, which reduces the gap between the best integer solution and the lower bound from CPLEX. However, because the bounds used by CPLEX are based on linear programming relaxations, they are often loose. In our problem, integer variables correspond to sending a direct or an indirect train, and the costs associated with these actions are the fixed costs of running a train. Whenever the train variables are allowed to be continuous, the optimal linear programming solution will have partial trains and incur partial costs, rather than the full cost which would be assessed in the integer version of the problem. Because we believe that our problem structure results in particularly weak lower bounds, we develop several approaches to improve these bounds.

5.1 Relaxation Technique

Our first procedure for improving the lower bound is based on the simple idea of variable relaxation with respect to a restricted subset of variables. As mentioned earlier, we obtain solutions with integer values for the container variables without imposing integrality constraints on these variables. Hence, all our implementations, including the initial runs, are made allowing these variables to be continuous. However, all variables corresponding to sending direct and indirect trains must be restricted to be integer.

Consider the outcome if a continuous relaxation of the direct train variables were allowed. In this case, sending direct trains would always be preferable to indirect trains since they are faster, less costly, and would permit a partial setup cost to be incurred which is proportional to the fraction of space used on the train.
Now consider relaxing the integrality requirements on the indirect trains. This is not as loose a relaxation, because fractional indirect trains do not clearly dominate integral direct trains. Indirect trains are not only more costly, but they are also slower and, for some demands, not a feasible consideration.

Hence, were we to relax the integrality requirements on the variables corresponding to the direct trains, all containers would be sent directly and the algorithm would yield a lower bound as poor as the linear programming bounds given to us by CPLEX at the root node of the branch and bound tree. If, however, we relax the integrality requirement on the variables corresponding to indirect trains scheduled both into and out of the hub, this relaxed problem is one in which there is no clearly dominated or dominating train type. In fact, the resulting problem is easier to solve because it contains fewer integer variables. Therefore, CPLEX usually provides a tighter lower bound on the problem. Because this problem is a relaxed version of the original problem and has a tighter lower bound, the bound can be used as an improved lower bound on the original problem.

5.2 Adding Cuts

Recall that the lower bounds for our problem instances derived from CPLEX are based on fractional trains, causing an inaccurately low cost to be associated with sending trains. If we add constraints which force a minimum number of trains to be sent, the search space will be restricted and the lower bound will be tightened (although the lower bound is still derived, in part, from sending fractional trains). The method described below places restrictions both on the minimum number of trains sent from an origin and on the minimum number of trains arriving at a destination based on the cumulative number of containers that arrive throughout the planning horizon. The method is motivated by requirements of the physical setting from which the problem is derived.

We introduce the concept of a "super train." As mentioned previously, direct trains have cost and time savings advantages over indirect trains, while indirect trains have geographical consolidation advantage over direct trains. Consider a fictitious train type that possesses the advantages of both direct and indirect trains. This "super train" emanating from the origin can deliver shipments to any destination, giving it the geographical consolidation advantage of indirect trains. However, like direct trains, it incurs neither the costs nor time.
delays associated with passing through a hub. Travel time to any destination is the direct travel time for the origin-destination pair.

Figure 5.1 shows the capability of a super train traveling between an origin, Seattle, and the three destinations. Note that it does not have to pass through a hub (in other words, it incurs no time delay or costs there), yet is capable of delivering shipments to all three destinations.

We can use these super trains to provide two different lower bounds on the total number of trains that must be sent outbound from an origin over the time horizon. The lower bounds pertain to the cumulative number of trains scheduled during the horizon rather than to the number of trains on a particular day. The primary reason for this is the substitutability of trains across time to service non-urgent containers.

One lower bound is imposed on the number of trains outbound from the origin to a specific destination. Figure 5.2 illustrates the arcs, corresponding to trains, on which a lower bound constraint of this type is imposed.

The other lower bound is on the number of trains from a specific origin bound for all destinations. Figure 5.3 illustrates the arcs, corresponding to train routes, on which a lower bound constraint of this type is imposed.
Figure 5.2: Arcs Included in Lower Bound Bundle Constraints: Single Origin to Single Destination: 1

Figure 5.3: Arcs Included in Lower Bound Bundle Constraints: Single Origin to All Destinations
Figure 5.4: Arcs Included in Lower Bound Bundle Constraints: Single Origin to Single Destination: 2

Similarly, we can derive two more sets of lower bounds, in this case on the number of trains inbound to a specific destination from a single origin, or on the number of trains bound for a specific destination from all origins. Figures 5.4 and 5.5 illustrate the sets of arcs on which we impose lower bounds on trains inbound to the destination from a specific origin, and from all origins, respectively.

To obtain lower bounds on the number of trains into the destination, we reverse our network such that the containers arrive at the destination at the time they would be arriving at the origin, and are demanded at the origin at the time they would be demanded at the destination. The fewest direct and indirect trains that would be required to send these containers in the reverse direction would be the fewest trains that could enter the destination while maintaining feasibility of the container flows on our network in the forward direction.

The four types of cuts, or valid inequalities, that are added to our formulation are described below:

Let $M_{ik}$ be the lower bound associated with the origin-destination pair ($i,k$). Then the constraints "Lower Bound: Single Origin to Single Destination: 1" ($sosd1$) on the minimum number of trains needed to service demand between a single origin-destination pair are given as:
Figure 5.5: Arcs Included in Lower Bound Bundle Constraints: All Origins to Single Destination

\[ \sum_{t=1}^{T} z_{ikt} + \sum_{t=1}^{T} \sum_{j} z_{ijt} \geq M_{ik} \quad \forall i, k \quad (sosd1) \]

Let \( M_i \) be the lower bound associated with an origin \( i \). Then, considering demand bound for any destination, the constraints “Lower Bound: Single Origin to All Destinations” \((soad)\) are given as follows:

\[ \sum_{t=1}^{T} \sum_{k} z_{ikt} + \sum_{t=1}^{T} \sum_{j} z_{ijt} \geq M_i \quad \forall i \quad (soad) \]

The super train problem on the reversed network considering demand from a single origin generates constraints “Lower Bound: Single Origin to Single Destination: 2” \((sosd2)\):

\[ \sum_{t=1}^{T} z_{ikt} + \sum_{t=1}^{T} \sum_{j} z_{jkt} \geq M_{ik} \quad \forall i, k \quad (sosd2) \]
Similarly, letting $M_k$ be the lower bound associated with a destination $k$, the super train problem on the reversed network considering demand from all origins generates the constraints "Lower Bound: All Origins to Single Destination" (aosd), which are given as follows:

$$\sum_{t=1}^{T} \sum_{i} z_{ikt} + \sum_{t=1}^{T} \sum_{j} z_{jkt} \geq M_k \quad \forall k \quad (aosd)$$

To derive each of these lower bounds on the number of trains, we solve the problem of minimizing the number of super trains rather than minimizing the total fixed and variable transportation, handling, and holding costs, and the decision variables in the problem are now the container shipments and the super train schedule.

In constructing this super train problem, we first establish a fixed direct train schedule in which all train variables are set equal to zero except in those instances in which they must accommodate containers that arrive at an origin with no slack in their schedule. The essence of the super train problem is then to find a schedule that minimizes the total number of super trains outbound from the origin to a "super hub" that services all destinations, given the fixed direct train schedule, while satisfying on-time delivery requirements. The decision variables in the problem are the container shipments and the super train schedule.

To make a translation from this problem containing both fixed direct trains and super trains, to a simpler representation in which all containers pass through some type of hub, we can create dummy hubs for each origin-destination pair, through which all containers traveling on fixed direct trains pass, and a single aggregate "super hub" through which the remaining containers pass. We can use an adjusted due date for each customer order, given as the latest time a container may reach a hub (i.e., a dummy hub, or the super hub) and still arrive at its destination on time. Because neither train type experiences delay at the hub, the adjusted due date is merely the original due date at the destination less the transit time between the hub and the destination.

In an optimal solution to this problem, containers requiring expedited service are sent on direct trains. If such a train is not full, other eligible containers with slack in the schedule may also be shipped on these trains. The remaining containers travel on super trains. With the direct train schedule fixed, the problem reduces to one of finding a lower bound
on the number of trains to service the remaining container traffic, which is derived from the super trains. This is clearly a relaxation of the original problem, and thus provides a lower bound on the number of required trains. First of all, the objective is to minimize the number of trains, whereas an optimal solution to the formulation in which the objective is to minimize cost may include more trains. Second, the trains incur no delay at the hub, although in reality any indirect trains scheduled would incur a delay. Finally, it is assumed that the containers sent on the super trains will arrive on time at the destination if they arrive “on time” at the hub. However, depending upon the schedule outbound from the hub, “perfect” connections may not be attainable in the original version of the problem, which would necessitate scheduling more trains outbound from the origin.

We can formulate weak upper bounds on the number of trains sent from each origin, and the number of trains arriving at each destination over the entire time horizon. These values are derived from the number of trains required to accommodate containers assuming that these shipments all arrive with no slack time in their schedules, i.e., they must be shipped immediately. We impose these constraints in the problem along with the lower bound constraints described above in order to restrict the solution space. However, the upper bound constraints affect the bound on the objective function in only a minor way. Clearly, we cannot impose a non-zero lower bound on the number of trains either inbound to or outbound from the hub, as an optimal solution could consist of sending no indirect trains whatsoever, depending on the mix of container traffic.

Table 5.1 shows the gap from the original CPLEX implementation $\frac{\text{out}_{l,b} - CPLEX_{l,b}}{CPLEX_{b,i} - CPLEX_{l,b}}$, and the reduction in the gap for each of the thirty-five problem instances when we simultaneously implement both the relaxation and cut addition procedures described above, along with the weak upper bounds on the number of trains. The reduction is calculated as follows:

$$\frac{\text{out}_{l,b} - CPLEX_{l,b}}{CPLEX_{b,i} - CPLEX_{l,b}}$$

The reductions in the gap are significant, i.e., between 40% and 50% on average. Greater reductions in the gap are realized for those problem instances in which a higher proportion of the containers require direct train service. The “necessary direct train” component of the lower bound is a tight bound, whereas the super trains provide a looser contribution. Specifically, test problems 6 through 10, 16 through 18, 21 through 23, and 26 through 30 have the highest demand for expedited service and an average reduction in the gap of
64.3%. Problem instances 11 through 15, 19 through 20, 24 through 25, and 31 through 35, have the next highest demand for expedited service and a corresponding average reduction in the gap of 36.9%. The first five problem instances have the lowest demand for expedited service, and an average gap reduction of 26.4%. Additionally, we observe that the reduction in the gap is independent of problem size. Hence, even for the largest problem instances, significant benefits are realized.

5.3 Relative Contributions: Tightening the Lower Bounds

Of the two approaches for tightening the lower bounds, the addition of cuts provides the majority of the contribution. However, because, these two methods for tightening the lower bound on our problem instances were developed independently of one another, we believed that they would complement each other well. That is, for instances in which relaxing the integrality requirements on the variables corresponding to the indirect trains would not provide a significant improvement in the lower bound, adding the cuts described above might. Similarly, for instances in which the added cuts would provide only a weak lower bound, relaxing the integrality requirements on the indirect train variables would have a significant impact on the improvement of the lower bound.

Our logic was as follows: Consider problem instances in which for many (perhaps a third to half of) origin-destination-time period combinations, a direct train must be sent to meet level of service requirements. These are problem instances in which the majority of trains are direct in an optimal solution. The greater is the representation of direct trains in the lower bound constraint, the tighter the cut will be. On the other hand, consider the result on such a problem instance from relaxing the integrality requirement on the indirect trains. Because a greater proportion of trains must be sent directly, allowing the integer program to "cheat" on incurring the cost of running an indirect train provides little benefit in terms of tightening the lower bound since relatively few indirect trains are sent in the optimal solution. As the number of direct trains that must be sent increases, more traffic is assigned to these trains. Even containers with slack time in their schedule, and which would otherwise be sent indirectly, are more likely to travel on direct trains, given that there is available capacity.

If the requirements of arrivals are such that fewer direct trains must be sent, the optimal
<table>
<thead>
<tr>
<th>Problem</th>
<th>initial CPLEX gap (%)</th>
<th>reduction in gap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>34.8</td>
<td>32.8</td>
</tr>
<tr>
<td>2</td>
<td>28.4</td>
<td>36.7</td>
</tr>
<tr>
<td>3</td>
<td>38.0</td>
<td>26.2</td>
</tr>
<tr>
<td>4</td>
<td>14.2</td>
<td>13.8</td>
</tr>
<tr>
<td>5</td>
<td>31.2</td>
<td>22.8</td>
</tr>
<tr>
<td>6</td>
<td>27.6</td>
<td>84.6</td>
</tr>
<tr>
<td>7</td>
<td>28.5</td>
<td>78.6</td>
</tr>
<tr>
<td>8</td>
<td>24.9</td>
<td>54.8</td>
</tr>
<tr>
<td>9</td>
<td>24.7</td>
<td>90.0</td>
</tr>
<tr>
<td>10</td>
<td>23.0</td>
<td>54.0</td>
</tr>
<tr>
<td>11</td>
<td>20.5</td>
<td>39.7</td>
</tr>
<tr>
<td>12</td>
<td>22.9</td>
<td>48.3</td>
</tr>
<tr>
<td>13</td>
<td>20.4</td>
<td>34.1</td>
</tr>
<tr>
<td>14</td>
<td>20.5</td>
<td>39.1</td>
</tr>
<tr>
<td>15</td>
<td>26.3</td>
<td>45.6</td>
</tr>
<tr>
<td>16</td>
<td>25.1</td>
<td>58.6</td>
</tr>
<tr>
<td>17</td>
<td>28.3</td>
<td>71.0</td>
</tr>
<tr>
<td>18</td>
<td>24.0</td>
<td>58.7</td>
</tr>
<tr>
<td>19</td>
<td>30.4</td>
<td>32.6</td>
</tr>
<tr>
<td>20</td>
<td>24.2</td>
<td>32.3</td>
</tr>
<tr>
<td>21</td>
<td>27.0</td>
<td>78.1</td>
</tr>
<tr>
<td>22</td>
<td>29.7</td>
<td>65.9</td>
</tr>
<tr>
<td>23</td>
<td>24.3</td>
<td>57.7</td>
</tr>
<tr>
<td>24</td>
<td>25.6</td>
<td>51.7</td>
</tr>
<tr>
<td>25</td>
<td>24.9</td>
<td>39.5</td>
</tr>
<tr>
<td>26</td>
<td>29.4</td>
<td>52.3</td>
</tr>
<tr>
<td>27</td>
<td>31.0</td>
<td>53.3</td>
</tr>
<tr>
<td>28</td>
<td>27.0</td>
<td>51.8</td>
</tr>
<tr>
<td>29</td>
<td>28.3</td>
<td>59.6</td>
</tr>
<tr>
<td>30</td>
<td>30.7</td>
<td>59.5</td>
</tr>
<tr>
<td>31</td>
<td>29.0</td>
<td>30.6</td>
</tr>
<tr>
<td>32</td>
<td>27.7</td>
<td>31.2</td>
</tr>
<tr>
<td>33</td>
<td>29.5</td>
<td>36.1</td>
</tr>
<tr>
<td>34</td>
<td>25.7</td>
<td>22.1</td>
</tr>
<tr>
<td>35</td>
<td>25.9</td>
<td>33.3</td>
</tr>
</tbody>
</table>

Table 5.1: Performance Improvement in Gap via Tightening the Lower Bound
schedule tends to consist of more indirect trains. The cuts are not as tight, since a greater portion of the contribution is derived from the weaker super train cover. However, it was our hope that allowing the integer program to “cheat” on incurring costs of running indirect trains would result in an easier problem to solve, because scheduling indirect trains is preferable in this instance. As a result, CPLEX would produce a tighter lower bound with the variable relaxation technique. In fact, in an earlier version of CPLEX (version 4.0), this was the case, and we obtained significant (10%-20%) improvements in the lower bounds by implementing this technique.

Table 5.2 reports the relative contributions of the variable relaxation and the cut generation to the reduction in the lower bound for our first fifteen problem instances with respect to the best integer solution via a straightforward implementation of CPLEX (5.0). The other twenty problem instances would exhibit similar qualitative behavior, namely, that the contribution from cut addition far outweighs that from the variable relaxation. The rationale for this is as follows: Variable relaxation helps to create an easier problem, i.e., one that contains fewer integer variables, thereby yielding a tighter lower bound. Problem instances sixteen through thirty-five are larger problems and therefore contain more integer variables. Hence, relaxing the same relative proportion of integer variables would still yield a problem with more integer variables than in the first fifteen instances, hence a “harder” problem. Therefore, variable relaxation is unlikely to result in a substantially easier problem, nor a correspondingly tighter lower bound to the original problem.

Note that in some instances the variable relaxation has a greater effect when coupled with the cut addition (cf. Table 5.1) than it does independently. Despite the fact that the relative contribution of variable relaxation using the newer version of CPLEX is fairly insignificant in our test problems, the contribution of this approach may be greater in problem instances with a different cost structure, e.g., ones in which indirect trains are much more costly to run than direct trains, traffic levels are generally low, and/or relatively few containers require expedited service.
<table>
<thead>
<tr>
<th>Problem</th>
<th>reduction in gap from cut addition (%)</th>
<th>reduction in gap from integrality relaxation (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>28.9</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>34.1</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>25.3</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>12.8</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>19.8</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>82.3</td>
<td>1.6</td>
</tr>
<tr>
<td>7</td>
<td>76.4</td>
<td>0.5</td>
</tr>
<tr>
<td>8</td>
<td>54.1</td>
<td>0.9</td>
</tr>
<tr>
<td>9</td>
<td>90.4</td>
<td>2.6</td>
</tr>
<tr>
<td>10</td>
<td>54.5</td>
<td>1.9</td>
</tr>
<tr>
<td>11</td>
<td>39.2</td>
<td>2.5</td>
</tr>
<tr>
<td>12</td>
<td>47.8</td>
<td>-</td>
</tr>
<tr>
<td>13</td>
<td>33.4</td>
<td>-</td>
</tr>
<tr>
<td>14</td>
<td>39.0</td>
<td>-</td>
</tr>
<tr>
<td>15</td>
<td>45.1</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 5.2: Relative Contributions to Improvement in the Lower Bound

5.4 Summary of Results and Gap Estimation for "Hard" Problems

We now combine our lower bound results with our best integer solution results described in the previous chapter. Table 5.3 gives a comparison of the gaps between the best integer solution and the lower bound for our thirty-five test problems. The second column reports the gap obtained via a straightforward implementation of CPLEX at the time when the computer reached its memory limit (i.e., $\frac{\text{CPLEX}_{b}}{\text{CPLEX}_{lb}} - 1$). The third column reports the gap after implementing both our decomposition procedure to find better integer solutions and our two methods to tighten the lower bound (i.e., $\frac{\text{our}_{b}}{\text{our}_{lb}} - 1$). In all problem instances but two, we are able to reduce the gap by over half, and most instances have a final gap well under 10%.

From Table 5.3 we note that there are a few problem instances in which gaps between the best integer solution obtained via decomposition and the tightened lower bound are between 10% and 15%. For such instances, we would like to better approximate the gap between the best integer solution and the lower bound.

In Chapter 4, we discussed a cause for suboptimality of the best integer solution from our decomposition procedure. While solving the origin scheduling problem, it is not possible
<table>
<thead>
<tr>
<th>Problem</th>
<th>initial CPLEX gap (%)</th>
<th>final gap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>34.8</td>
<td>15.4</td>
</tr>
<tr>
<td>2</td>
<td>28.4</td>
<td>12.7</td>
</tr>
<tr>
<td>3</td>
<td>38.0</td>
<td>17.6</td>
</tr>
<tr>
<td>4</td>
<td>14.2</td>
<td>8.3</td>
</tr>
<tr>
<td>5</td>
<td>31.2</td>
<td>17.6</td>
</tr>
<tr>
<td>6</td>
<td>27.6</td>
<td>1.9</td>
</tr>
<tr>
<td>7</td>
<td>28.5</td>
<td>2.7</td>
</tr>
<tr>
<td>8</td>
<td>24.9</td>
<td>4.1</td>
</tr>
<tr>
<td>9</td>
<td>24.7</td>
<td>1.5</td>
</tr>
<tr>
<td>10</td>
<td>23.0</td>
<td>3.3</td>
</tr>
<tr>
<td>11</td>
<td>20.5</td>
<td>8.3</td>
</tr>
<tr>
<td>12</td>
<td>22.9</td>
<td>10.9</td>
</tr>
<tr>
<td>13</td>
<td>20.4</td>
<td>7.7</td>
</tr>
<tr>
<td>14</td>
<td>20.5</td>
<td>8.8</td>
</tr>
<tr>
<td>15</td>
<td>26.3</td>
<td>6.8</td>
</tr>
<tr>
<td>16</td>
<td>25.1</td>
<td>5.5</td>
</tr>
<tr>
<td>17</td>
<td>28.3</td>
<td>4.4</td>
</tr>
<tr>
<td>18</td>
<td>24.0</td>
<td>5.6</td>
</tr>
<tr>
<td>19</td>
<td>30.4</td>
<td>13.4</td>
</tr>
<tr>
<td>20</td>
<td>24.2</td>
<td>10.2</td>
</tr>
<tr>
<td>21</td>
<td>27.0</td>
<td>3.5</td>
</tr>
<tr>
<td>22</td>
<td>28.7</td>
<td>3.8</td>
</tr>
<tr>
<td>23</td>
<td>24.3</td>
<td>4.5</td>
</tr>
<tr>
<td>24</td>
<td>25.6</td>
<td>7.2</td>
</tr>
<tr>
<td>25</td>
<td>24.9</td>
<td>8.2</td>
</tr>
<tr>
<td>26</td>
<td>29.4</td>
<td>8.0</td>
</tr>
<tr>
<td>27</td>
<td>31.0</td>
<td>6.7</td>
</tr>
<tr>
<td>28</td>
<td>27.0</td>
<td>7.2</td>
</tr>
<tr>
<td>29</td>
<td>28.3</td>
<td>5.9</td>
</tr>
<tr>
<td>30</td>
<td>30.7</td>
<td>6.1</td>
</tr>
<tr>
<td>31</td>
<td>29.0</td>
<td>9.5</td>
</tr>
<tr>
<td>32</td>
<td>27.7</td>
<td>9.8</td>
</tr>
<tr>
<td>33</td>
<td>29.5</td>
<td>10.9</td>
</tr>
<tr>
<td>34</td>
<td>25.7</td>
<td>11.1</td>
</tr>
<tr>
<td>35</td>
<td>25.9</td>
<td>10.1</td>
</tr>
</tbody>
</table>

Table 5.3: Comparison of Initial Performance and the Gap after Implementing our Decomposition and Lower Bound Improving Procedures
<table>
<thead>
<tr>
<th>Problem</th>
<th>remaining gap (%)</th>
<th>gap estimate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15.4</td>
<td>2.5</td>
</tr>
<tr>
<td>2</td>
<td>12.7</td>
<td>2.6</td>
</tr>
<tr>
<td>3</td>
<td>17.6</td>
<td>1.2</td>
</tr>
<tr>
<td>5</td>
<td>17.6</td>
<td>3.0</td>
</tr>
<tr>
<td>12</td>
<td>10.9</td>
<td>1.2</td>
</tr>
<tr>
<td>19</td>
<td>13.4</td>
<td>1.5</td>
</tr>
<tr>
<td>20</td>
<td>10.2</td>
<td>3.3</td>
</tr>
<tr>
<td>33</td>
<td>10.9</td>
<td>0.3</td>
</tr>
<tr>
<td>34</td>
<td>11.1</td>
<td>0.0</td>
</tr>
<tr>
<td>35</td>
<td>10.1</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Table 5.4: Gap Estimates for “Hard” Problems

to provide the decoupled subproblems with all the information necessary to construct a solution that will result in an optimal schedule outbound from the hub. This results in containers arriving at the hub with less slack in their schedules, and hence less opportunity to consolidate shipments outbound from the hub. Translated into a cost contribution to the objective function, this corresponds to extra setup cost incurred at the hub. Let us assume that we (reasonably) expect to send one train outbound to a destination for each train sent inbound to the hub. The positive difference between the number of trains outbound from the hub and inbound to the hub is the number of “excess” trains, potentially caused by lack of foresight at the origin. Assume we ignore holding costs, take transportation and handling costs as fixed, and weight the setup cost at the hub by demand at a given destination. Then if we multiply the number corresponding to the train imbalance at the hub by the setup cost of a train at the hub, we generate an estimate of the true gap, i.e., the loss of optimality due to the myopic nature of the hub scheduling problem. Of course, we cannot guarantee that the excess number of trains departing from the hub is the sole cause for the nonoptimality of our solution. However, such an estimate serves as a benchmark, and a more precise analysis would be extremely difficult.

Table 5.4 shows gap estimates for problem instances in which the gap exceeds 10% after implementing the decomposition procedure and using both the relaxation and cut addition procedure to tighten the lower bound. We note that the estimates for the true gap are much lower in all instances, in general, about 2%.
Chapter 6

Simple Heuristics

We devise two heuristics for providing fast and rough solutions to our train routing and container scheduling problem. The purpose for this is three-fold. First of all, we would like to test our conjecture that even a sensible implementation of a scheduling procedure via simple priority rules yields solutions which are far from optimal. Secondly, the results serve as a benchmark for those instances in which we have moderate gaps (between 10% and 20%) even after implementing our techniques for improving the best integer solution and tightening the lower bound. In other words, even if we cannot prove that we are within a few percentage points of the optimal solution, we would like to demonstrate that the solution is significantly better than a solution obtained via sensible, but not necessarily optimization, techniques. Finally, the heuristics we propose are characteristic of current operating plans and, as such, illustrate the cost and service efficiency benefits that could be realized were our solution procedures to be implemented.

Many current intermodal operations are based on running a number of indirect trains each day. Since we do not allow for any late shipments, we require that direct trains be sent when the due date of a container requires it. In practice, some shipments are simply sent late, rather than on a direct train. However, this would be infeasible for our formulation and, as such, would make a fair comparison difficult. For this reason, we require on-time delivery in the simple heuristics as well.

6.1 Heuristic 1:

The first heuristic is as follows: Send a direct train between each origin-destination pair at each time period the due date of a container requires it. Containers not requiring
expedited service are placed on these trains after the containers requiring urgent service have been given priority. For the remaining containers not shipped directly, for each day on which containers arrive at the origin, send as many indirect trains from the origin as necessary to ship all remaining containers. Send indirect trains from the hub to each destination each day to ship the containers arriving at the hub. Because trains are sent from the origin each day, for each origin-destination-arrival date at origin combination, it is only necessary to differentiate between container departures from the origin based on whether the container requires expedited service or not. Similarly, because all demand is serviced at the hub each day, containers may be placed on the trains in any order.

Using the notation introduced in Section 2.1, let us state Heuristic 1 mathematically as follows:

First, for each origin-destination pair, compute the number of direct trains that must be sent in each period:

\[
\text{Set } z_{ikt} = \left\lfloor \frac{\sum_{l=t+\alpha_{ik}}^{t+\beta_{ij}+\gamma_{jk}+\delta_{j}-1} b_{ikl}}{C} \right\rfloor \quad \forall i, k, t
\]

The corresponding number of containers that are sent directly is given as follows:

\[
x_{ikt} = b_{ikl} \quad \forall i, k, t, t + \alpha_{ik} \leq l < \beta_{ij} + \gamma_{jk} + \delta_{j}
\]

\[
x_{ikt} = b'_{ikl} \quad \forall i, k, t, l \geq t + \beta_{ij} + \gamma_{jk} + \delta_{j}
\]

where \(b'_{ikl} \leq b_{ikl} \quad \forall i, k, t, l \geq t + \beta_{ij} + \gamma_{jk} + \delta_{j}\) is a subset of containers not requiring expedited service whose total quantity fills the direct trains as much as possible.

Then, for each origin and time period, compute the number of indirect trains needed to service the remaining containers not sent on a direct train:

\[
\text{Set } z_{ijt} = \left\lfloor \frac{\left\{ \sum_k \sum_{l=t+\beta_{ij}+\gamma_{jk}+\delta_{j}}^{t+\beta_{ij}+\gamma_{jk}+\delta_{j}-1} b_{ikl} - (C \cdot z_{ikt} - \sum_{l=t+\alpha_{ik}}^{t+\beta_{ij}+\gamma_{jk}+\delta_{j}-1} b_{ikl}) \right\}}{C} \right\rfloor + 1 \quad \forall i, j, t
\]

For the case of a single hub which we treat in this dissertation, the corresponding number of containers sent indirectly to hub \(j\) is given as follows:

\[
x_{ijkt} = b_{ikl} - b'_{ikl} \quad \forall i, j = 1, k, t, l \geq t + \beta_{ij} + \gamma_{jk} + \delta_{j}
\]
Compute the number of containers ready to depart the hub for destination $k$ at time $t + \beta_{ij} + \delta_j$ as follows:

$$\sum_l u_{ijk(t+\beta_{ij}+\delta_j)}l = \left\{ \sum_{l \geq t+\beta_{ij}+\gamma_{jk}+\delta_j} b_{iktl} - (C * z_{ikt} - \sum_{l=t+\alpha_{ih}} b_{iktl}) \right\}^+ \quad \forall i, j, k, t$$

Finally, for each destination and time period, compute the number of trains outbound from the hub to accommodate the number of containers ready to depart from the hub:

$$\text{Set} \quad z_{jk(t+\beta_{ij}+\delta_j)} = \left[ \frac{\sum_i \sum_{l \geq t+\beta_{ij}+\gamma_{jk}+\delta_j} u_{ijk(t+\beta_{ij}+\delta_j)}l}{C} \right] \quad \forall j, k, t$$

### 6.2 Heuristic 2:

When traffic levels between certain origin-destination pairs are high enough, it may be more cost advantageous to ship containers directly, even if they do not require direct service. In this way, extra setup and handling costs associated with passing through the hub are avoided. Based on this observation, our second heuristic is a modification of our first in that we allow direct trains to be sent in all of the following scenarios: (i) when they are carrying any number of expedited containers, (ii) when they are full, regardless of whether the containers require expedited service, and (iii) when at most one train not carrying expedited containers is at least $\theta$ full, $0 \leq \theta \leq 1$. In practice, $\theta$ is determined based on managerial insight taking into account the relative values of fixed and variable costs incurred at the origin and at the hub, and typical traffic patterns. For congested routes, a higher value of $\theta$ may be used, whereas on routes for which customers pay premium rates for dedicated service, a smaller value may be allowed. Again, the only relevant differentiation in due dates that must be made when computing containers departing from the origin is whether they must be sent directly or not. Similarly, because all demand is serviced at the hub each day, containers may be placed on the trains in any order.

Mathematically, we state Heuristic 2 as follows:

$$\text{Set} \quad z_{ikt} = \left[ \frac{\sum_{l=t+\alpha_{ih}} b_{iktl}}{C} \right] + y \quad \forall i, k, t$$

where $y$ represents the number of additional direct trains to be sent other than those carrying containers requiring expedited service, and is the smallest nonnegative integer such that:
\[
\{ \sum_{l \geq \beta_{ij} + \gamma_{jk} + \delta_j} b_{ikt} - (C \cdot z_{ikt} - \sum_{l = t + \alpha_{ik}} b_{ikt}) \}^+ > \theta \cdot C + (y - 1) \cdot C \quad y \in \{0, Z^+\}
\]

The corresponding number of containers that are sent directly is given as follows:

\[x_{ikt} = b_{ikt} \quad \forall i, k, t, t + \alpha_{ik} \leq l < \beta_{ij} + \gamma_{jk} + \delta_j\]

\[x_{ikt} = b'_{ikt} \quad \forall i, k, t, l \geq t + \beta_{ij} + \gamma_{jk} + \delta_j\]

where \(b'_{ikt} \leq b_{ikt} \quad \forall i, k, t, l \geq t + \beta_{ij} + \gamma_{jk} + \delta_j\) is a subset of containers not requiring expedited service chosen for direct shipment because either (i) they fill a direct train (with or without expedited containers), or (ii) they constitute at most one train not carrying any expedited containers, but which is at least \(\theta\) full.

Then, for each origin and time period, compute the number of indirect trains needed to service remaining containers not sent on a direct train:

\[\text{Set } z_{ijt} = \left[ \sum_{k} \left\{ \sum_{l \geq t + \beta_{ij} + \gamma_{jk} + \delta_j} b_{ikt} - (C \cdot z_{ikt} - \sum_{l = t + \alpha_{ik}} b_{ikt}) \right\}^+ \right] / C \quad \forall i, j, t\]

For the case of a single hub, the corresponding number of containers sent indirectly to hub \(j\) is given as follows:

\[x_{ijkt} = b_{ikt} - b'_{ikt} \quad \forall i, j = 1, k, t, l \geq t + \beta_{ij} + \gamma_{jk} + \delta_j\]

Compute the number of containers ready to depart the hub for destination \(k\) at time \(t + \beta_{ij} + \delta_j\) as follows:

\[\sum_{l} u_{ijk(t + \beta_{ij} + \delta_j)l} = \left\{ \sum_{l \geq t + \beta_{ij} + \gamma_{jk} + \delta_j} b_{ikt} - (C \cdot z_{ikt} - \sum_{l = t + \alpha_{ik}} b_{ikt}) \right\}^+ \quad \forall i, j, k, t\]

Finally, for each destination and time period, compute the number of trains outbound from the hub to accommodate the number of containers ready to depart from the hub:

\[\text{Set } z_{jk(t + \beta_{ij} + \delta_j)} = \left[ \sum_{l} \sum_{l \geq t + \beta_{ij} + \gamma_{jk} + \delta_j} u_{ijk(t + \beta_{ij} + \delta_j)l} \right] / C \quad \forall j, k, t\]
6.3 Heuristics: Numerical Results

Table 6.1 presents the performance of the two heuristics expressed as the percent deviation above the best integer solution found by our decomposition technique. In Heuristic 2, we use a value of 0.65 for $\theta$ for all problem instances. The two heuristics perform similarly (i.e., within a percentage point or two). In general, the heuristics provide solutions about 15% above those found via our decomposition technique, although the performance is better for the larger problem instances, in which the simple heuristic solutions are only about 8% above those found via our specialized procedure. The results demonstrate both that simple approaches to our problem provide poor solutions, and that significant cost savings are to be realized from our proposed procedures over current operational procedures.
<table>
<thead>
<tr>
<th>Problem</th>
<th>Heuristic 1</th>
<th>Heuristic 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15.7</td>
<td>16.8</td>
</tr>
<tr>
<td>2</td>
<td>12.6</td>
<td>13.6</td>
</tr>
<tr>
<td>3</td>
<td>4.8</td>
<td>6.0</td>
</tr>
<tr>
<td>4</td>
<td>23.0</td>
<td>23.0</td>
</tr>
<tr>
<td>5</td>
<td>10.8</td>
<td>11.2</td>
</tr>
<tr>
<td>6</td>
<td>12.4</td>
<td>12.4</td>
</tr>
<tr>
<td>7</td>
<td>12.0</td>
<td>11.7</td>
</tr>
<tr>
<td>8</td>
<td>14.4</td>
<td>13.9</td>
</tr>
<tr>
<td>9</td>
<td>16.9</td>
<td>16.6</td>
</tr>
<tr>
<td>10</td>
<td>15.0</td>
<td>15.3</td>
</tr>
<tr>
<td>11</td>
<td>15.1</td>
<td>14.3</td>
</tr>
<tr>
<td>12</td>
<td>15.6</td>
<td>14.3</td>
</tr>
<tr>
<td>13</td>
<td>13.5</td>
<td>12.4</td>
</tr>
<tr>
<td>14</td>
<td>18.8</td>
<td>19.1</td>
</tr>
<tr>
<td>15</td>
<td>18.1</td>
<td>18.5</td>
</tr>
<tr>
<td>16</td>
<td>9.8</td>
<td>9.6</td>
</tr>
<tr>
<td>17</td>
<td>12.1</td>
<td>12.0</td>
</tr>
<tr>
<td>18</td>
<td>14.9</td>
<td>14.2</td>
</tr>
<tr>
<td>19</td>
<td>13.1</td>
<td>12.6</td>
</tr>
<tr>
<td>20</td>
<td>18.2</td>
<td>17.2</td>
</tr>
<tr>
<td>21</td>
<td>11.3</td>
<td>11.1</td>
</tr>
<tr>
<td>22</td>
<td>17.5</td>
<td>17.5</td>
</tr>
<tr>
<td>23</td>
<td>11.5</td>
<td>11.4</td>
</tr>
<tr>
<td>24</td>
<td>19.0</td>
<td>18.7</td>
</tr>
<tr>
<td>25</td>
<td>24.0</td>
<td>22.3</td>
</tr>
<tr>
<td>26</td>
<td>6.5</td>
<td>6.6</td>
</tr>
<tr>
<td>27</td>
<td>8.0</td>
<td>7.9</td>
</tr>
<tr>
<td>28</td>
<td>9.3</td>
<td>9.3</td>
</tr>
<tr>
<td>29</td>
<td>9.4</td>
<td>9.4</td>
</tr>
<tr>
<td>30</td>
<td>7.3</td>
<td>7.4</td>
</tr>
<tr>
<td>31</td>
<td>7.0</td>
<td>8.3</td>
</tr>
<tr>
<td>32</td>
<td>8.2</td>
<td>9.9</td>
</tr>
<tr>
<td>33</td>
<td>6.9</td>
<td>8.5</td>
</tr>
<tr>
<td>34</td>
<td>9.9</td>
<td>10.9</td>
</tr>
<tr>
<td>35</td>
<td>8.9</td>
<td>9.4</td>
</tr>
</tbody>
</table>

Table 6.1: Heuristics- Percent Above our Best Integer Solutions
Chapter 7

Conclusions and Future Research

In this dissertation, we have developed a procedure to provide near-optimal solutions to the problem of simultaneously determining train scheduling and container routing decisions in a rail intermodal setting within a reasonable amount of time. The contributions from this work are three-fold. First of all, we have developed a formulation for this setting that takes advantage of the embedded network structure. Secondly, we have developed a solution procedure that yields near-optimal solutions both for small problems, and for problems containing many destinations. Finally, we have created methods for tightening the lower bounds to improve the guarantee that our solutions are indeed near-optimal.

There are several immediate extensions of this work beyond the dissertation. First, we would like to apply the procedures developed here to a real-life setting and analyze the outcome. To this end, cost and demand data would have to be collected and analyzed. Parameters in the model may need to be modified to be consistent with data availability. Additionally, the model may need to be tailored to take extra considerations into account, which are discussed below.

The model presented here can be thought of as a "skeleton" model. That is, basic constraints on levels of service and train capacity are taken into consideration. However, other requirements are not enforced. These may be constraints on capacity at a hub, or on the feasibility of early container arrival at the destination. Another modification includes allowing for a small violation of the train capacity constraints. These aspects of the problem may need to be included in certain real-life applications. We would like to consider the effect of such factors on the solvability of our model. Additionally, we would also like to develop methodology to obtain good solutions for the problem with multiple hubs. These hubs may
have a parallel configuration, such that shipments may pass through one of several hubs, or they may be positioned in serial, such that containers may be shipped through multiple hubs en route to their destination.

Our model does not take limited locomotive capacity into consideration, which may result in constraints on the maximum number of trains that can be scheduled on a given route per day. This modification could be incorporated indirectly by placing constraints on the number of trains scheduled on a given route as a consequence of available tractive power. Alternatively, the model could be extended to incorporate the flow of locomotives. The structure of the model would be more complicated, and the solution and lower bound procedures would have to be modified to include the location of locomotives at a particular point in time, as well as logical constraints ensuring that a train of a given length could only be sent if the necessary number of locomotives were available.

We require that shipments be delivered to their destination on time, or early. In a variation of this model, we could allow for late shipments with a penalty. The basic methodology we have developed for finding a good solution would be valid, with some minor bookkeeping modifications to the origin and hub scheduling formulations. However, since the multicommodity flow problem is always feasible given that containers can be shipped arbitrarily late or not at all, there is less feedback in the search strategy used to obtain a solution to these problems. This may increase the difficulty in obtaining solutions to the subproblems. Additionally, we anticipate that the lower bounds will not be as tight. Because direct trains would never have to be sent, the lower bounds would be derived solely from super trains which do not provide as tight a bound on the number of trains required to service all the demand.

Additionally, since we are solving a problem over a time horizon of approximately a week, we will most likely encounter boundary effects. We should include provisions to help mitigate these effects when implementing the model in a real setting. To this end, we might duplicate the network over several, identical horizons. In this way, the second or third “copy” of the network would use values of container inventories from a previous horizon as starting values for the following horizon.

Furthermore, all our traffic movements are currently in one direction. It can be argued that it is more important to solve our problem in the “bottleneck direction” and send
equipment (both rail cars and containers) in the reverse direction empty if necessary. In fact, many rail companies face an imbalance in their traffic flows and incur a nontrivial number of empty miles due to equipment repositioning. Therefore, it would be both interesting and economically advantageous to take this issue into account. This broader approach would result in better equipment usage and a more “systemwide optimal” solution.

Finally, our results are not limited to applications within the rail industry. Indeed, any transportation setting concerned with the tradeoffs between sending freight, or even passengers, directly versus indirectly can benefit from this work. For example, commercial airlines face decisions regarding the time-cost tradeoff of sending their passengers on direct (nonstop) versus connecting flights. Federal Express routes its packages through hubs. The trucking industry is interested in these concerns, especially for less-than-truckload operations. We could extend the techniques and analysis developed in this dissertation to these other transportation settings.
References


Appendix A

A.1 A Technical Note on Parameter Settings

All problem instances were solved on a Sun SparcStation with 128 megabytes of RAM, and with the most current version of CPLEX software available at this time, Version 5.0. Results obtained via a straightforward implementation of CPLEX, from the decomposed subproblems, and from the problems with the added valid inequalities were obtained after adjusting the software parameter settings (i.e., the results were not obtained with the default settings).

Specifically, we varied the \textit{backtrack} parameter and observed improvements of about 5% in the objective value for initial problem runs when this parameter was set to "3", specifying the use of a depth-first search. Similar improvements were realized when a depth-first search was used to obtain solutions to any of the subproblems in the decomposition procedures. The parameters and structure of the problem are such that there are many feasible, similar alternatives with few dominating or dominated strategies. A breadth-first search results in less progress, because the solutions at the top of the branch and bound tree yield similar objective function values.

Using a setting of "1" for the \textit{nodeselect} parameter also proved useful. The \textit{nodeselect} parameter controls the method for choosing the node to branch on when backtracking through the branch and bound tree because of an infeasibility or branch pruning. Setting \textit{nodeselect} to "1" specifies the "best bound" rule for backtracking which reduces the memory usage substantially; however, there is little improvement in the initial bound, even if the process is allowed to run for hours.

The \textit{variableselect} parameter selects the variable on which to branch at each node. We set the parameter to "1" to choose the variable with the largest fractional value in the linear
programming relaxation. This tightens the lower bound by 1% to 2% in our instances.

Other parameter settings were investigated, for example, generating clique and cover sets, but with no noticeable improvement in performance. Finally, priority branching (specifically, on various subsets of direct trains) was also implemented, without significant improvements. We suspect the cause for this to be the cost structure of the problem which leads to many similar, feasible alternatives.