FREIGHT ROUTING AND CONTAINERIZATION IN A PACKAGE NETWORK THAT ACCOUNTS FOR SORTATION CONSTRAINTS AND COSTS

Final Report

Metrans Project 03-07

February, 2004

Apichat Chayanupatkul
Randolph W. Hall
Daniel J. Epstein Department of Industrial and Systems Engineering
University of Southern California
Los Angeles, CA 90089-0193

ACKNOWLEDGEMENTS

I first wish to express my gratitude to my advisor, Professor Randolph Hall, for his guidance throughout this research, without which this work could not be completed. He provided me insight into this area of research and had been very helpful for ideas to initiate the work. With his strong research experience and scientific knowledge, he has supplied me valuable suggestions and constructive feedbacks for the whole process of conducting this research. It is my most appreciation to learn from him. Thanks to Dr. Fernando Ordonez, for his fruitful comments and suggestions on developing problem-solving approaches and his support in using CPLEX 8.0 on the 'illposed' server. Thanks to all the faculty members and staffs of the Daniel J. Epstein Department for great academic environment and friendly atmosphere.

I cannot give sufficient thanks to my family; my father, my mother, and five of my sisters for their continuing support and precious encouragement during difficult times.

Last but not least, I would like to acknowledge with gratitude the financial support of US DOT and California Department of Transportation through the METRANS Center throughout this research.

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, and California Department of Transportation in the interest of information exchange. The U.S. Government and California Department of Transportation assume no liability for the contents or use thereof. The contents do not necessarily reflect the official views or policies of the State of California or the Department of Transportation. This report does not constitute a standard, specification, or regulation.

ABSTRACT

In this research, we investigate package transportation routing that accounts for sorting activities at the terminals visited. Due to the large number of packages originated from different locations and destined to different locations, at each terminal visited, sorting operations are essential to regroup and dispatch incoming shipments from different origins to different destinations. This process is time, labor, and resource intensive. Proper containerization allows shipments to bypass terminals without entering sorting operations. The efficient utilization of sorting facilities, in turn, enables parcel carriers to handle more shipments with existing resources while maintaining the same service level. However, container capacity and limits on alternative routes resulting from bypass may cause the containers to be moved via longer routes. Thus, the efficiency of package transportation operations may deteriorate, leading to higher transportation costs.

The primary objective of this research is to study package routing that accounts for containerization to minimize transportation costs and sorting costs imposed. To obtain container routings, a network is modified so that each arc represents a container group on a flight. The solution method is segmented into three phases: construct a new network, perform LP relaxation, and find the solution of the original problem. The solution provides a sequence of container groups along the whole route for each commodity. Three heuristic approaches are developed and evaluated: the grouping heuristic (GH), the forcing constraint heuristic (FCH) and the combined heuristic (CH). The approaches are benchmarked with lower bounds calculated by solving the LP relaxation with forcing constraints. Model extensions for congested periods of sorting operations are also proposed. To account for the sorting option at source terminals, heuristic approaches that build from the GH and the FCH are developed and examined under different parameters and problem instances.

Experimental results show that the GH is the fastest algorithm and provides solutions within 4% of lower bounds for test cases. The FCH outperforms the GH in terms of solution quality, in which solutions are within 2% of lower bounds for test cases, but it is slower than the GH. As for the CH, computation time is between the GH and the FCH but it yields inferior objective values relative to the FCH and the GH. Sensitivity analysis shows that container capacity has little effect on the solution quality. Sorting costs, however, significantly affect the heuristic performance.

Results for the model extensions correspond to the results from the core model in which the developed heuristic incorporated with the GH is faster than one with the FCH but yields poorer objective values. Sensitivity analysis shows that increase in sorting costs at source terminals has impacts on the heuristic performance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

DISCLOSURE

Project was funded in entirety under this contract to California Department of Transportation.

1. INTRODUCTION

1.1 Motivation and Objectives

In the past few years, parcel carriers have enjoyed explosive growth due to a large number of transactions emerging from e-commerce. Among these carriers, FedEx and UPS, the two largest in the parcel distribution industry, have benefited most from the expansion of e-commerce. They provide similar classes of package delivery services; however, FedEx is stronger in air transportation while UPS dominates in ground transportation. Other freight carriers focus on different sectors and provide weaker services than FedEx and UPS. DHL, for example, has concentrated on international shipments and provides fewer service options for domestic freight.

The strength of FedEx in air transportation implies that the company focuses more on express shipment services. With 56,000 drop-off locations, 640 aircraft, and nearly 54,000 vehicles, FedEx is considered the world's #1 express delivery company. On the other hand, with truck-based transportation, UPS is the world's largest package delivery firm delivering more than 14 million parcels per business day throughout the United States and to more than 200 countries and territories. To survive in a highly competitive market, parcel distribution companies need to exploit optimization models to offer low service costs to customers while maintaining service standards as promised.

The domestic service options that both companies offer are similar and can be segmented into three broad categories: Next day, 2nd day and 3-5 days services. Each category can be further divided into several classes with slightly different delivery commitments. For instance, UPS offers two options for 2nd day service: typical 2nd day, which provides on-time delivery to every address on the second business day and 2nd day AM which is guaranteed delivery by noon on the second day but this service option is available only for delivery to metropolitan commercial addresses. The essence of all the services is time commitments, especially for express services for which customers pay a premium to receive faster and more reliable services. Thus, shipment transportation time is significantly constrained.

A shipment transportation process can be divided into two sections: local distribution and long haul transportation. The local distribution involves pickup and delivery in small service areas, performed by tractor-tailors or trucks. The long haul transportation occurs over relatively long distances between major terminals where packages from pickups in local areas are aggregated. The long haul operation may be carried out using ground service exclusively, air service exclusively, or a combination of two transportation modes.

Associated with the transportation process are sorting operations at each terminal. Each afternoon, packages from several local operating centers are directed to nearby hubs. The hubs are served as exchange points where packages from different directions are sorted, consolidated, and redirected to their destinations. At a sorting terminal, packages are unloaded and put onto conveyor belts for sorting. When the sorting process is completed, the packages are put onto a feeder vehicle, which may be truck or airplane, and routed to other hubs.

In most terminals, packages are unloaded by manually lifting and lowering onto conveyor belts. Equipped with high-speed conveyors and overhead scanners, some terminals can significantly reduce the lifting and lowering of packages and the average package cycle-sorting span. Currently, UPS's Louisville hub can process 500,000 packages per hour, improving from the sorting rate of 165,000 packages per hour before the sorting automation is implemented. Nevertheless, due to high costs of the automated equipment, the system is only implemented in a few large sorting terminals. Also, since packages are collected from operating centers and sent into the hub at almost the same time, especially in each afternoon, sorting facilities and labor are only required during peak periods of a day. Thus, it may not be worthwhile to make a big investment in the automated system. Rather, the efficiency of sorting resource utilization should be thoroughly investigated.

In the field of operations research, various problems in the freight transportation process have been extensively studied. Numerous optimization models have been developed with a variety of scenarios. However, less studied, yet important in practical situations, is the package transportation problem that accounts for containerization and sorting processes. The sorting process is a critical step to the performance of the system since it is time, labor, and resource intensive. Packages should enter the sorting process only when necessary to lessen transportation time, save operating costs at terminal and conserve resources for vital operations. This can be accomplished by allowing some containers to bypass sorting operations. The advantages of proper containerization are as follows.

1) Sorting resources and labor are more efficiently utilized, resulting in the reduction of operating costs.

2) Parcel carriers are capable of handling more shipments with existing resources while maintaining the same service level.

3) Without involving unnecessary processes, the percentage of delayed shipments can be reduced.

In this research, optimization methods are developed for package transportation that explicitly account for sorting operations. The focus of this research is shipment routing assignment, incorporating containerization. The solution will provide shipment routes beginning from operating centers, where packages are locally aggregated to their destination operating centers, where packages are prepared for local deliveries. Each shipment route accounts for container groups, which identify sorting activities along the whole route. Different service classes are also considered to conform to real-world scenarios. The innovation of this research is providing a new approach for the large-scale package transportation problem by comprehensively considering sorting operations and containerization.

The primary objective of this research is to study package routing and containerization over a large network and present optimization models to minimize total relevant costs under a set of realistic constraints. Besides determining a sequence of terminals for each shipment, container assignment on each route is also provided to specify where packages are sorted and how they are grouped.

1.2 Package Transportation Process and Industry Practices

1.2.1 Overview of package transportation process

The package transportation process begins from pickup. Shipments can be collected in various ways. At UPS, for instance, three types of services are available: regular customers, drop boxes, and call-ins. Packages from local pickup operations are aggregated in local operating centers, which serve as home bases for pickup and delivery vehicles. The operating center is the minimal unit that distributes local parcels. The following describes two general transportation modes by which packages, collected at local operating centers, may be moved.

Ground transportation

After being aggregated in a local operating center, packages are moved by tractor-trailers or trucks to a major terminal serving as a hub. In the hub, packages from several operating centers are sorted by ZIP code, consolidated, and then transported to their destinations via the carrier's ground network. During their journeys, shipments may be resorted and consolidated in a number of terminals before arriving at the destined local operating centers. Then, all the incoming packages at the local operating centers are sorted, scheduled, and loaded onto feeder vehicles for local delivery.

Shipments should be moved by ground service option if possible since transportation costs are much cheaper. Even for express shipments, ground transport may be feasible if the distances between origins and destinations are relatively short. Also, since only business days are counted for all services, additional time is available over weekends, permitting ground service to meet the time commitment.

Air transportation

An air network includes a set of airports and a set of aircraft flights that connect these airports. After packages are collected at a local operating center, shipments requiring air services are routed to nearby airports. Generally, a local operating center is served by a single airport; however, some operating centers may be served by two or, in rare cases, three airports, depending on its geographic location. Airports come in different sizes with different ranges of connectivity. In a rural area where long flights are not available, to meet the requirement of time standards, packages from local operating centers are moved to a rural airport and air fed to larger airports, and then generally directed to hub airports by flights originated from the larger airports. Coast-to-coast shipments mostly pass through a hub airport, which provides large sorting facilities. However, shipments from major airports may or may not be routed to the hub due to the connectivity of major airports. At the hub airports, some containers might be opened and shipments in the containers individually sorted. Some containers with pre-sorted destinations are not necessarily opened and can be directly loaded onto outbound aircraft. After reaching the destination airports, the shipments are distributed to local operating centers and then ground fed to their final destinations.

Associated with each shipment is a priority class specifying commitment time. Taking account of service classes adds complexity to the problem of containerization since packages are classified into more categories. For example, a shipment with three day service may be transported to a hub airport on the same day the shipment is received but, at the hub airport, all outbound aircraft are fully utilized. The shipment could be stored at the hub airport for later departing aircraft as long as the time commitment is satisfied. For this scenario, to provide operational planning, we need to account for sorting capacity and aircraft capacity of later days, which significantly complicates routing assignment.

The typical air shipment routing is illustrated in Figure 1.1.

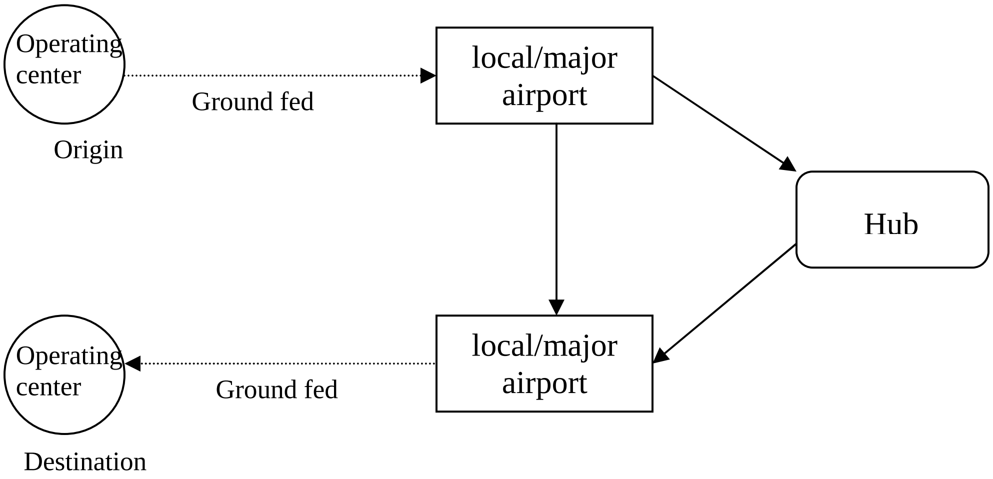

Figure 1.1. Package routing via air network

The typical package transportation process of door-to-door service can be divided into two elements: local distribution and long-haul transportation. In optimizing package routing, these two segments are often separately solved. For local pickup and delivery operations, since every address is usually served by a single pickup (delivery) center in the service area, it is unnecessary to consider connection between the operating center and other terminals and the association between local distribution and long-haul transportation can be neglected. The optimization of local operations is to assign vehicle routes starting from and ending at the same operating center to perform transportation activities in the area. The local distribution problem has been extensively studied in the name of vehicle routing problem (VRP). Laporte [1992] provides a general review of the VRP. VRP models and methods can be found in Golden et al. (1998), Laporte and Louveaux (1998), Toth and Vigo (1998), Zhong and Hall (2000).

The focus of this research is on long-haul transportation, which is concerned only with the movement of packages over long distance, starting from local pickup center and ending at local delivery center.

1.2.2 Terminal operations

In general, air shipments pass through local terminal, originating airport, hub airport, destination airport, and local terminal. Sorting operations are usually involved in every terminal visited. Sorting time at each terminal may depend on numerous factors such as number of sort categories, sorting capacity, sorting facility, number of incoming shipments, etc.

The terminal operations at an intermediate airport can be described as follows.

1. Unload containers from inbound aircraft to ground.
2. Transfer containers to sorting station.
3. Sort the shipments in containers according to their next airstops.
4. Transfer the sorted containers to outbound aircraft.
5. Reload the sorted containers onto outbound aircraft.

The terminal operations can be exhibited on Figure 1.2.

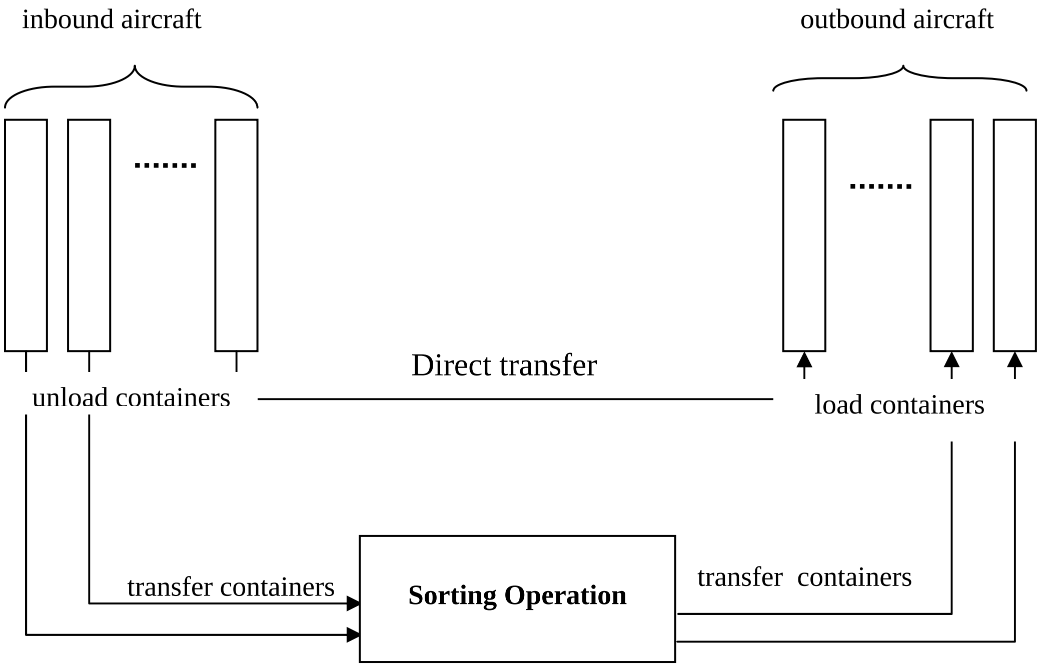

Figure 1.2. Terminal operations at intermediate airport

Due to limited sorting capacity and incurred costs, the sort should only be performed if necessary. Containers arriving at an intermediate airport may contain shipments with the same next airstop. In this case, the containers can bypass the sorting process; i.e., after unloaded from aircrafts, the containers are directly transferred and loaded onto outbound aircraft. With containers properly bypassing the sorting process, a sorting facility is more efficiently utilized. For the whole transportation network, this means parcel carriers are capable of handling more incoming shipments with existing resources.

However, shipments must be classified into several categories defined by o-d pairs and service classes, each with numerous possible routings. A larger number of categories results in a smaller number of shipments for each category, making it difficult to fully utilize container capacities. Only a few containers can be filled by all shipments with the same category to take advantages of bypassing along the whole route. Putting shipments with different categories in bypassing containers restricts routing alternatives as the number of possible routes for each category is significantly reduced. This may cause some shipments to be transported via longer routes than they should and higher transportation costs may be incurred. As a consequence, the efficiency of transportation must also be carefully considered and needs to be balanced with the efficiency of sorting operations that benefit from bypass.

In air transportation, aircraft schedules are predetermined according to tactical (medium-term) planning since it is infeasible to change aircraft schedules from day to day. Aircraft schedules are determined by average demand collected from historical data over a certain period. Thus, daily air shipment flows must follow the aircraft schedules. On the other hand, for ground transportation, truck routing is much more flexible. Although, to some extent, certain truck routes are already assigned based on tactical planning, these routes may be altered if demand is not consistent from day to day to efficiently utilize resources and achieve the best possible performance of the system. Truck routing adjustment is possible as the routing assignment is performed by local management.

1.3 Contribution of Research

Sorting operations are a critical step in parcel transportation process as a large number of shipments with different origin-destination pairs are involved in sorting facilities. The problem we consider is different from a general multicommodity flow network routing problem in that the number of shipments to be sorted at each terminal is bounded. Intuitively, limitation on sorting capacity at terminals may result in deferred transportation when the number of shipments entering the sorting operation exceeds the sort limit. On the other hand, shipments may be directed via alternative routes to avoid queuing at fully utilized terminals; however, higher transportation costs may result. To minimize transportation costs while satisfying sorting capacity constraints, bypass containers can be exploited so that shipments do not need to enter every sorting facility they visit. Thus, the whole system will be capable of handling more shipments under delivery commitment time restrictions.

By enforcing an additional set of sorting constraints restricting the maximum number of incoming packages to be sorted at each terminal, shipment routings will be altered to satisfy the sorting capacity while minimizing transportation costs and sorting costs. It is imperative to consider entry units for sorting operations as containers, not individual shipments, to conform to real-world operations. This underlying fact adds complexity to the network routing problem since the problem becomes a large-scale MIP. Due to the large size of the network, we develop a set of heuristic procedures for this scenario to keep computation time reasonable. The solution to our optimization model provides routing and container grouping decisions to utilize the resources (i.e. a fleet of vehicles and sorting facility) as efficiently possible.

1.4 Organization of the report

The report is segmented into 7 sections. Section 1 provides an introduction to industry operations and description of shipment transportation processes. Review of relevant papers is presented in Section 2. In Section 3, the model of sorting process and arrival patterns of aircraft are described in detail. Section 4 provides the developed containerization model and formulation. The optimization methodology and experimental results of the core model are presented in Section 5. Model extensions and experimental results are exhibited in Section 6. Last, conclusions of the report are provided in Section 7.

2. Literature Review

2.1 Multicommodity Flow Research

The problem we consider is similar to the blocking problem in the rail transportation area. A block comprises a set of cars with the same origin and destination. The block may pass through several yards on its route from origin to destination. Sorting activities are completed at yards where incoming cars are selected, inspected, and assembled into blocks according to their destination yards. Similarly, in our problem, we need to decide which containers should be sorted, and consolidated at each terminal they pass through. A car in the blocking problem may be viewed as a container in the containerization problem and a block may be considered as an airplane or truck. The constraints on yard capacity in the rail formation problem can be considered as sorting capacity constraints in our problem. In general, the objective of the railroad blocking problem (RBP) is to assign sequences of blocks to deliver shipments to minimize total mileage, handling and delay costs. The modeling efforts for the RBP in the literature are summarized as follows.

A heuristic approach based on linear function was first applied to determine freight routing for rail network by Thomet [1971] with the objective of minimizing the sum of delay costs and operating costs. Petersen and Fullerton [1975] study models of line and yard operations for delay minimization in the context of Canadian railways. A primal-dual assignment algorithm is used to solve the model. Assad [1980] reviews the previously studied models for rail transportation. The author categorizes important issues in the rail systems into several sections, including facilities location and installment, yard and terminal models, line models, blocking and train formation, and train schedules and timetables. In the following paper, Assad[1982] analyzed classification policies using a measure of work that reflects the peculiarities of processing cuts. A dynamic programming approach is formulated for the general problem and the form of the optimal policy is analytically derived under special assumptions. Bodin, Golden, and Shuster [1980] present an arc-based multicommodity flow formulation for the railroad blocking problem to determine a classification strategy for all the classification yards in a railroad system at one time. They use a piece-wise linear model for shipping, handling, and delay costs. However, priority classes are not considered in the scenario. The problem is solved by a heuristic algorithm based on relaxation techniques. Crainic, Ferland, and Bousseau [1984] develop an MIP model that simultaneously addresses the blocking, train routing, and block-to-train assignment decisions. They also consider different priority classes in their model. Decomposition heuristics and column generations approach are applied to solve the non-linear, MIP model.

Daganzo [1986] investigates how the allocation of blocks to classification tracks affects switching work. Given are the sorting classes, as well as the schedule and block makeup of the departing trains. The author develops simple formulas for the minimum number of tracks needed to implement a strategy, and for the amount of switching work given a number of tracks. The objective is to minimize the amount of switching. Haghani [1989] proposes a decomposition heuristic approach to solve the problem of train formulation and distribution of empty cars. Keaton [1989, 1995] uses Lagrangean relaxation technique to solve a linear MIP model for rail routing problem. The author also considers limits in the yards capacity in the model. In their series of papers, Martinelli et al. [1994, 1995] present neural network and genetic algorithm techniques to solve non linear, MPI models for rail routing with different objective functions. Bostel and Dejax [1998] address the problem of container loading on trains in rail-rail transshipment shunting yards to minimize the transfers within the yard. The problem is to determine the initial loading place of containers on arriving trains and their reloading place on departing trains to minimize their transfers within the yard and therefore the use of handling equipment. Newton, Barnhart and Vance [1998] model the problem as a network design problem in which yards are represented by nodes and blocks by arcs. Their model also considers priority classes for shipments as they have different service requirements. Column-generation approach and branch-and-bound algorithm are employed to find near-optimal solutions. The objective is to choose which blocks to build at each yard and to assign sequence of blocks to deliver each shipment to minimize total mileage, handling, and delay costs.

2.2 Network Design Research

Network design models have been extensively used to represent a broad variety of operations in transportation. The formulations can be used to determine decisions as to the logistics structure, the service network, and the operation of long distance freight transportation systems. In this research, we focus our attention on express shipment routing problems, which include both modes of transportation with more emphasis on air transportation. Generally, the problem is to determine feasible routes under time and resource constraints over a network where demand is defined between pairs of origin destination nodes. In air shipment transportation, several network models have been used to capture the characteristics of real world situations.

Kuby and Gray [1993] study a real world hub network design problem by looking at the actual network used by the Federal Express Corporation. They develop a path-based {0,1} network design model to design the least-cost single hub air network assuming that the hub location is already determined. The model is solved with a standard mixed-integer programming. They also perform analyses showing the cost effectiveness of a design with multiple stops over a pure hub-and-spoke network.

Barnhart and Schneur [1996] study the design of aircraft routes and schedules to pick up and deliver the shipments, using a single hub. The authors present models, solution procedures, and results in designing and implementing Express Shipment Service for a large carrier. However, in their model, there are several operational restrictions that greatly simplify the problem. Kim et al. [1999] extend the paper by considering more realistic constraints, resulting in the much higher complexity of the problem. The objective is to find the cost minimizing movement of packages from their origins to their destinations, given very tight service windows, limited package sort capacity, and a finite number of ground vehicle and aircraft. Their strategy is to use column generation techniques to limit LP size. Then, branch-and-bound is used to find an integer solution.

To solve the problem of the reorganization of the German postal services, Grunert and Sebastian [1999] decompose the problem into several subproblems and then use vehicle and routing techniques for the optimization of the night airmail network, the design of groundfeeding and delivery transportation system, and the scheduling of operations.

2.3 Large-scale Optimization Research

In practice, most transportation network problems are very large and cannot be solved to optimality within reasonable computational time. To deal with such large-scale network problems, several approaches are exploited to reduce the problem size and find near-optimal solutions with less computational effort. Although most practical problems are non-linear in nature, LP approximation can yield good results. The column generation approach is one method that has been extensively used. The method corresponds to Dantzig-Wolfe [1960] decomposition where the constraints are placed in the subproblem. The network constraints are divided between the subproblem and the master problem. Then, the two problems can be solved separately.

Glickman T. and Sherali H. (1984) study the optimization of empty freight cars distribution with the objective of total minimum cost. A large-scale network algorithm is used in conjunction with decomposition to obtain solutions. Barnhart and Sheffi [1993] develop a primal-dual heuristic approach for solving large-scale multicommodity networks and apply this technique to the problem of vehicle routing optimization. Desrosiers et al. [1984] uses the column generation approach on a set partitioning problem solved by simplex and branch-and-bound; column are generated by a shortest path algorithm with time windows on the nodes. Crainic and Rousseau [1987] propose a column-generation type algorithm to find a good set of pairings for the airline crew scheduling problem. Barnhart et al. [1998a] propose formulations of integer programs with a huge number of variables and their solution by column generation methods. They present classes of models for which this approach decomposes the problem, provides tighter LP relaxations, and eliminates symmetry. Computational issues and implementation of column generation, branch-and-bound algorithms, including special branching rules and efficient ways to solve the LP relaxation are also discussed. Barnhart et al. [1998b] apply this approach to real-world applications. They present a string-based model and a branch-and-price solution approach to solve both the fleet assignment and aircraft routing problems. Basically, Branch-and-price is branch-and-bound with a linear programming relaxation solved at each node of the branch-and-bound tree using column generation. Barnhart et al. [2000] consider an origin-destination integer multi-commodity flow problem, a constrained version of the linear multicommodity flow problem in which flow of a commodity (defined in this case by an origin-destination pair) may use only one path from origin to destination. Examples include bandwidth packing problems and package flow problems. They present a column-generation model for the class of integer multicommodity flow problems and a branch-and-bound solution approach involving column and row generation. Rexing et al. [2000] study airline fleet assignment with time windows problems. Due to the vastly large size of the problem, it is first reduced by preprocessing steps and then they use the direct solution approach (DST) to solve the problem. The concept of the algorithm is to iteratively add necessary flight arc copies to the model, thereby minimizing the model size.

The issue we consider is a combination of the rail classification and network design problems. In package transportation processes, involvement of sorting operations is critical to the time commitment. The tradeoff here is the flexibility of routing assignment and the time saving from sorting operations. From the previous study, there has been little research in the subject of freight routing that includes sortation in the models. In this paper, sorting operations have been explicitly considered for package routing problems over a large-scale network.

2.4 Terminal Operation Research

At terminals, packages are unloaded from aircraft and placed on conveyor system to be sorted. To prevent overflows that can occur when one or more conveyors merge on another conveyor segment, the package routing must be assigned. Geinzer C. and Meszaros J.(1990) present an application of conveyors within a high volume sorting system for a Federal Express distribution facility with the objective of maximizing the flow of packages through the conveyor system in a given time period. Alternative approaches for modeling high volume conveyor systems are presented and discussed. A simulation model is used to evaluate how the conveyor system reacts to unexpected conveyor failures and overflows. Nobert and Roy [1998] present a new model and an appropriate solution approach for the work-scheduling problem at air cargo terminals using data provided by a major Canadian airline. The problem they study is segmented into two sub-problems. The first one consists of determining the manpower requirement for each job category and for a typical workday in a given planning period. The second comprises designing efficient work schedules that satisfy the manpower requirements and comply to all rules and regulations and budget constraints that apply to this problem. An integer linear programming model is applied to the scheduling model formulation.

3. Model of Sorting Process

In this section, the sorting process and its critical elements will be described, as well as the effects of containerization on the whole system. The sorting operation is inherently a queuing process, which requires a proper strategy to achieve optimal facility utilization under time commitments and sorting capacity constraints. The sorting strategy includes sorting start time, processing time, order of shipments to be sorted, and completion time. These factors must be taken into account in accordance with sorting capacity, the shipment arrival pattern, and allowable processing time for each terminal.

Arrival patterns of shipments affect the sorting strategy with respect to keeping the sort busy to effectively utilize the sorting facility. The arrival patterns of the ground and air layers are different and should be individually examined. As for the ground layer, shipment arrivals at a terminal at the end of day are less predictable as a result of uncertainty of ground traffic from the dispatching terminal. For the air layer, shipment arrivals at intermediate airports are not likely to vary as much from day to day due to small deviations of flying times. However, in the US, arrival patterns of aircraft differ among airports in different regions due to time zone differences and locations. The variations of arrival patterns result in different intervals of allowable processing time and, thus, different strategies shall be applied.

To complete the sort under allowable time at a terminal, the sorting capacity must be sufficient to handle incoming shipments during peak periods of the day; the shortage of sorting facility has tremendous effects on departure delays at the terminal, leading to arrival delays at subsequent terminals. Two ways to solve terminal delays are: increasing sorting capacity and reducing the number of shipments to be sorted. The first option, increasing sorting capacity, requires an investment in sorting equipment, which may not be worthwhile since the sorts are in high demand only during congested periods of the day. The second option is the approach we are addressing. The number of shipments to be sorted can be reduced under proper containerization as shipments are consolidated so that fewer sorts along their routings are required. Nevertheless, containerization has a price to pay. The containerization operation requires additional time at the originating terminal, resulting in arrival delays at subsequent terminals. In addition, bypass containers may waste because demand is insufficient to fill them, leading to lower aircraft capacity utilization. However, with proper containerization, these arrival delays may have minor effects on the shipment queuing process and the aircraft capacity utilization could be traded-off with sort savings. These effects of containerization will be discussed in detail in this section.

In developing a containerization model, three factors need to be considered: arrival patterns of incoming shipments, the sorting time incurred in containerization, and the sorting rates. The optimal results could yield better productivity of sorts or earlier completion times. This section provides insights into the sorting process and how the model is developed under the sorting capacity and time constraints.

3.1 Physical Description

In this section, we will discuss physical characteristics of a network related to sorting processes and considerations in developing a prescriptive model of the system.

The physical network for package transportation consists of two layers: air and ground networks.

The air network comprises directed links (air flights) connecting airports spread across the country. There are five types of airports: national hubs, regional hubs, major airports, local airports, and rural airports, each of which has a sorting capacity and a set of inbound flights, and set of incoming shipments. The national hub holds the largest sorting capacity and serves all origin-destination pairs. The regional hubs have the second largest capacities, followed by major airports, local airports, and rural airports, respectively. Each sorting capacity is designated to serve the estimated numbers of inbound flights and incoming shipments at the terminal. The connectivity between airports is based on airport types. Larger airports tend to have more inbound and outbound flights and thus have more options to route shipments. The network connectivity will be discussed later in Section 4 in detail.

The ground layer we consider is composed of local terminals where shipments are collected from local service centers, which are the smallest units of terminals. Packages from local service centers are typically routed to a single nearby local terminal. At a local terminal, shipments are sorted and express shipments are directed to nearby airports while non-express shipments are usually transported via trucks. Typically, local terminals have a single option for air shipments but some terminals in large metropolitan areas could have two alternatives.

Inbound and outbound vehicles of each terminal type are shown as follows.

Terminal Types	Inbound	Outbound
Local Terminal	Trucks	Trucks
Local/Major Airports	Trucks	Planes
Regional/National Hubs	Planes	Planes

Table 3.1 Inbound/outbound vehicles for each type of terminals

Arrival patterns of aircraft

An aircraft can depart from the originating airport after the sort for the aircraft has been completed and the aircraft have been fully loaded. The latest time that an aircraft can depart the originating airport is represented as the cut-off time. The cutoff time varies from flight to flight, depending on travel time from originating cities to the sorting terminal and time zone differences. The difference between the cutoff time and the arrival time specifies the time window for transporting packages from origin airport to destination airport.

Effects of Time Zones for National Hub

In the United States, the time lag of the west coast behind the east coast tightens the time window of transporting shipments from the west coast to the east coast and conversely elongates the time window from the east coast to the west coast. Thus, the arrival pattern of aircraft depends on both distance and time zone differences which, in turn, depend on locations of city pairs. Among all types of terminals, a national hub is particularly susceptible to time zone differences because incoming shipments come from all regions across the country. Hall (1988) studies the effects of locations of national hubs on arrival patterns of aircraft. The result indicates that, if a national hub were located in the Eastern Time zone region, the arrival times of aircraft would show relatively large variations because planes from other regions depart later. On the other hand, locating a national hub in the Pacific Time Zone region would make variations of arrival times relatively small as a result of earlier departures from other regions. The arrival patterns of aircraft at terminals in central regions lies in between the two extremes.

Figure 3.1 shows a cumulative curve of aircraft arrivals of a national hub in different regions. The curve consists of several steps, each representing arrivals from different regions.

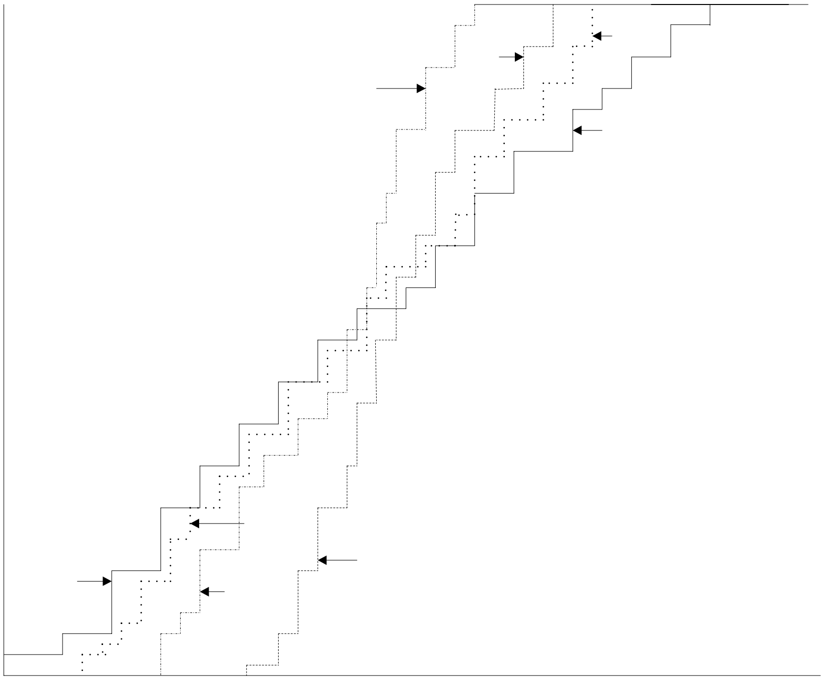

Figure 3.1. Arrival patterns of aircraft at different regions

The differences in arrival time intervals over different regions across the country result in different time windows for sorts, leading to different strategies in allocating sorting capacity.

Assuming that the numbers of incoming shipments at two terminals in the east and west are identical, at the eastern terminal, the high standard deviation will result in a longer timespan for sorting process. Therefore, the sorting rate can be reduced, leading to lower costs in investment in sort equipment. On the other hand, to complete the sorts under a short time window, the western terminal with a low standard deviation requires a higher sorting rate and, thus, the equipment cost will be larger. However, the advantage of the low standard deviation over high standard deviation is that planes do not need to wait as long for departures due to shorter timespan.

Unlike a national hub, time zone differences are not a critical issue for regional hubs because they serve as exchange points of shipments in their regions. Most inbound aircraft originate from airports in the regions and, thus, travel distances and time zone differences are not as great among inbound aircraft.

3.2 Sorting Operations/Containerization

Sorting operations could be different for each terminal type due to terminal sizes and inbound and outbound vehicles. This affects the containerization process as well as benefits from containerization for each terminal type. This section will describe sorting operation and the role of containerization for each terminal type.

3.2.1 Local Terminals/Rural Airports

The first sort in the package transportation process is at local terminals, where trucks are dispatched and shipments from several local operating centers are collected. These local terminals also serve as sorting facilities in the vicinity of cities served. Incoming shipments are first sorted into express and ground shipments.

Ground shipments are further sorted by ZIP code, consolidated, and loaded to outbound trucks. At this point, the packages enter a long-haul transportation process. The shipments are then transported from local terminal to local terminal to their destinations via the ground network. Most express shipments will be directed to large airports in the vicinity, but shipments will be routed via the ground network if time windows are sufficient. At an origin airport, incoming shipments are sorted, put into containers, and loaded onto outbound aircraft. Then, shipments are directed to airports or hubs before reaching their destination airports. The sorts may occur at any intermediate airport along the entire route.

At local terminals, packages can be containerized after being sorted so that the containers do not need to be sorted at the airports. However, it could be more economical to transport all incoming packages to the airports, which are larger sorting facilities, without being sorted because of economy of scale of sorting costs. Also, since inbound and outbound vehicles at local terminals are transported by trucks, they are flexible in a sense of adjusting routing options. Doing so, we also need to consider sorting capacity of the terminal where shipments are transported to, as well as routing costs. It is likely that transportation costs would increase; this element of costs must be traded-off with sorting costs saved from changing sorting facility.

Typically, a local terminal is connected to one or two airports. In the case of single airport connection, routing option from the local terminal is restricted so consideration in minimizing total costs are sorting capacity and sorting costs of the connected terminal. In the case of two optional connected airports, besides those considerations, routing costs and flight capacities are also major concerns in optimizing the whole network.

Rural airports and local terminals have a common characteristic in a sense that they are the very origins of package routings and also have one or two connected airports. However, containerization at rural airports is not applicable due to the size of aircraft from rural airports.

In this research, we focus on package transportation comprising both ground and air layers. The ground layer we consider is the portion of the ground network that provides ground feeders to airport terminals. The air layer is the entire air network employed in routing air packages.

3.2.2 Local/Major Airports

At each intermediate airport, shipments that need to be sorted are unloaded from aircraft and enter the sort operation. The sorting operations can be segmented into two relatively distinctive processes – automated and manual systems. Automated systems are typically employed in large terminals to handle a huge number of packages in a short time period. The systems exploit information on package labels issued at originating centers to categorize and move the packages. The automated sorting operations can be described as follows. An incoming package is unloaded from a delivery vehicle, aircraft or truck, and then put onto a system of conveyor belts. While the package flows through the system, the package label is scanned by a detector camera located above the package. The code on the package is decoded to provide information as to its geographical destination and to determine its drop-off point for the sort system. For air shipments, sorted packages with the same destination group are consolidated and put into a container before being loaded onto outbound aircraft. In small terminals, packages are manually sorted by operators. The finer the categories to be sorted, the more time and costs are incurred. Thus, it is necessary to determine the best number of categories for the sort.

Typically, an individual outbound vehicle can depart when all the shipments have been sorted and fed to the vehicle. Thus, the completion time is defined by the time required to sort the entire set of shipments. For ground shipments, this may not be the case since truck capacity is much smaller than aircraft capacity so each truck can depart as soon as filled. The completion time of ground shipments is not as critical as that of air shipments since the air service requires that shipments be picked up and delivered within specified small time intervals.

3.2.3 Hubs

At the air hub, the sorting process begins as soon as the first plane arrives and ends when the outbound aircraft have been filled. Sorting operation at hubs requires careful attention to the arrival pattern of aircraft, especially at a national hub, as discussed in the previous section. Also, due to the large number of incoming packages, significant savings could be made at hubs. A national hub processes packages from every region across the country and send packages to every direction. The distance and the time zone of its destination are taken into account to indicate the latest time the shipment may leave and still meet the time commitment. This latest time is used to identify which shipment should be processed first.

3.3 Queuing Model

3.3.1 Productivity and Completion Time of Sorting Facility

Regardless of terminal locations, it is desirable to keep sortings busy to effectively utilize sorting facility. The productivity, however, must be traded off with shipment waiting time, which, in turn, affect the completion time of the sorts. In this section, we model shipment arrivals as a queuing process to investigate interactions of productivity and waiting time in sorting operation.

Let

S = Maximum sorting rate in terms of number of shipments per minute.

A = Number of total arrival shipments.

I = Idle time of the whole sorting process.

$\Lambda(t)$ = Cumulative arrivals of shipments at time t.

$\Omega(t)$ = Cumulative processed shipments at time t.

N(t) = Number of containers arriving at time t.

t_s = Start time of the sorting process.

t_c = Completion time of the sorting process.

t_i = Arrival time of airplane i.

s_i = Number of shipments on airplane i.

w(t) = Number of shipments waiting to be processed at time t.
= $\Lambda(t)$ - $\Omega(t)$

Then,

$$\Lambda(t) = \sum_i s_i \quad \text{, where } t_i \le t. \tag{1}$$

Let r(t) = Sorting rate at time t. Then,

$$r(t) = \begin{cases} S, w(t) > 0. \\ 0, w(t) = 0. \end{cases} \tag{2}$$

And assuming that the sort begins as soon as the first shipment arrives, the start time, $t_s = 0$,

$$\Omega(t) = \int_0^t r(t)dt \tag{3}$$

The completion time, t_c, is the time that all shipments have been processed. Assuming that $t_s = 0$, to maximize the productivity, t_c must be minimized, which is tantamount to minimizing idle time, I. The idle time depends on the arrival patterns of shipments. Since $t_s = 0$, the completion time is total time composed of idle time and operational time. As the sort occurs, sorting rate is S. The operational time is total arrival shipments divided by sorting rate. Thus, the completion time, t_c, can be defined as

$$t_c = I + A/S \tag{4}$$

The productivity is the ratio of the actual number of processed shipments to the number of shipments that could have been processed during the sort period, which can be computed by

$$P = \Omega(t_c) / (S(t_c - t_s)). \qquad (5)$$

At time t_c, $\Omega(t_c) = \Lambda(t_c) = A$, which is the number of total arrival shipments, so it is a constant number regardless of change in the completion time. However, the number of shipments that could be processed, $S(t_c - t_s)$, depends on the completion time and the start time. To keep the idle time as small as possible, the next arrival of shipments should be ready to be processed when the previous arrival has been finished so that the sorting system will be fed throughout the process. This suggests that a queue of work could be desirable to maximize the productivity. One way of forming a queue is to delay the start time so that shipments that arrive before the start time will be queued. Nevertheless, delaying the start time can directly lengthen the completion time. Thus, the start time should be carefully determined to balance the tradeoff between productivity and completion time.

Increasing the sorting rate could also be an option in improving the system performance. Suppose the departure schedule is predetermined, meaning that there is no benefit of earlier completion time.

From eq. (4),

$t_c = t_s + I + A/S$, where I = idle time.

As t_c and A are constant, increasing S allows us to delay the start time and/or lengthen the idle time by $A/S – (t_s + I)$. t_s and I are dependent because, as t_s increases, I will be reduced because of work queuing. The interaction of the two variables mutually benefits the system in that delaying the start time allows the cutoff times at preceding terminals to be extended and reduced idle time possibly improves the system productivity.

However, in reality, shipment arrivals, especially for ground shipments, may not be completely deterministic as scheduled. And delayed arrivals could affect sorting processes at subsequent terminals. Thus, it may be preferred to begin the sort as soon as the first shipment arrives. In this case, it is obvious that increasing the sort rate will shorten the completion time, as shown in Figure 3.2.

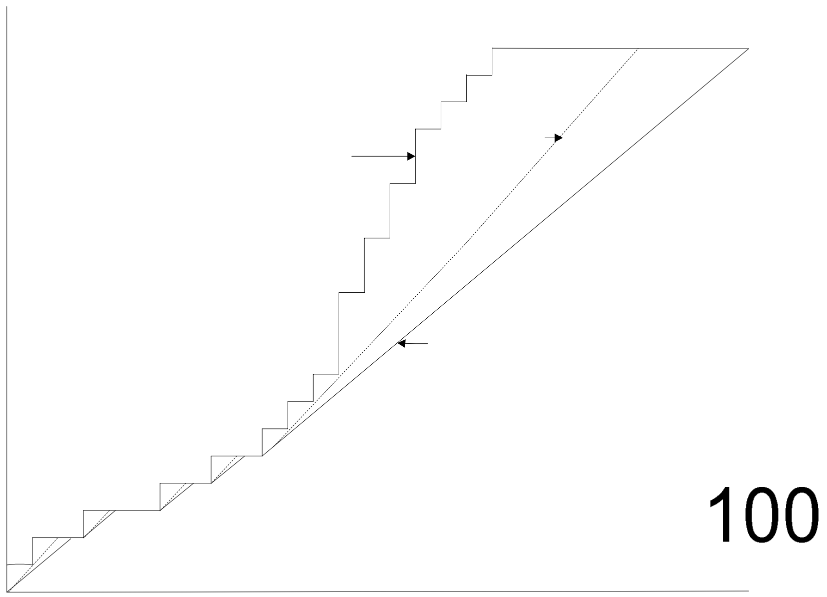

Figure 3.2. Cumulative shipment arrivals and sorting rate

Nevertheless, the increased sorting rate does not shorten the sort time when the sorting facility is not busy, specifically from hour 0 to hour 2.5. On the other hand, it causes more idle time, which results in lower productivity. Moreover, increasing the sorting rate also involves an investment in sorting equipment. And it sometimes may not be worthwhile to invest in the expansion since the sorting facility is in high demand only in peak periods of a day.

3.3.2 Role and Effects of Containerization

To enhance the system performance in terms of productivity and completion time, another strategy is to reduce the number of shipments that need to be sorted, and this is where containerization has a role, as shipments in pre-sorted containers do not need sorts. Containerization allows incoming shipments to bypass terminals and thus, the number of shipments that need sorts will be reduced, as shown in Figure 3.3.

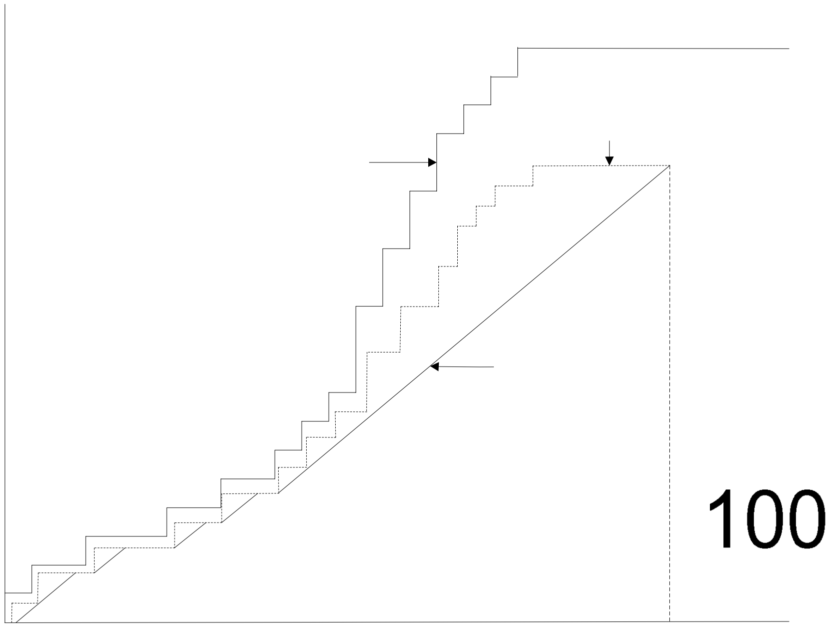

Figure 3.3. Cumulative arrival shipments with containerization

Evidently, with the same sorting rate, the completion time can be smaller as a result of the reduction of workload. However, two considerations need to be taken into account in containerization. First, containerization has much less benefit if the sorting facility is not busy; the completion time will remain unchanged. In fact, more idle time may lead to lower productivity as seen in the case of increased sorting rate. Second, containerization also incurs time and labor in the operations. This may cause delays in departures, which affects subsequent terminals. Figure 3.4 illustrates the impacts of containerization.

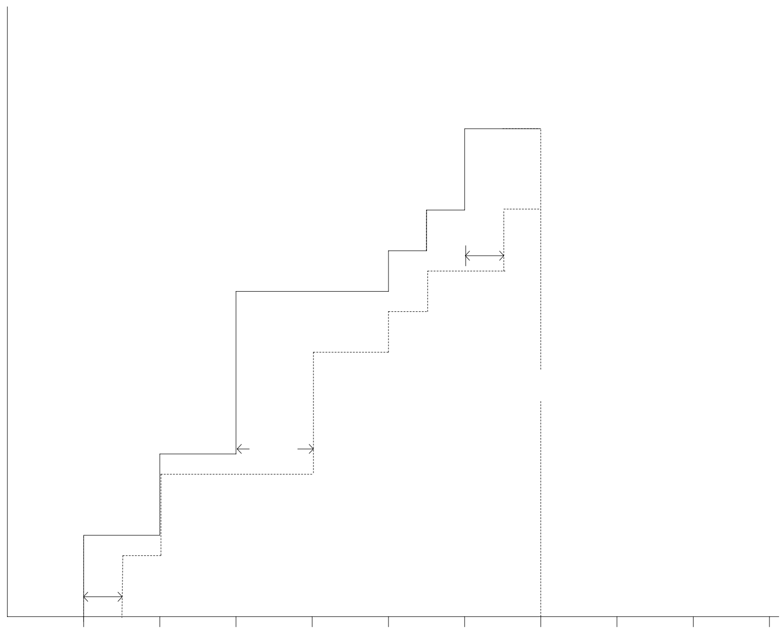

Figure 3.4. Effect of departure delay on subsequent terminal

Suppose there are six arriving airplanes at a terminal. The sorting process is completed at time t_c. From Figure 4, the full line represents cumulative arrival shipments without containerization and the dotted line represents the cumulative arrival shipments with containerization. Each step represents arrivals from an inbound airplane. Although containerization reduces the number of containers to be sorted at the terminal, it also incurs additional time and costs since packages need to be sorted into more categories and utilize more sort equipment at the preceding terminals. More importantly, the deferred completion time of the sorts will result in delays of aircraft departures which, in turn, affect aircraft arrivals at subsequent terminals. From Figure 3.5a, the numbers of containers requiring sorts of aircraft 1 and 3 are smaller. On the other hand, their arrivals are delayed, proportional to the number of containers sorted at the preceding terminal and arrived at hour 1.5, 4, and 6.5, accordingly. The effects of the reduced number of packages and the delayed time over the completion time can be illustrated as follows.

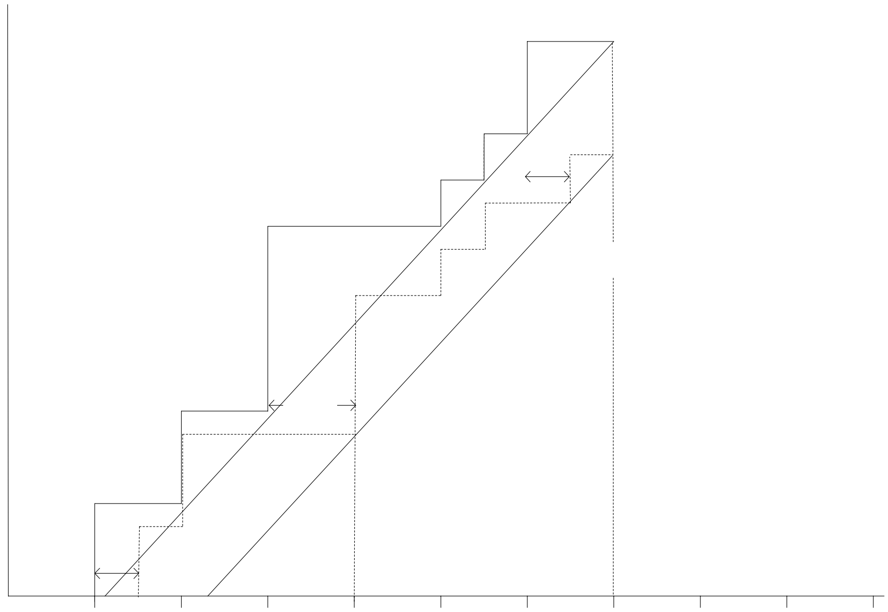

Figure 3.5a. Effects of containerization

With optimal start times, the sorts are completed at the same time, t_c. However, with containerization, the start time can be delayed and benefit the system as discussed earlier.

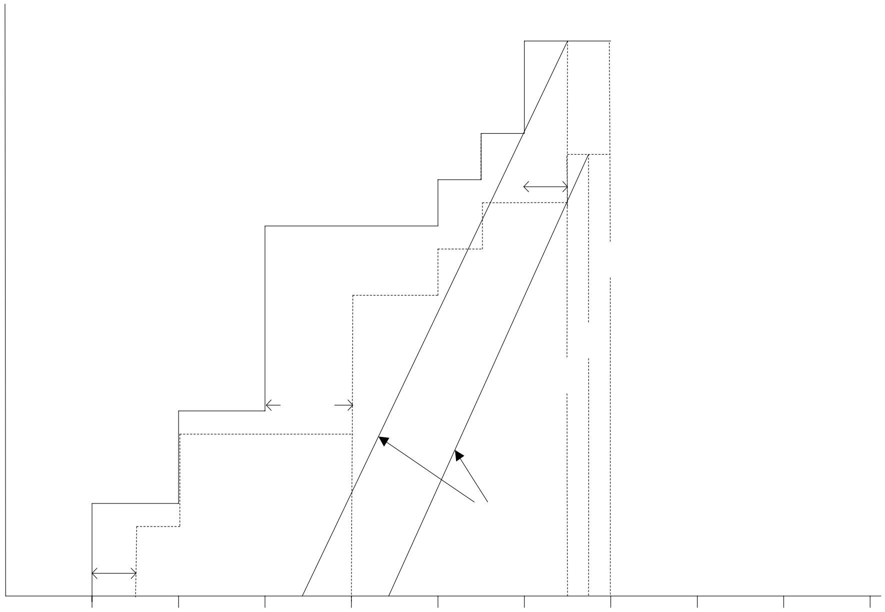

Figure 3.5b. Effects of containerization with higher sorting rate

Now, suppose the terminal holds a higher sorting rate as shown in Figure 3.5b. The same rationale can be applied to sorts with higher sorting rates. The start time of a sort with containerization can be delayed due to the reduced number of packages to be sorted. Nevertheless, with containerization, the completion time could be extended from time $t_{c,h}$ to time $t'_{c,h}$ since, by the time the last plane arrives, the sort without containerization already ends. With containerization, the aircraft arrival is delayed and, thus, the completion time is the arrival time of the last plane plus the time used in sorting, equal to hour 6.75. This occurrence results from improper containerization, leading to undesirable delays and, in this case, the reduction of workload does not benefit the system in terms of the completion time.

The relationship of the delays and the reductions to the sort rate can be illustrated as follows.

Suppose, by the time of the last arrival, all shipments have been processed or the same amounts of workload are in process. Now, considering the last arrival, let

t_d = delayed time
t_l = arrival time of the last arrival without containerization
a_l = number of shipments in the last arrival
r = shipment reduction
Without containerization, the completion time, t_c, is
$t_c = t_l + a_l / S$.
With containerization, the completion time, $t_{c'}$, will be

$t_{c'} = t_l + t_d + (a_l – r) / S$.
If we like to maintain the completion time, then $t_{c'} \leq t_c$,
$t_l + t_d + (a_l – r) / S \leq t_l + a_l / S$
Then,
$S \leq r/t_d$.

This relation shows that containerization will not delay the completion time as long as the sorting rate is less than the ratio of the shipment reduction to the deferred time. However, this restriction shall be considered together with all other arrivals since delayed time has minor effects on the completion time if the sort system is not busy by the time of arrival but the reduction of to-be-processed shipments can make savings on sorting costs.

Overall, in selecting shipments to be containerized, three factors are needed to be considered together: arrival patterns of aircraft at subsequent terminals, the sorting time incurred in performing containerization, and the sorting rates at subsequent terminals. The optimal result may yield better productivity and/or earlier completion time.

3.3.3 Benefits of Containerization

Containerization allows incoming shipments to bypass terminals and, thus, the number of shipments that need to be sorted is reduced. However, from a standpoint of sort capacity, the significance of the reduction is pronounced only when the bypass containers carry shipments that require sorts in peak periods.

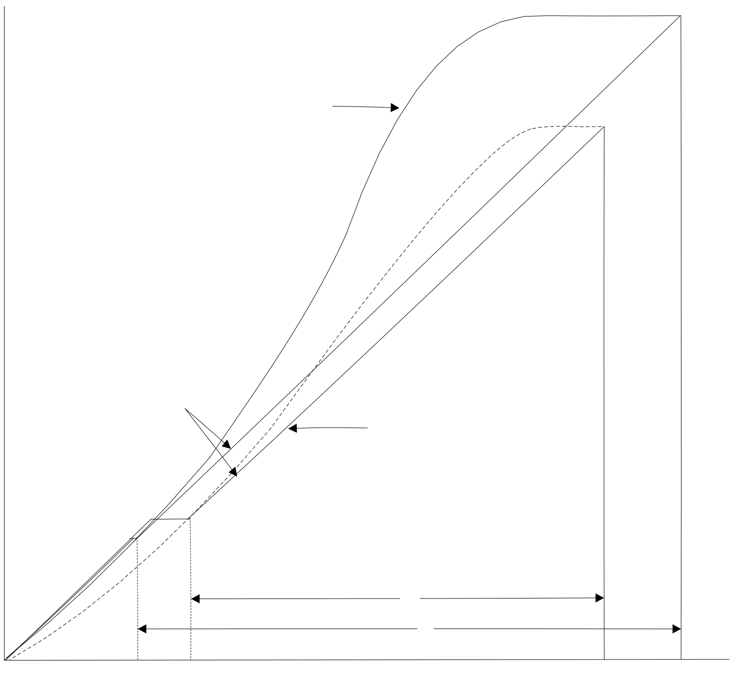

Figure 3.6a. Benefits of containerization on sort capacity

Figure 3.6a illustrates how the system can maximize benefits from containerization. Before time t = 1.8, the shipment arrival rate is lower the sort rate; thus, all the shipments can be processed as soon as they arrive. A queue is formed after time t = 1.8 since the arrival rate is greater than the sort rate. The queued shipments are processed in C units of time and the sort process ends at time t = 9. With containerization, the number of shipments that need to be sorted can be reduced. As a result, the queue is formed after time t = 2.5 and the processing time for shipments in queue can be reduced from C to C' and the completion time becomes t = 8. Before the queue is formed, the sort rate is greater than the arrival rate, meaning that incoming shipments can be processed as soon as they arrive and, thus, the completion time of the process in this non-congested period would be the same for shipments with or without containerization. From a standpoint of sorting cost, the reduction of shipments that need to be sorted may be favorable since the sorting cost is proportional to the number of shipments served but would not benefit the completion time. On the other hand, when the queue is formed as the arrival rate is greater than the sort rate, the sort capacity is inadequate to handle incoming shipments. For these congested periods, with containerization, the reduction of shipments to be processed could be more advantageous in terms of completion time since the reduced queue shortens shipment waiting time, making the process finished earlier.

However, containerization also incurs time and could result in later arrival time. Figure 3.6b illustrates delayed arrival times at a terminal that results from containerization at a preceding terminal.

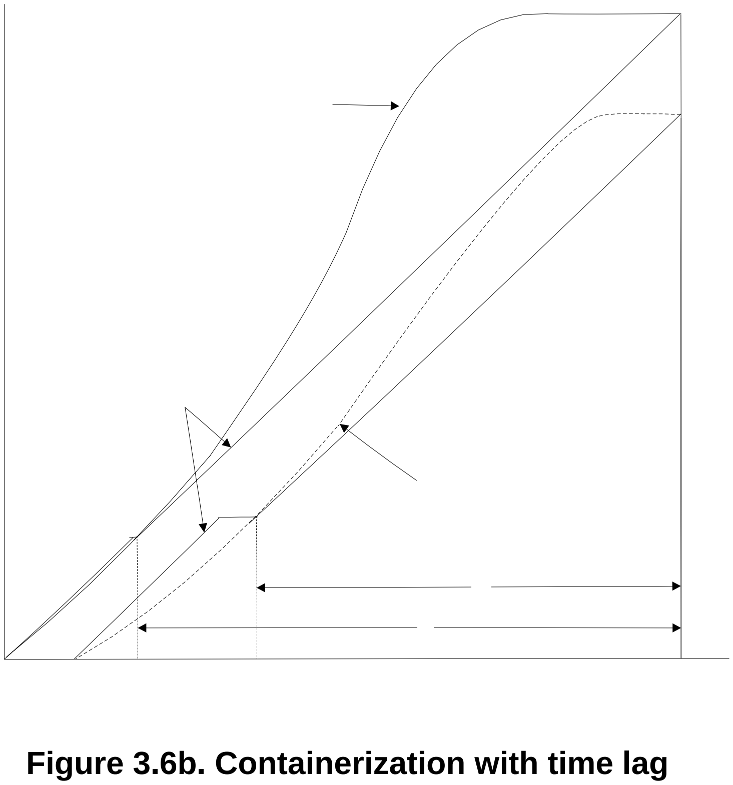

Figure 3.6b. Containerization with time lag

As shown in Figure 3.6b, arrival shipments arrive one hour late because of time incurred in containerization. Although containerization could reduce the number of shipments to be sorted, leading to shortened processing time, it may not be advantageous to the completion time due to the delay. Thus, to achieve the optimal result, aircraft arrival patterns and interactions of containerization for all terminals should be taken into account.

3.4 Implications of Net Modeling

In modeling the package transportation that accounts for sorting process, major concerns are additional elements associated with sort operations: sorting costs and capacity. The sort cost element is straightforward as we can simply add costs to processed shipments at each terminal. The sort capacity part is much more challenging since, in practice, sort capacity is measured in terms of processing rate and, thus, for a time interval, the number of shipments that can be processed at a terminal may vary, depending on arrival patterns and queuing process as discussed earlier. The sort capacity element could be captured into three classes of modeling complexity as follows.

3.4.1 Single Aggregate Capacity

In this model, sorting process is simplified by assuming that arrival schedule is predetermined. As a result, during a time interval which a fleet of aircraft arrive at a terminal, the number of shipments that can be processed is given; this number is the sort capacity for the terminal. As the sorting capacity is measured in number of shipments, the model does not take shipment arrival times and queuing dynamics into account. This model serves as the core model from which extensions are made.

3.4.2 Capacity by Time Period

As discussed in section 3.4, improper containerization not only has little benefit to the system but possibly worsens the system performance. The major consideration is determining which containers should take higher priority over some others to yield the best system performance. This element of the process could be handled by segmenting the sorting period into two time intervals – peak and non-peak periods. Here, the fixed schedule assumption still holds. Under the assumption, we can determine which shipments will arrive at the next terminal in peak periods and these shipments, if containerized, will yield better objective values than other shipments that arrive in non-peak periods. Each time period holds a constant sort capacity with the likelihood that the peak periods have much more limited sort capacity.

3.4.3 Detailed Model Considering Each Arrival Flight

The first two models assume that arrival and departure schedules are predetermined and sorting capacity is constant in terms of number of shipments that could be processed. Thus, the queuing process is not taken in to account. To make the problem more realistic, we may allow arrivals and departures to be altered by time incurred in containerization. This would tremendously complicate the problem since changes in schedules have effects on subsequent terminals over the whole network in terms of arrival patterns which, in turn, affects decisions in containerization. The interactions among all connected terminals will bring on several additional issues. Another aspect of this class of modeling is sorting capacity that is measured in number of shipments per time unit, meaning that the number of incoming shipments could exceed sort capacity at a time interval but a queue would form and delay the completion time. Whether sort period is peak or non-peak depends on congestion of sorting facility, which also interacts with shipment arrivals and containerization time at the preceding terminals. These interactions among shipment arrivals, processing times, departure times, and time in containerization significantly increase the problem complexity. However, this class of modeling would capture the characteristics of sorting process to the greatest extent.

In this research, the first two classes of the problem will be examined. The detailed model is a topic for future research.

4. Containerization model

4.1 Introduction

In this research, we focus on package routing assignment where routes are defined by

1) A sequence of terminals that a shipment visits, starting from an origin terminal and ending at a destination terminal.
2) A container group that the shipment is carried on each segment between connected terminals.

The decision is to assign shipment routes to meet service requirements while satisfying vehicle capacity and sorting facility capacity. We exclude local pickup and delivery operations, so an origin terminal is a local terminal where the long haul transportation is started. Demand collected from local pickup operations is given for each origin terminal. Each shipment is associated with an origin-destination pair and a service class. The service class is used to identify commitment time of the shipment. Total routing costs are composed of sorting costs and transportation costs of routing all shipments.

In this section, an air network is mainly considered and used as a core model for expanded network. The assumptions we made are as follows:

1) Aircraft schedules are pre-determined. The aircraft schedule includes arrival time, departure time, and origin-destination pair for each flight.

2) Sorting capacity of each sorting facility is constant. With pre-determined aircraft schedules, each sorting facility has a limited time window to perform sorting operation. Thus, the sorting capacity is assumed to be fixed and measured in number of shipments.

3) The number of containers is unlimited. There is no restriction on the number of containers used to carry shipments on each flight as long as aircraft capacity is satisfied.

4) Customer demand is known at every origin terminal.

5) Shipment size is the same for all shipments. We assume that volume and weight of all shipments are identical so as to eliminate complexity of packing problems in containers and additional aircraft operating costs that may result from weight of shipments.

6) Container size is the same for all containers. Each container has the same capacity measured in number of shipments. This assumption also allows us to define aircraft capacity as number of containers.

In a later stage of the research, some of these assumptions are relaxed to make the model better fit real-world situations.

4.2 Model Elements and Mathematical Formulation

4.2.1 Model Elements

As mentioned earlier, we define a route as a path specifying a sequence of terminals a shipment visits and a set of container groups carrying the shipment on each flight. Each arc between two connected nodes represents a container group in a certain flight. An original network consists of a set of nodes and a set of flights connected to these nodes. This network needs to be modified to be compatible with *routings* in our definitions. To make the modification, we created additional arcs, each representing the sequence of terminals that accounts for the sorting operation at each terminal. A container group on each flight identifies whether or not shipments in the container group are sorted at the next airstop. Two cases may occur for each container group.

1) If a container is sorted, shipments in the container can be routed via several alternatives according to the number of flights departing the sorting terminal.

2) If a container bypasses the next airstop, all containers in the container group must be directed to the same next terminal.

Note that for arcs representing bypassing container groups, certain terminals after the next visits must be specified.

The existing network is a subset of the modified network that includes the same set of nodes and arcs representing container groups. Container groups that are to be sorted will utilize sorting facilities at the subsequent terminals, which provide limited capacities and sorting costs are incurred. Besides sorting capacity, we must also satisfy limits on aircraft capacity and container capacity. The solution specifies shipment routes that minimize total relevant costs comprising sorting costs and shipment routing costs. Sorting costs are assumed to be constant for each terminal and measured in dollars per shipments. Shipment routing costs depend on the number of shipments on arcs since operating costs such as fuel are increased for aircraft with more shipments.

4.2.2 Mathematical Formulation

The problem can be formulated as follows.

Parameters

K = set of all commodities *k* identifying o-d pair and service class.

Q(k) = set of all possible shipment paths *q* for commodity *k*. A path *q* is a set of connected arcs in the modified network, which is identical to a *route* in our definition.

I = set of terminals i.

A(i,j) = set of flights *a* between connected airports *i* and *j*.

$G(a_{ij})$ = set of all container groups *g* on flight a_{ij}.

$M(a_{ij})$ = capacity of flight a_{ij} in terms of number of containers.

d^k = demand of commodity *k*.

S(i) = sorting capacity of terminal *i*.

SC^{k}_{q} = Sorting cost for sorting one unit of commodity *k* on path *q*.

RC^{k}_{q} = Shipment routing cost of flowing one unit of commodity *k* on path *q*.

B = container capacity in terms of number of shipments.

$u^{q}_{a_{ij},g}$ = the binary parameter specifies the association of arc *a, g* on path *q*. Each arc is denoted by flight *a* and container group *g*. If arc *a,g* is included in path *q*, then $u^{q}_{a_{ij},g}$ = 1; otherwise, 0.

w_{iq} = the binary parameter specifies the association of paths q with sorting terminals i. If terminal i is involved with path *q*, then $u^{q}_{a_{ij},g}$ = 1; otherwise, 0.

Decision Variables

$f^{k}_{q} =$ flows of shipment *k* on path *q*.

$c_{a_{ij},g}$ = flows of container on arc *a*a_{ij},*g*, specifying container group *g* on flight a_{ij}.

Formulation

$$\text{Min. } \sum_{k\in K}\sum_{i\in I}\sum_{q\in Q(k)} SC^{k}_{q} w_{iq} f^{k}_{q} + \sum_{k\in K}\sum_{a_{ij},g}\sum_{q\in Q(k)} RC_{a_{ij},g} u^{q}_{a_{ij},g} f^{k}_{q}$$

$$\text{s.t. } \sum_{g\in G(a_{ij})} c_{a_{ij},g} \le M(a_{ij}) \qquad \forall a \in A \qquad (1)$$

$$\sum_{k\in K}\sum_{q\in Q(k)} u^{q}_{a_{ij},g} f^{k}_{q} \le Bc_{a_{ij},g} \qquad \forall a \in A, \forall g \in G \qquad (2)$$

$$\sum_{q\in Q(k)} f^{k}_{q} = d^{k} \qquad \forall k \in K \qquad (3)$$

$$\sum_{k\in K}\sum_{q\in Q(k)} w_{iq} f^{k}_{q} \le S(i) \qquad \forall i \in I \qquad (4)$$

$$f^{k}_{q} \ge 0 \qquad \forall q \in Q(k), \forall k \in K$$

$$c_{a_{ij},g} \in I \qquad \text{I = set of integer numbers, } \forall a_{ij} \in A(i,j), \forall g \in G(a_{ij})$$

The objective function is the sum of penalty costs from delayed deliveries and transportation costs, which are proportional to the number of shipments on arcs. The aircraft capacity constraint (1) limits the number of containers on each aircraft. The container capacity constraint (2) ensures that the number of shipments in each container group does not exceed capacity of the container group. The demand constraint (3) requires that the sum of shipment flows of each commodity must be equal to the demand. The limit on sorting capacity of each terminal is modeled by the sorting capacity constraint (4). Shipment flows must equal or exceed zero as forced by the non-negativity constraint. Container flows must be non-negative integer.

4.3 Input Generation

The network we consider consists of ground and air layers. The air component comprises airports and air flights that connect the airports while the ground component consists of local terminals that provide feeder services to airports. These two layers are linked by truck routings of the ground layer.

To test our algorithms, sample data were generated based on the U.S. territory and real data of regional population. Terminal location information is based on the real network of a parcel carrier company. Based on an estimated number of incoming shipments per day, sorting capacity of each terminal is a deterministic number in terms of number of shipments. Demand is based on the population in the region where the terminal is located. Network connectivity is based on terminal types. Arc costs and sorting costs are approximated based on empirical data.

In this section, we will discuss how a sample dataset is generated in detail, including

4.3.1 Terminal locations and number of terminals
4.3.2 Network connectivity
4.3.3 Demands
4.3.4 Arc costs and capacities
4.3.5 Sorting costs and sorting capacities.

4.3.1 Network Connectivity

Figure 4.1 illustrates the network connectivity of the system. Two metropolitan areas are exhibited in a region. Each metropolitan area is served by a major airport (e.g. Los Angeles). The eclipse area represents the area served by a local airport (e.g., Burbank, Long Beach). The minimal unit in this system is the local terminal, which serves as a ground feeder to nearby airports of any type. Serving as air feeders, aircraft from rural airports are not managed for long distance routings. A rural airport (e.g., Palm Springs) is served by larger nearby airports, which could be major airport, regional hub, or a national hub. A local airport is only connected to regional hubs (e.g. Ontario) or national hub (Louisville) within moderate distance. A major airport could be ground fed from local terminals or air fed by rural airports, and is linked to the nearest regional hub and the national hub. In addition, it may also be connected to other major airports (e.g., Chicago), when demand between the city pair is large enough. A regional hub is connected to all types of terminals within the region served and may also be linked to other regional hubs, depending on demands. The national hub serves all regional hubs as well as major airports across the country and is also connected to nearby local airports, rural airports, and local terminals.

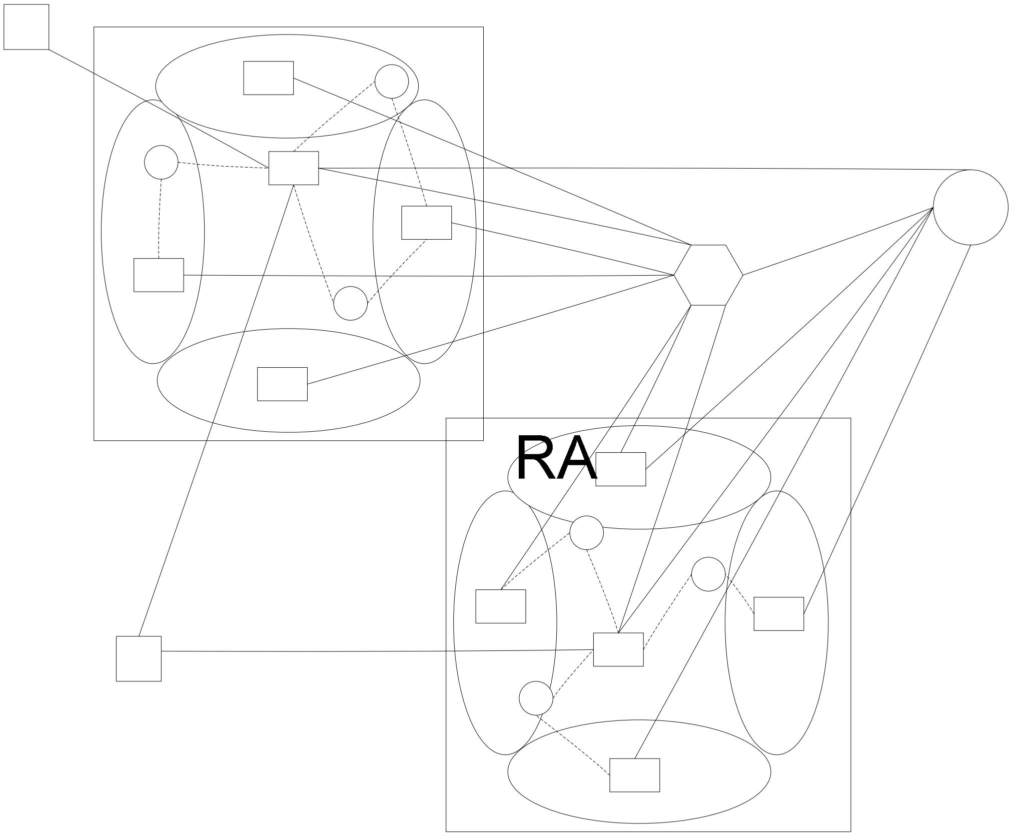

Figure 4.1. Terminal connectivity

Note:
LT – Local terminal
RA – Rural Airport
LA – Local Airport
MA – Major Airport
RH – Regional Hub
NH – National Hub

——— Air routing
------- Truck routing

	LT	RA	LA	MA	RH	NH
LT	T	NA	T	T	T	T
RA	NA	NA	NA	A*	A*	A*
LA	T	NA	NA	NA	A*	A
MA	T	A*	NA	A**	A*, A**	A
RH	T	A*	A*	A*, A**	A	A
NH	T	A*	A	A	A	NA

Table 4.1. Network Connectivity

Note:

NA - Not applicable.
T - Truck routings.
A - Air routings.
A* - Air routings, available if the distance between terminals is relatively close within the same region.
A** - Air routings, available if demand between the city pair is large enough.

4.3.2 Terminal Locations and Number of Terminals

In test problems, terminal location data were generated to replicate locations of major terminals of UPS.

Air Layer

There are five types of airport terminals in the network.
National Hub is the largest terminal in the network, where shipments from all locations are sorted, exchanged, and transported to their destinations. The national hub for UPS is located in Louisville, KY. *Regional Hubs* serve as exchange points in their region. UPS has six regional hub airports located in Philadelphia, PA; Dallas, TX; Ontario, CA; Rockford, IL; Columbia, SC; Hartford, CT. *Major Airport*s are large airports in a major city that serves a large population. *Local Airports* serve aircraft in a large area in addition to a major airport. *Rural Airports* acts as feeders from rural areas to other larger airports.

In a sample dataset, national hubs and regional hubs are based on real geographical locations mentioned above, whereas major airports, local airports, and rural airports are classified by population of metropolitan areas where the terminals are located, as shown in Table 4.2.

Airport Type	Population of area served
Major	>3,000,000
Local	300,000-3,000,000
Rural	100,000-300,000

Table 4.2. Airport Types and Populations

Ground Layer

Local terminals are service centers where packages are colleted and trucks are dispatched. Local terminals are located around major terminals in metropolitan areas across the country. The number of local terminals is proportional to the size of metropolitan areas.

4.3.3 Demands

Rodrigue, J-P *et al.* [2004] demonstrates how the gravity model is exploited to generate demands between city pairs. The gravity model is widely used in the transportation area of research because it offers a good application of interactions of freight flows. This model is similar to Newton's gravity model in that it is proportional to their mass, which is reflected in populations in transportation area, and inversely proportional to their respective distance.

The formulation of the gravity model is applied as follows.

$$T_{ij} = k(P_i^{\lambda} P_j^{\alpha} / d_{ij}^{\beta})$$

where,
T_{ij} = Demand from origin city o to destination city d.
k = Adjustment factor.
p_i = Number of populations at origin city i.
p_j = Number of populations at destination city j.
d_{ij} = Distance between cities i and j.
λ = Potential to generate demands.
α = Potential to attract demands.
β = Transport friction.

In our input generation, we use $\lambda = \alpha = 2/3$ and $\beta = 1/8$ to acquire reasonable demands between city pairs in practice.

4.3.4 Arc costs and capacities.

Transportation costs are computed based on travel times between city pairs. Travel times are proportional to distances, which are calculated from geographical locations. The costs are measured in terms of dollars per shipment.

Arc capacities are calculated from demands to assure that they are sufficient to serve all demands.

4.3.5 Sorting costs and capacities.

Sorting costs are assumed to be fixed costs per shipment for each location. Larger terminals incur lower costs per shipment due to economy of scale. Sorting capacities are proportional to population of cities where terminals are located.

4.4 Summary

In this section, a prescriptive model under a set of assumptions is presented along with the inputs for test problems. The major differences between the containerization model and a conventional multi-commodity flow model are inclusion of container capacity for aircraft and sort capacity for terminals as well as sorting costs that are reflected in the objective values. These added elements significantly complicate the problem due to a large number of integer-restricted variables. Moreover, the problem we consider is large-scale. In the next section, we will show how these model elements are represented in a modified network, along with solution approaches and experimental results.

5. Optimization Methodology

In this section, problem solving approaches and experimental results are presented. We first show how to modify the existing network to capture the elements of sorting operations. Based on this modified network, a schema incorporating column generation to solve an LP relaxation is proposed to solve large scale problems. To find MIP solutions, three heuristic approaches are developed as well as a lower bound used to evaluate the performance of the heuristic approaches. The first heuristic approach is based on a greedy algorithm in which the highest sort savings are chosen iteratively. The second approach exploits forcing constraints to tighten the LP relaxation and acquire feasible MIP solution. The third approach is the combination of the first two approaches. Experimental results of the developed heuristics are evaluated by percentage closeness to lower bound and computation time. Sensitivity analysis for affected factors is also presented.

5.1 Proposed Method

The optimization method can be segmented into 3 phases as follows.
5.1.1 Constructing a new network.
5.1.2 Solving the LP relaxation.
5.1.3 Finding IP solution.

5.1.1 Construct a new network

To account for sorting operations, artificial arcs are introduced and added to an original network. These artificial arcs represent bypass container groups that indicate sorts at intermediate terminals visited by shipments. Flows on each arc represent the number of each commodity as well as the number of containers.

Consider the example network with actual flights a, b, c, and d connecting terminals A, B, C, D, and E, as depicted below.

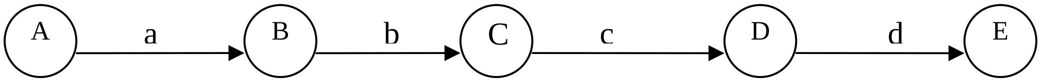

Figure 5.1. Example network

A set of artificial arcs is created to capture sorting at intermediate terminals. Each artificial arc represents a container group on connecting flights and also indicates terminals bypassed by the container group. The entire set of arcs in the network comprises a set of actual arcs which are physical links representing non-bypass container groups on the flights and a set of artificial arcs that represent bypass container groups. These artificial arcs are connecting flights that indicate 1) flights that route container groups and 2) intermediate terminals bypassed by container groups. On each arc, two sets of variables are used to indicate two types of flows –1) an integer variable of container group associated with a flight or connecting flights and 2) a set of continuous variables of all commodity flows using the container group. All possible paths for each commodity can be composed of a set of actual arcs and/or artificial arcs that link its origin and destination. The network representation is depicted in Figure 5.2.

Given a commodity with origin A and destination E, if the commodity is routed via path ABCDE, we account for sorting operations at each terminal by creating artificial arcs as follows.

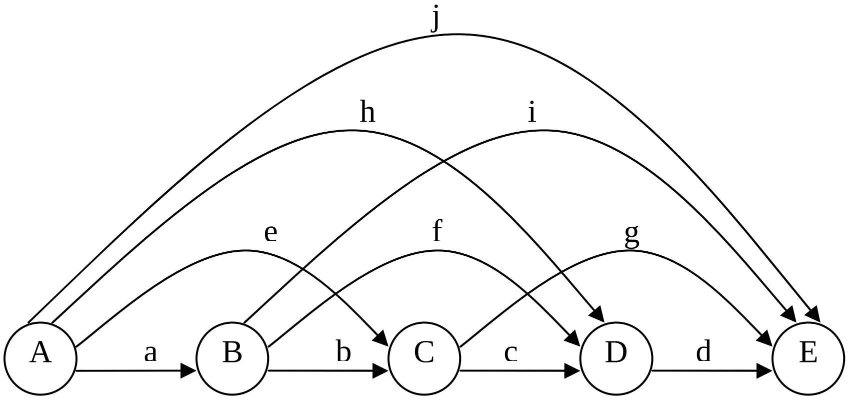

Figure 5.2. Modified network

From the modified network, the possible routings are 1) a-b-c-d, 2) e-c-d, 3) a-f-d, 4) a-b-g, 5) e-g, 6) h-d, 7) a-i, 8) j. With this network representation, all the routings use the same connecting flights a-b-c-d but with different sorting involvements. For example, routing via path e-c-d indicates that flows on this path originate from A, bypass sorting operations at terminals B, enter sorting operation at C and D, and arrive at its destination E. However, the shipments in fact are routed via flights a, b, and c, which are actual flights between terminals. Arc e represent using connecting flights a-b without entering sorting operations at terminal B.

5.1.2 Solving the LP relaxation

In our model, shipment flow variables are continuous numbers while container flow variables are restricted to integer values. Consequently, the linear programming problem we are considering is a mixed integer programming (MIP) problem. MIP problems are difficult to optimize because they require considerable computational time to find the optimal combination of specific integer values and the corresponding set of continuous variables. Moreover, the number of integer variables increases exponentially with the size of the problem. A method to solve the model is to relax the MIP by omitting all of the integer restrictions. Then, the initial solution obtained from the LP relaxation is set as a lower bound and search strategies such as branch-and-bound or cutting planes are applied to find the optimal or acceptable solutions of the MIP problem.

Consider the first two constraints of the problem formulation,

$$\sum_{g \in G(a_{ij})} c_{a_{ij},g} \le M(a_{ij}) \qquad \forall a \in A$$

$$\sum_{k \in K} \sum_{q \in Q(k)} u^{q}_{a_{ij},g} f^{k}_{q} \le Bc_{a_{ij},g} \qquad \forall a \in A, \forall g \in G$$

If the integer restrictions are omitted, we can merge these two constraints.

$$\sum_{k \in K} \sum_{q \in Q(k)} \sum_{g \in G(a_{ij})} u^{q}_{a_{ij},g} f^{k}_{q} \le BM(a_{ij}) \qquad \forall a \in A, \forall g \in G \qquad (5)$$

The aircraft capacity now is measured in number of shipments. The left-hand-side is the sum of shipment flows of all commodities on aircraft a_{ij}.

Thus, the problem formulation for the LP relaxation becomes as follows.

Formulation

Min. $\sum_{k \in K} \sum_{i \in I} \sum_{q \in Q(k)} SC^{k}_{q} w_{iq} f^{k}_{q} + \sum_{k \in K} \sum_{a_{ij},g} \sum_{q \in Q(k)} RC_{a_{ij},g} u^{q}_{a_{ij},g} f^{k}_{q}$

s.t.

$$\sum_{k \in K} \sum_{q \in Q(k)} \sum_{g \in G(a_{ij})} u^{q}_{a_{ij},g} f^{k}_{q} \le BM(a_{ij}) \qquad \forall a \in A, \forall g \in G \qquad (6)$$

$$\sum_{q \in Q(k)} f^{k}_{q} = d^{k} \qquad \forall k \in K \qquad (7)$$

$$\sum_{k \in K} \sum_{q \in Q(k)} w_{iq} f^{k}_{q} \le S(i) \qquad \forall i \in I \qquad (8)$$

$$f^{k}_{q} \ge 0 \qquad \forall q \in Q(k), \forall k \in K$$

Column generation

For large scale problems, it is impractical to put all variables into an MIP solver. Column generation is an approach that can be applied to solve the LP relaxation. With column generation algorithm, not all variables need to be included in the constraint matrix. Rather, we consider only a subset of a large LP. For each iteration, we determine if any columns can enter the basis by computing their reduced costs. If no column has negative reduced costs, the optimal solution of the subproblem, called the restricted master problem (RMP), is optimal for the LP relaxation.

Solving the pricing problem

Our formulation is similar to the multi-commodity flow path-based formulation. The difference is that we have an additional constraint of sorting capacity (8), and that we have path costs that account for sorting costs, $SC_q^k w_{iq} f_q^k$, in the objective function besides the arc costs, $RC_q^k f_q^k$.

Let
- $\pi_{a_{ij}}$ = dual variables associated with constraints (6).
σ^k = dual variables associated with constraints (7).
- δ_i = dual variables associated with constraints (8).

The dual of the problem is as follows.

(D) Max $\sum_{a_{ij}} BM_{a_{ij}}(-\pi_{a_{ij}}) + \sum_k d^k \sigma^k + \sum_i S_i(-\delta_i)$

st.

$$\sum_{a_{ij}}\sum_g u_{a,g}^q f_q^k(-\pi_{a_{ij}}) + \sigma^k + \sum_i w_{iq}(-\delta_i) \le \sum_i SC_i\, w_i + \sum_{a_{ij}}\sum_g RC_{a_{ij},g} u_{a,g}^q \quad ,$$
$$\forall q \in Q(k), \forall k \in K \qquad (9)$$
$$-\delta_i \ge 0$$
$$-\pi_{a_{ij}} \ge 0$$
$$\sigma^k = \text{unrestricted}$$

From our formulation, the reduced cost of column q for commodity k is denoted as follows.

$$\overline{c_q^k} = [\sum_i SC_i\, w_i + \sum_a \sum_g RC_{a,g} u_{a,g}^q] - [\sum_a \sum_g (-\pi_{a_{ij}}) u_{a,g}^q + \sigma^k + \sum_i (-\delta_i) w_{iq}]$$
$$\forall q \in Q(k), \forall k \in K \qquad (10)$$

The optimality condition holds when the reduced costs of all k, $\overline{c_q^k}, \ge 0$. From the reduced cost equation (10), the optimality condition can be rewritten as follows.

$$\sum_a \sum_g (RC_{a,g} + \pi_{ij}) u_{a,g}^q + \sum_i (\delta_i + SC_i) w_{iq} \ge \sigma^k \qquad \forall q \in Q(k), \forall k \in K \qquad (11)$$

This pricing problem (11) can be solved by a shortest path algorithm on a modified network. The network can be modified as follows.

(1) For each arc, a$_{ij}$, impose arc cost equal to $(RC_{a,g} + \pi_{ij})$.

(2) For each arc departing from node i, add arc cost ($SC_i + \delta_i$).

Then, the shortest path algorithm can be used to find paths that price out for each commodity k.

If a set of paths q* satisfy the equations (11) for all commodities k, the LP is solved to optimality. For each q* that does not satisfy the optimality condition, the column q* is added to the RMP. After the LP is solved, container flows, $c_{a_{ij},g}$, on each container group can be found by computing $c_{a_{ij},g} = \sum_{q \in Q(k)} f_q^k u_{a,g}$.

The flowchart of solving the LP relaxation is follows.

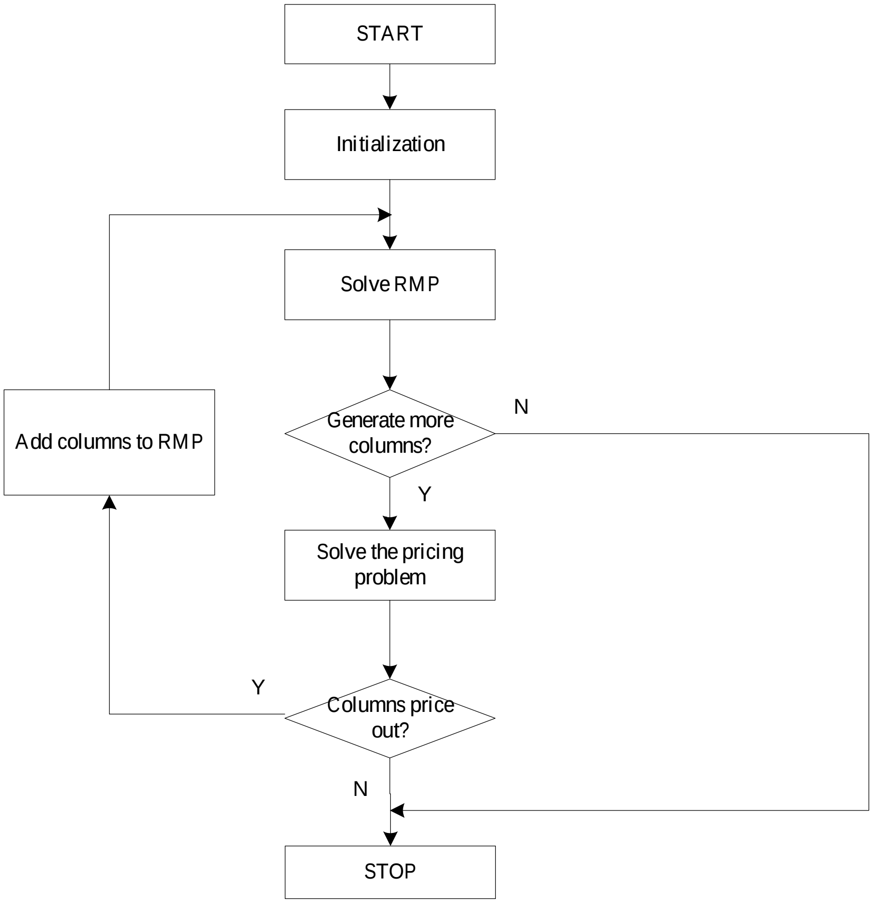

Figure 5.3. The LP relaxation

Column generation could be an approach to solve the LP relaxation for large scale problems. However, in this research, it was not needed since CPLEX can handle problems with network size we consider.

5.1.3 Finding the IP solution

5.1.3.1 Grouping Heuristic

The solutions from the LP relaxation provide very loose LP bounds because container variables are likely to bypass all intermediate terminals, resulting in small fractions for integer variables. The solution obtained from the LP relaxation is not a feasible IP solution and using this solution as a starting point to find a feasible IP solution requires intensive computation time. To acquire a decent starting point, we developed a heuristic approach called the grouping heuristic to obtain a feasible IP solution.

The underlying idea of the grouping heuristic approach is to aggregate commodities that are possibly routed in the same container groups with large savings on both transportation costs and sorting costs. The heuristic begins with solving the LP relaxation without bypass routings, assuming that sorting capacity is unlimited. In other words, we first solve a multi-commodity flow problem with sorting cost incurred at each terminal. The acquired solution provides sequences of terminals for each commodity. With this information in hand, connecting flights used for each commodity are also known. Then we assign shipments routed on the same flight to associated containers with potential savings on sorting costs. Container selections are based on savings over sorting costs of containers. The assignments are repeated iteratively as assigned shipments are excluded and parameters are updated at each iteration. The grouping proceeds until no more flows can be aggregated. Then, flight capacities are examined to find possible container assignments that could make sort savings by virtue of unused flight capacities. At this step, to make more savings under limited flight capacity, we prioritize this set of container groups by examining $\sum_i s_i / \sum_{a,g} w_{a,g}$ where s_i = sorting cost at bypassed terminal i. $w_{a,g}$ is wasted flight capacity resulting from unused container capacity. This ratio represents potential sort savings traded-off with wasted flight capacities. At each iteration, a container group with max $\sum_i s_i / \sum_{a,g} w_{a,g}$ has the highest priority. The heuristic terminates when no further savings can be made. This heuristic will be referred to as the GH for the following sections.

The GH procedure is described as follows.

Grouping Heuristic

1. *Finding shipment routings*. Exclude all bypass routings and solve the LP relaxation with large sorting capacities for all nodes.

2. *Large flow assignments*. For all commodities k whose flows exceed one container capacity,

    2.1. Assign all commodities k to associated container groups along their routing paths q.

    2.2. Update sorting capacity that accounts for container bypass at each node.

3. *Grouping flows to full containers*. Enumerate possible commodity flows that could be contained within each container group, based on flows on each arc. At this step, we only consider container groups whose sums of possible flows exceed one container capacity. Container group selection is prioritized based on sort savings. Among these container groups,

3.1. Select container group $c_{a,g}$ that makes the largest sort savings.

3.2. Group smallest set of flows in the container group up to one container.

3.3. Assign the grouped flows into the selected container.

3.4. Update sorting capacity for nodes that the selected container group bypasses.

3.5. Update possible flows in other associated container groups.

3.6. Repeat steps 3.1-3.5 until no container group whose sum of flows exceeds one container capacity exists.

4. *Grouping flows to make more savings*. For the remainder of container groups,

4.1. Select container group with max $\sum_i s_i / \sum_{a,g} w_{a,g}$, where s_i = sorting cost and $w_{a,g}$ = wasted flight capacity.

4.2. Check if associated flight capacities are available for the selected container group.

4.2.1. If available, assign possible flows into the selected container group.

4.2.2. If not, repeat steps 4.1-4.2.

4.3. Update sorting capacities of nodes that the selected container group bypasses.

4.4. Update possible flows in associated container groups.

4.5. Update associated flight capacities.

4.6. Repeat steps 4.1-4.5 until no container group can be assigned.

After completing the heuristic procedure, it is possible that the solution obtained may not be a feasible solution because sorting capacities are violated at some nodes. To make the solution feasible, some flows may need to be reassigned. The following procedure, called *sorting capacity adjustment*, is used to amend the solution so as to satisfy sorting capacity constraints.

Sorting Capacity Adjustment

1. Exclude grouped containers and assigned flows from the solution.

2. Solve the LP relaxation by setting the following input parameters:

2.1 Flight capacity: The remainder of flight capacities for all arcs.

2.2 Sorting capacity:

2.2.1. For nodes with exceeded sorting capacities, remaining sorting capacities after grouped flows removal.

2.2.2. For all other nodes, unlimited capacities.

3. Apply the *Grouping heuristic*.

The GH proceeds based on a greedy scheme in that the best possible option among candidate container groups is selected iteratively. Sort savings are made based on the routings obtained from solving the LP relaxation without bypass arcs. The selection criterion is sorting costs saved on bypass container groups so container groups that bypass more nodes tend to be chosen first. Flight capacity constraints are satisfied as long as commodity flows are assigned to full containers. When possible flows in container groups are less than a full container, associated flight capacities will be examined. Since grouping is based on the acquired solution, it is anticipated that the computation time will be very fast and comparable to solving the LP relaxation alone. Also, it is likely that the solution quality will be satisfactory if the optimal solution is weighted in favor of transportation costs.

The overall flow of the GH is depicted in Figure 5.4.

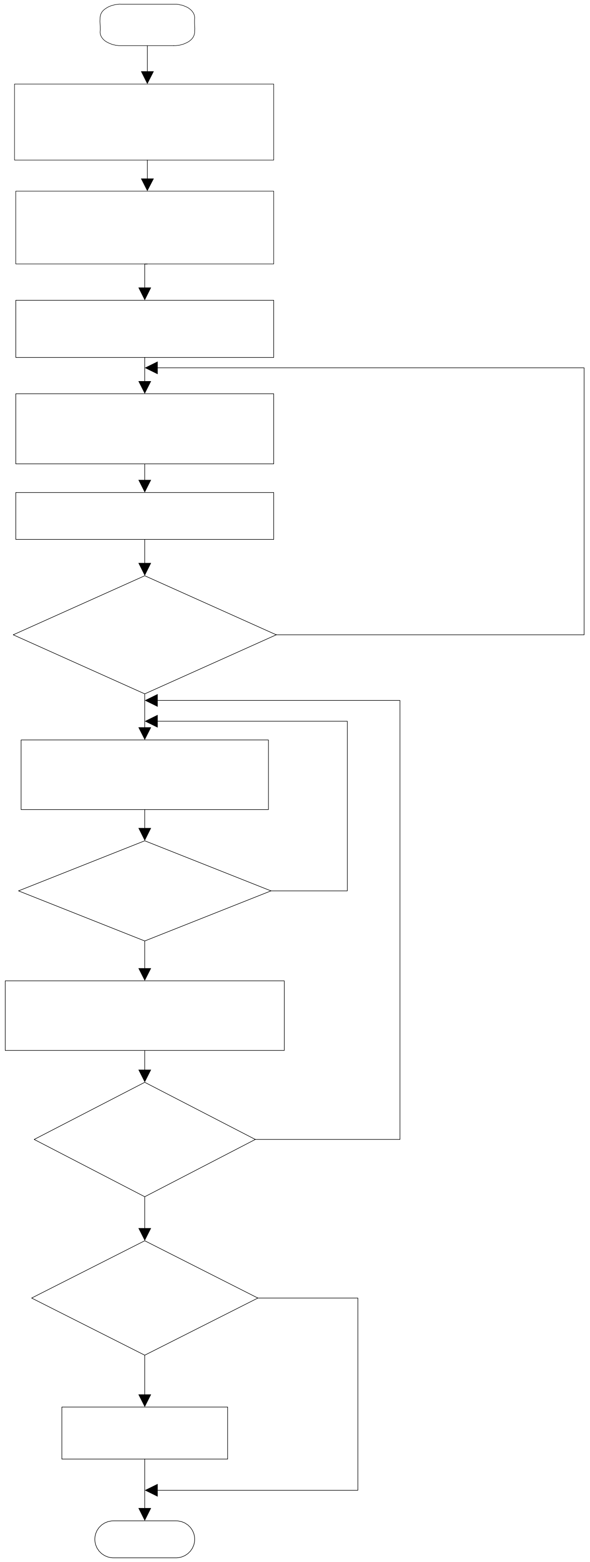

Figure 5.4. Grouping Heuristic

5.1.3.2 Forcing Constraint Heuristic

The LP relaxation for our problem can be viewed as the original problem with varying container capacity, ranged from zero to aircraft capacity limit. As we omit integer restrictions, it is anticipated that container flow variables of solutions to the LP relaxation will become smaller and smaller fractions as the number of iterations increases. This essentially means that the LP solution may likely provide a very weak bound to the IP solution and also be a poor starting solution for the IP solution phase. To overcome this drawback, we include a set of forcing constraints as used in Barnhart [1998] to force the container variables to take larger values. Barnhart [1998] proposed a path-based formulation for the constructing railroad blocking problem. In the paper, additional constraints are added to guarantee that sum of the proportion of commodity flow k on path q that contains arc a is less than or equal to the binary variable y_a that represents the use of the block associated with arc a. Their forcing constraints are denoted as

$$\sum_{q \in Q(k)} f_q^k \delta_a^q - y_a \leq 0 \qquad \forall a \in A, k \in K \qquad (9)$$

This set of constraints in fact is a disaggregation of the bundle constraints in their formulation,

$$\sum_{k \in K} \sum_{q \in Q} v^k f_q^k \delta_a^q - u_a y_a \leq 0 \qquad \forall a \in A \qquad (10)$$

The forcing constraints are redundant for the LP relaxation but to find an IP solution they effectively serve as additional cuts.

The results from their experiments show that, with forcing constraints, the LP relaxation of a test problem with 2333 binary variables can be obtained with 13 fractional variables and a feasible IP solution can be acquired within a minute. Without forcing constraints, the LP relaxation had 458 fractional variables and they were unable to obtain a feasible IP solution within 4 hours.

In our formulation, the integer variables are container group variables so the forcing constraints are applied as follows.

$$\sum_{q \in Q(k)} u_{a_{ij},g}^q f_q^k \leq c_{a_{ij},g} d^k \qquad \forall g \in G, k \in K \qquad (11)$$

The left hand side is, for each commodity, the sum of flows from different paths that share a container group, which obviously could not exceed its total demand. On the right hand side, the total demand is multiplied by associated container variable, meaning that if the container variable is zero, then no flows can be routed on this container group and if the container variable holds some value, it at least must be large enough to allocate aggregated flows from different possible paths of each commodity.

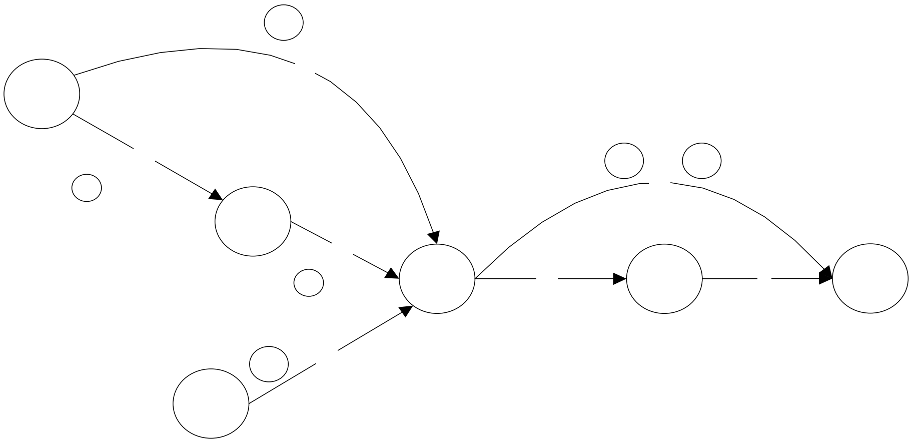

Figure 5.5. Different paths that share a container group

d16=15

Figure 5.5 depicts flows of two commodities that share a container group. Commodities d_{16} and d_{36} have the same destination, node 6. Suppose a feasible solution is that the two commodities share the same container group g. From the container capacity constraints of the original formulation, we have

1

$$\sum_{k \in K} \sum_{q \in Q(k)} u_{a_{ij},g}^{q} f_{q}^{k} \leq Bc_{a_{ij},g} \qquad \forall a \in A, \forall g \in G .$$

From the example, total flows routed on container group g are 25. Given that the container capacity is 100, $c_{a,g}$ variable for container group g will be 0.25.

d16

5

Now, if we include the forcing constraints (9),

$$\sum_{q \in Q(k)} u_{a_{ij},g}^{q} f_{q}^{k} \leq c_{a_{ij},g} d^{k} \qquad \forall g \in G, k \in K .$$

For commodity d_{16} and container group g, using the feasible solution from the original formulation, the right hand side will be 0.25*15 = 3.75. With this right hand side value, total flows of 15 could not be allocated to this container group. In other words, the container variable value of 0.25 is no longer a feasible solution under the forcing constraint. As a result, the container variable is forced to take a larger value.

Using forcing constraints, we developed a heuristic approach called the *forcing constraint heuristic*, referred to as the FCH. The underlying idea is to add forcing constraints into the LP relaxation so that flows in the solution will be large enough to be assigned to container groups, and then resolve the problem with updated input until all flows are assigned. We first begin with solving the LP relaxation with all forcing constraints then, after the solution is obtained, the nearest lower integer number of container flows that are greater than 1 container will be saved as parts of the solution and excluded from the problem. All fractional flows of containers will be resolved. The FCH is repeated iteratively until the solution no longer changes. Then we examine if any container groups can bypass by wasting some flight capacities. When no more savings can be made, all fractional flows will be assigned to associated non-bypass container groups. One advantage of the FCH over the GH is that the FCH already accounts for sorting capacity in the LP solving step; therefore, it does not require additional procedures for sorting capacity adjustment.

The FCH procedure is as follows.

Forcing Constraint Heuristic

1. *Find shipment routings.* Solve the LP relaxation with all the forcing constraints.
    1.1 If all container variables are integer, the solution is optimal IP solution.
    1.2 Do step 2, otherwise.

2. Large flow assignments.
    2.1 If the solution does not change from the last iteration, apply step 4.
    2.2 If the solution changes, assign flows in the same container group that are greater than one container capacity to the associated container group, exclude the assigned flows from the problem, update inputs, redo step 1.

3. Assignments for flows in non-full container groups.
    3.1. Select container group with max $\sum_i s_i / \sum_{a,g} w_{a,g}$, where s_i = sorting cost and $w_{a,g}$ = wasted flight capacity.
    3.2. Check if associated flight capacities are available for the selected container group.
        3.2.1. If available, assign possible flows into the selected container group.
        3.2.2. If not, repeat steps 3.1-3.2.
    3.3. Update sorting capacities of nodes that the selected container group bypasses.
    3.4. Update possible flows in associated container groups.
    3.5. Update associated flight capacities.
    3.6. Repeat steps 3.1-3.5 until no container group can be assigned.

The FCH exploits the forcing constraints to cut LP solutions that are not feasible MIP solutions. This set of constraints force fractional flows to take larger values so that they can be assigned to container groups. However, due to a large number of constraints added, the computation time is drastically increased, especially for large problems. To handle this issue, we introduce alpha value to heuristically eliminate constraints that are probably unnecessary in acquiring feasible MIP solutions so as to reduce computation time. The alpha value will be discussed in section 5.1.3.4.

The flowchart of the FCH is depicted in Figure 5.6.

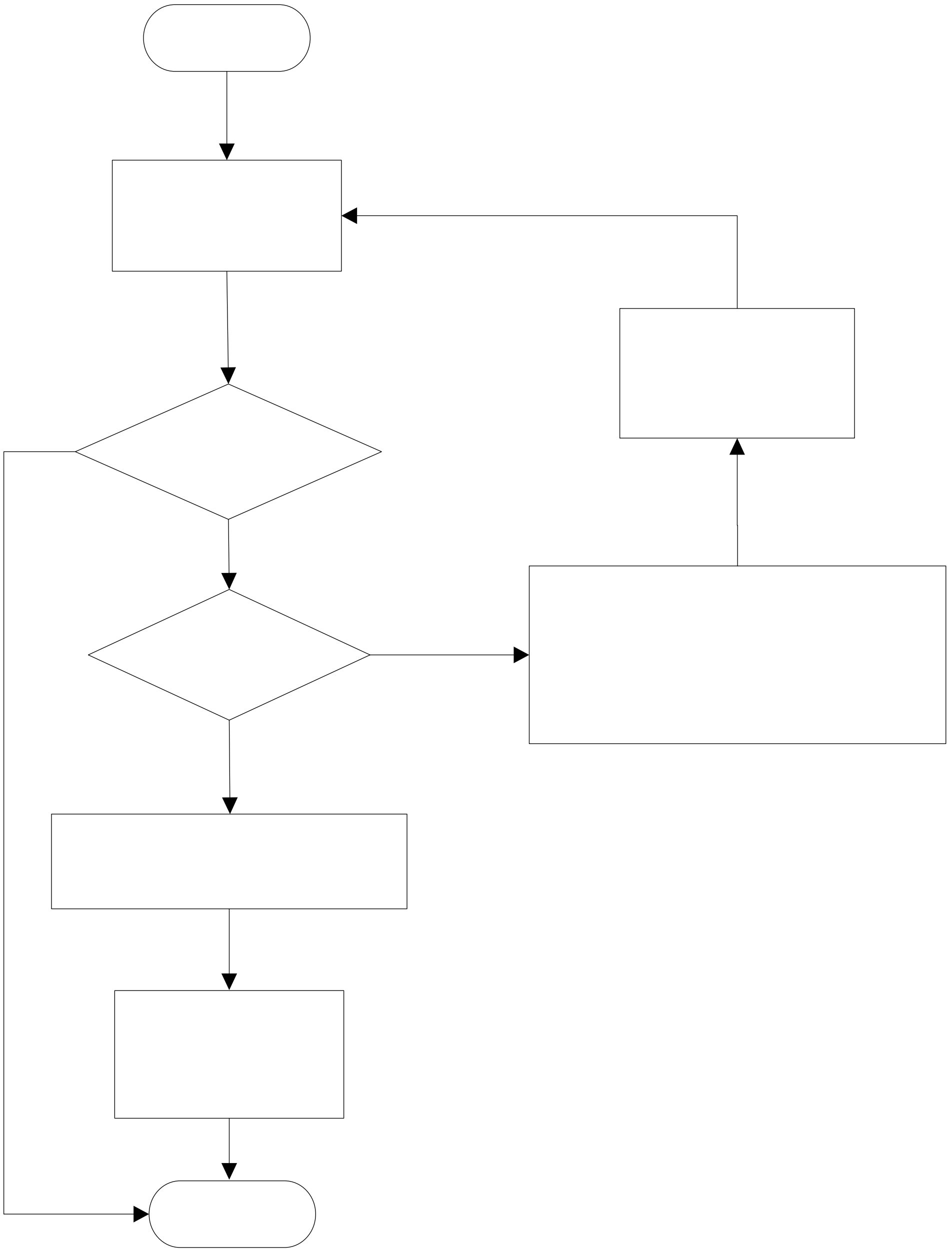

Figure 5.6. Forcing Constraint Heuristic

5.1.3.3 Combined Heuristic

The combined heuristic is the combination of the two heuristics presented in the previous sections, the GH and the FCH. The combined heuristic, referred to as the CH, attempts to combine advantages of the GH and the FCH in the sense of calculation speed and solution quality, respectively.

The CH starts with grouping as done in the GH. As some flows have been assigned, the numbers of flow and container variables are reduced, and then forcing constraints are inserted to drive flows to optimal paths. The LP relaxation with forcing constraints will be solved repeatedly until the solution no longer changes. Then, flight capacity will be examined, as in the GH and FCH, to determine if sort savings on unused flight capacity can be made. The detailed procedures are as follows.

Combined Heuristic

1. *Performing the GH.* Execute GH until no container group can bypass two or three nodes.
2. *Updating inputs.* Remove assigned flows, eliminate all $c_{a,g}$ and f_{kq} variables that bypass two or three nodes, and recalculate inputs.
3. *Performing the FCH.* Insert forcing constraints for the updated inputs and solve the LP relaxation with forcing constraints.
    3.1. If the IP solution is obtained, stop.
    3.2. If the solution changes from the last iteration, assign flows that are greater than one container capacity to associated container groups, then repeat step 2.
    3.3. If the solution does not change from the last iteration, execute step 4.
4. *Grouping flows to make more savings*. For the remainder of container groups,
    4.1. Select container group with max $\sum_i s_i / \sum_{a,g} w_{a,g}$, where s_i = sorting cost and $w_{a,g}$ = wasted flight capacity.
    4.2. Check if associated flight capacities are available for the selected container group.
        4.2.1. If available, assign possible flows into the selected container group.
        4.2.2. If not, repeat steps 4.1-4.2.
    4.3. Update sorting capacities of nodes that the selected container group bypasses.
    4.4. Update possible flows in associated container groups.
    4.5. Update associated flight capacities.
    4.6. Repeat steps 4.1-4.5 until no container group can be assigned.

The underlying concept of the CH is to obtain parts of the solution from the GH so that the number of forcing constraints is reduced for applying the FCH, leading to the reduction of the entire calculation time.

The flow of the CH is depicted in Figure 5.7.

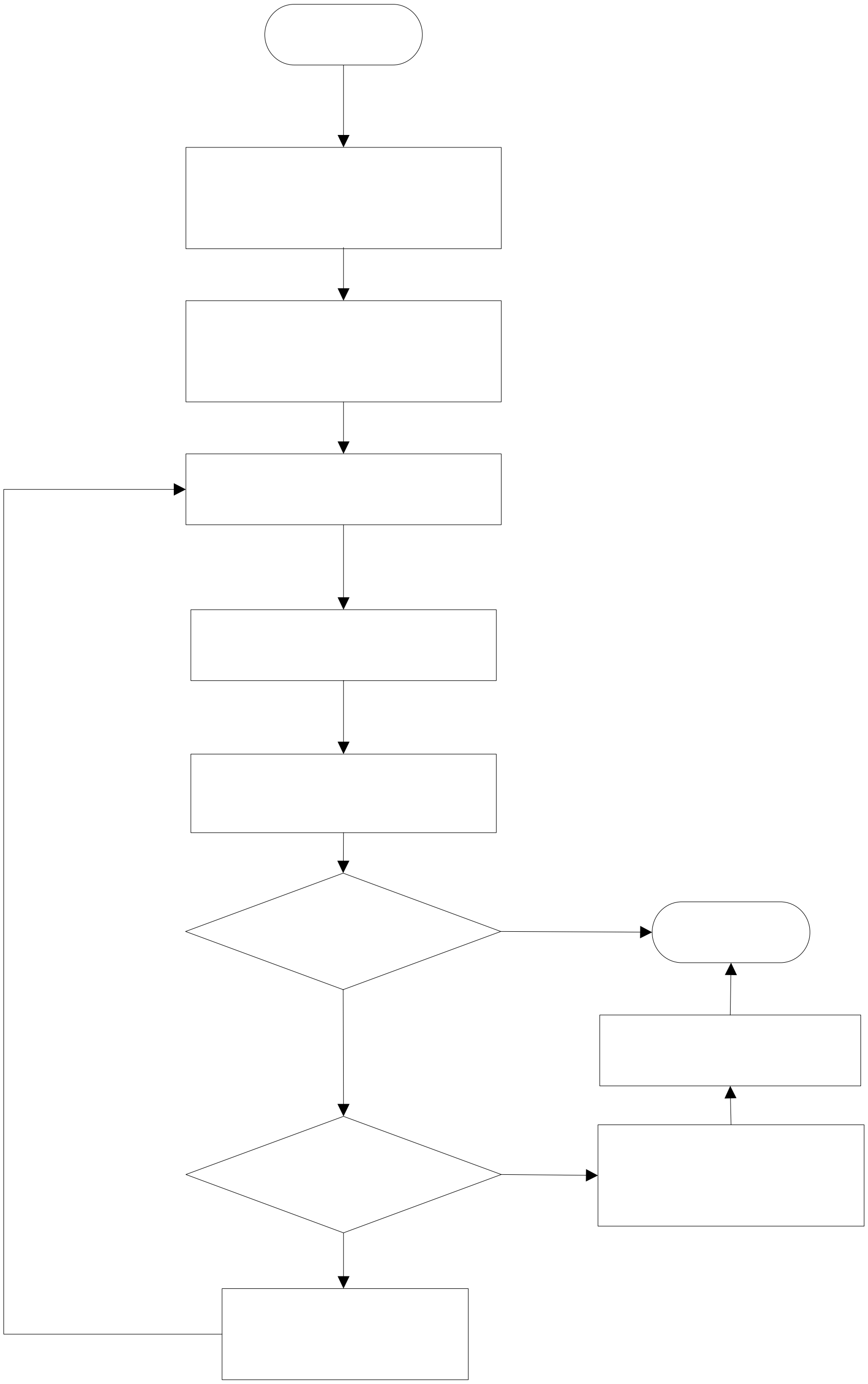

Figure 5.7. Combined Heuristic

5.1.3.4 Alpha Value

For large problems, the computation time of the FCH is still not acceptable due to a large set of forcing constraints. To balance the speed and the solution quality, *alpha* *value*, α , is introduced to eliminate variables that are not likely to be part of the final solution. The alpha value represents a fraction of full container capacity. If the maximum number of units of a commodity in any path q is lower than alpha*container capacity, the associated flow variable and container variables are eliminated from the formulation. However, it is possible that eliminated variables are part of an optimal solution. As the alpha value increases, the solution quality tends to deteriorate due to fewer routing options but the computation time tends to get faster as a result of the reduction of number of variables. Therefore, an optimal value of alpha should be determined to balance the tradeoff between the solution quality and computation time. In our experiment, different alpha values are tested with the FCH. The results are shown in section 5.2.

5.2 Experimental Approach

The performance is evaluated by calculating the solution quality and computation time with different test problems. For large problems, optimal solutions cannot be obtained within reasonable time using the CPLEX MIP solver. The solution quality is then measured by the percentage closeness of the objective value to lower bound, calculated as follows.

$$\text{Percentage closeness} = (1 - \frac{(Obj - LB)}{LB})*100$$

Test problems were experimented on an Intel Pentium 4 Processor, running at a clock speed of 2.4 GHz with 1 GB RDRAM. Computation time is the CPU time in obtaining MIP solutions. This section describes how lower bounds are computed and scenarios of problem instances.

5.2.1 Lower Bound

Since optimal solutions for large MIP problems cannot be obtained using CPLEX 8.0 within acceptable time, we use lower bounds to benchmark the objective values of the proposed heuristics. A lower bound could be the LP relaxation solution but it would be a loose bound because integer variables tend to be fractional values to avoid sorting costs as discussed earlier. To obtain a tight bound, we insert forcing constraints as discussed in section 5.1.3.2 into the LP relaxation to tighten the LP solution. The optimal solution of the LP relaxation with forcing constraints is then the lower bound. The performance of the lower bounds is demonstrated in the experimental results section.

PROPOSITION 1. The objective value of the optimal solution of the LP relaxation with forcing constraints is less than or equal to the objective value of the optimal integer solution.

PROOF. Since all forcing constraints must be satisfied for an optimal IP solution, the optimal IP solution is in the feasible region of the solution set of the LP relaxation with forcing constraints. Thus, the minimum objective value of the solution in the feasible region will be less than or equal to the objective value of the optimal IP solution.

5.2.2 Problem Instances

In our experiments, test problems were examined in different network sizes, each of which has a different number of terminals as follows.

Network Size	# National Hub	#Regional Hubs	# Major Airports	# Local Airports	# Rural Airports	# of Local Terminals
20	1	2	2	15	0	0
35	1	2	3	19	10	0
50	1	2	4	25	18	0
63	1	2	5	30	25	0
78	1	3	7	35	32	0
95	1	4	8	40	42	0
112	1	6	10	47	48	0

Table 5.1. Number of nodes of each terminal type for different network sizes

The network size is based on the number of nodes; arcs were created based on the network connectivity discussed in section 4.3. For each problem size, we created three different scenarios in which flight capacities are varied under the same network. Specifically, in each scenario, a possible path for each commodity was assigned a proportion of total amount of the commodity.

Let,

p_q^k = proportion of total amount of a commodity k for a possible path q.

Then

$\sum_q p_q^k$ = 1, for all k.

The capacity of each flight is an aggregation of these proportions of all commodities that use the flight on their routing paths plus allowable wasted capacity.
Let q consists of a set of arcs $a_{ij.}$ Flight capacity a_{ij} can be written as follows,

M(a_{ij}) = $\sum_q \sum_k p_q^k d^k \varphi_{ij}^q * (1+A_w)$

, where
M(aij) = flight capacity of arc a_{ij}.

d^k = demand of commodity k.

φ_{ij}^{q} = incident vector for arc a_{ij} associated with path q.

A_w = allowable wasted capacity.

In our experiments, we set the allowable wasted capacity equal to 2% of flight capacity. Three scenarios are varied by the proportions used in allocating flight capacities which results in different flight capacities under the same network size and connectivity. Container capacity is 200 for all problem sizes. Average flight cost per shipment per arc and average sorting cost per shipment per terminal are forced to be a fixed ratio since the solution quality is sensitive to these costs. We examine these factors in the sensitivity analysis section. All other parameters were generated as discussed in section 4.3.

5.3 Experimental Results

In this section, we first show the tightness of lower bounds and benchmark the solution quality of the three heuristics against the lower bounds. Calculation time of each approach is also presented.

5.3.1 Lower Bound

Based on experiments with our test problems, CPLEX can find an optimal MIP solution within reasonable time for most problems with network size up to 35 nodes. Even for some 35-node problem instances, it took more than 12 hours to acquire the optimal solutions. We evaluate the tightness of the lower bound by the percentage closeness to optimality for these smaller problems, calculated as

$$\text{Percentage closeness to optimality} = (1 - \frac{(LB - O)}{O}) * 100$$

, where *O* is the objective value for the optimal solution obtained from CPLEX 8.0. The percentage closeness to optimality and computation time of lower bounds for network size of 10 nodes to 35 nodes is shown in Table 5.2. The lower bounds obtained from running our test problems are all within 1% of optimality.

Size	LB			CPLEX Optimal solution	
	Percentage Closeness	Computation time (secs.)	Objective Value	Computation time (secs.)	Objective Value
10	100.00%	0	363200	0	363200
20	99.29%	1	3335418	2	3359362
20	99.19%	1	3311636	2	3338847
35	99.91%	3	43041287	91	43081215
35	99.46%	8	46769523	44748	47025212

Table 5.2. Percentage closeness to optimality and computation time of lower bound

5.3.2 Grouping Heuristic

The GH was tested with three scenarios under different network sizes. The experimental results are shown in Figures 5.8 and 5.9.

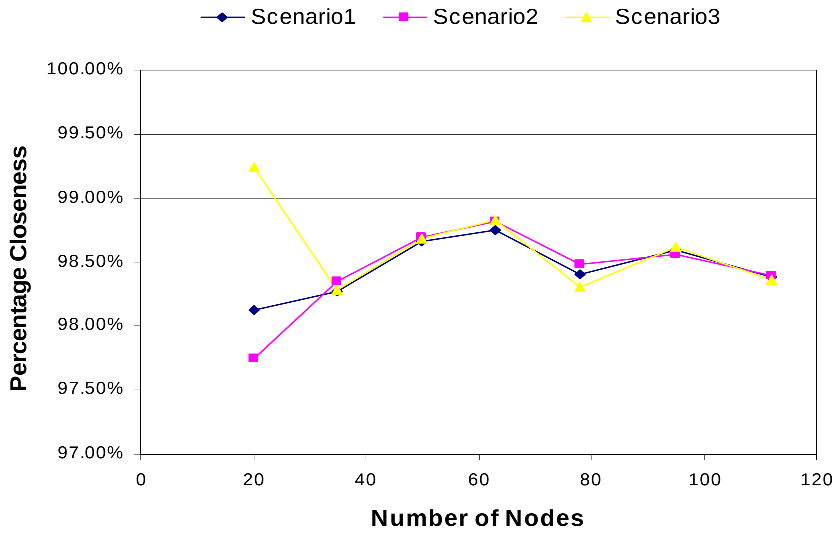

Figure 5.8. Percentage closeness to LB for the GH with three different scenarios vs. network size

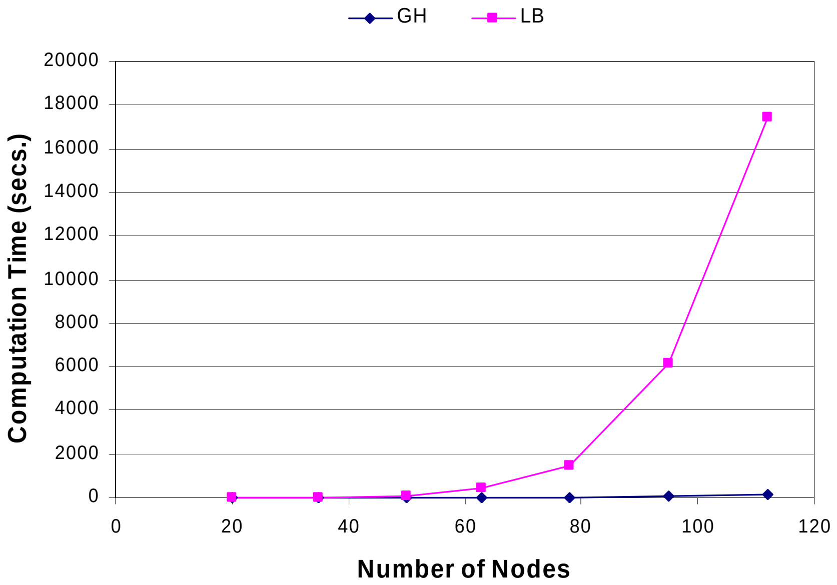

Figure 5.9. Computation time of GH and LB calculation vs. network size for the first scenario

As shown in Figures 5.8, the solution quality of the GH ranges between 97.75% and 99.24%. The results show slights differences in solution quality among three scenarios. This indicates that varying flight capacities under the same network size does not have effects over the solution quality. Figure 5.9 indicates that computation time of the GH is relatively small compared to the lower bound calculation time, due to a large number of forcing constraints added for computing lower bounds. The trends are also the same for the other two scenarios.

5.3.3 Forcing Constraint Heuristic

Experiments for the FCH were conducted in the same manner as with the GH. The results are shown in Figures 5.10 and 5.11.

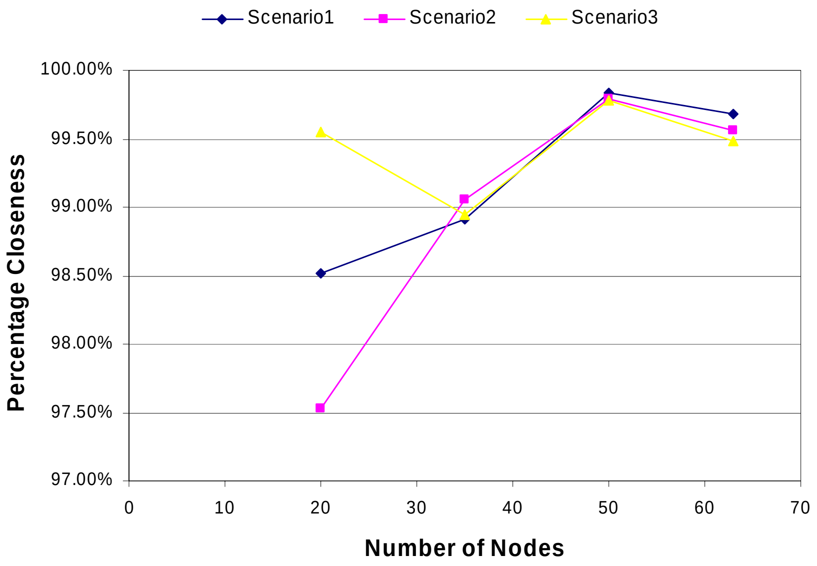

Figure 5.10. Percentage closeness to LB for the FCH with three different scenarios vs. network size

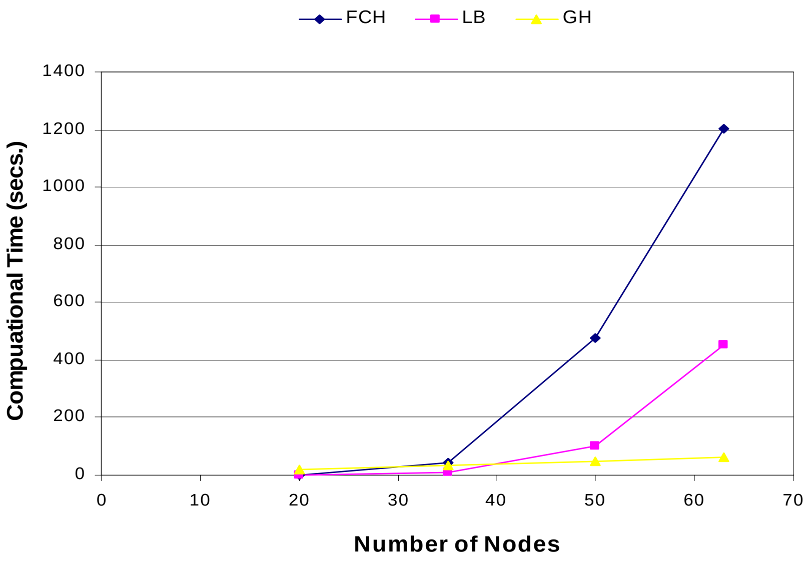

Figure 5.11. Computation times of the FCH, GH, and lower bound vs. network size for the first scenario

As shown in Figure 5.10, the percentage closeness of the FCH ranges between 97.53 and 99.79, giving better results than the GH on average. However, in terms of computation times, the GH outperforms the FCH, as shown in Figure 5.11 for the network size larger than 60 nodes; the speed of the FCH is rather unacceptable. Experiments for different alpha values were conducted to obtain feasible solutions for large problems.

5.3.4 Alpha Value

Alpha values, ranged from 0.0 to 1.0, were incorporated into the FCH and tested with two different problem sizes of 35 and 50 nodes. The results are shown in Figures 5.12 and 5.13.

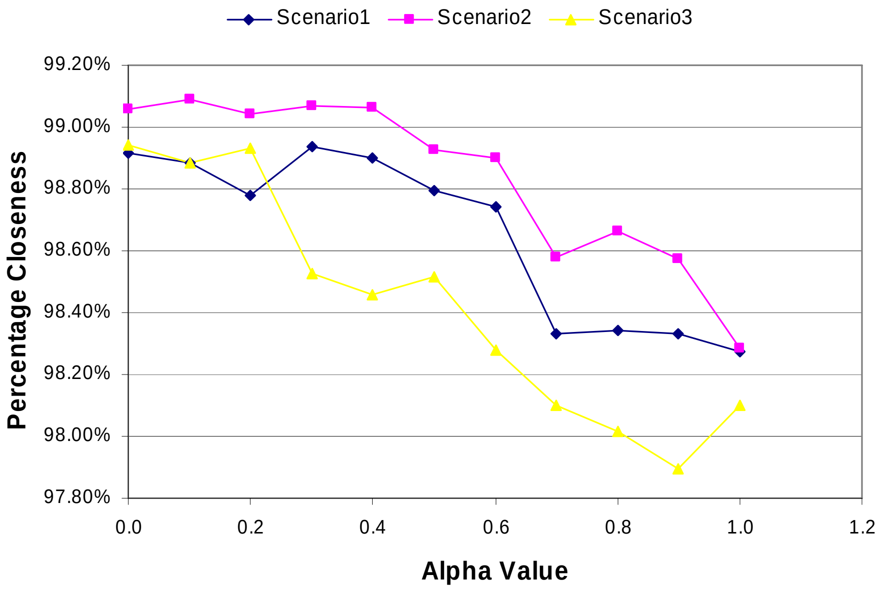

Figure 5.12. Percentage closeness to LB for the FCH with three different scenarios vs. alpha value for network size of 35

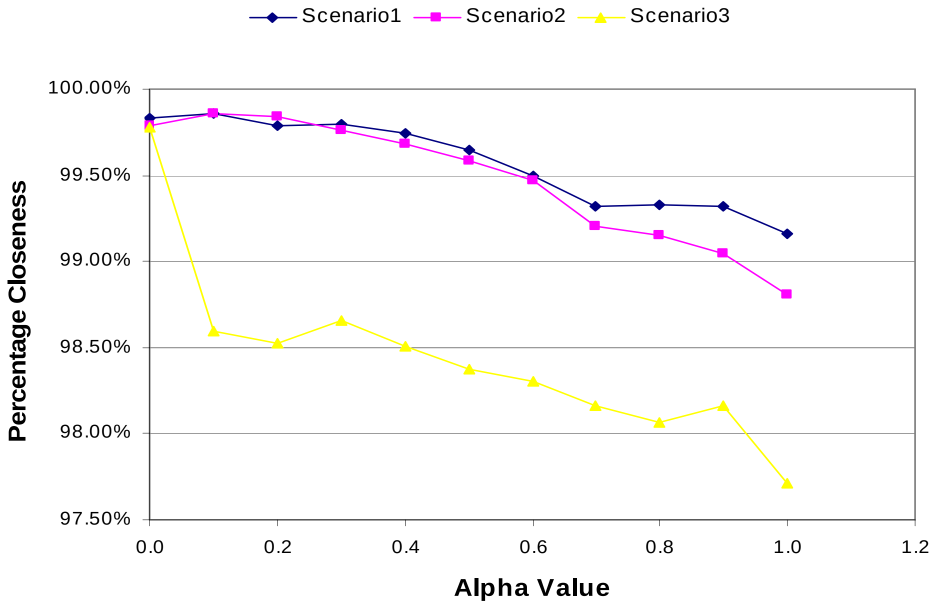

Figure 5.13. Percentage closeness to LB for the FCH with three different scenarios vs. alpha value for network size of 50

Figures 5.12 and 5.13 show the solution quality for different values of alpha under different network sizes. The results show that the solution quality deteriorates when alpha value is increased. However, no common sign of swift change can be inferred from various scenarios. An interesting point here is that the objective value could be improved when the alpha value increases. An explanation is that with lower alpha values, flows that are in the optimal container group might not be assigned to the optimal containers in the first place because these flows are distributed to different container groups. As the FCH proceeds, flight capacities are occupied and unavailable for these flows. When the FCH terminates, these flows possibly end up entering sorting nodes, resulting in higher objective values. However, in general, objective values tend to increase when alpha values increase. Similarly, when alpha values increase, despite fewer variables, computation time could possibly increase because the FCH may take more loops to complete the procedure, depending on flow assignment on each iteration. Nevertheless, the trend of improvement in computation time is pronounced for relatively large alpha values.

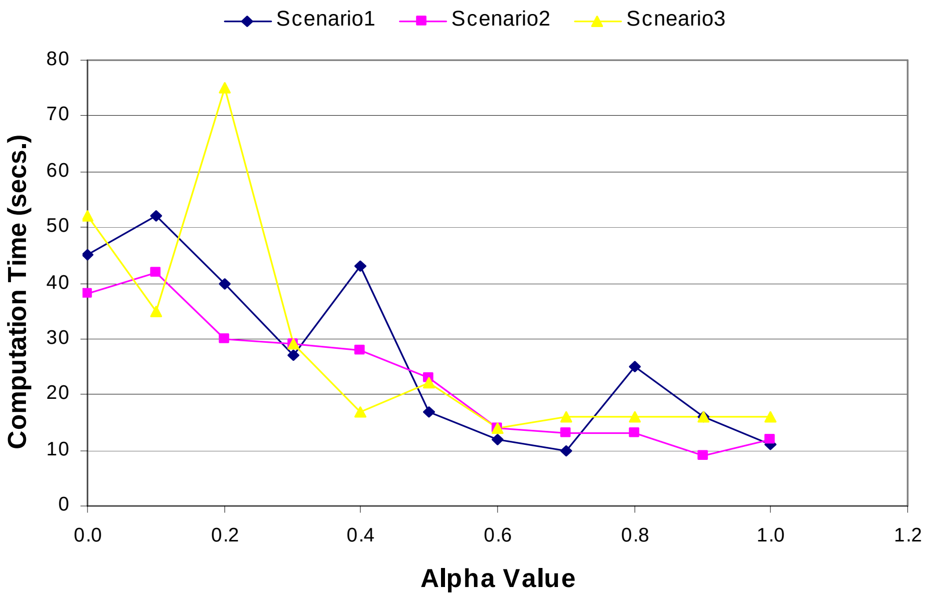

Figure 5.14. Computation time for the FCH with three different scenarios vs. alpha value for network size of 35

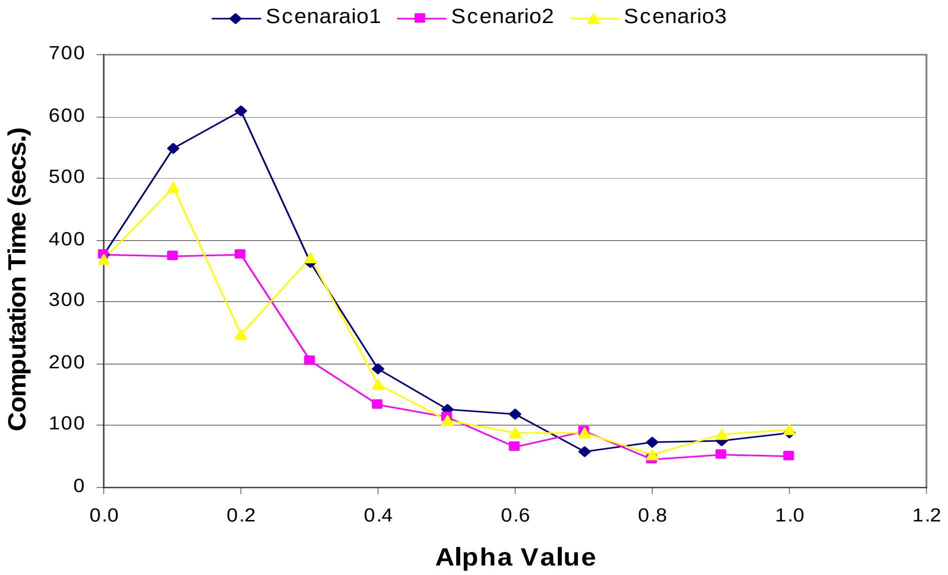

Figure 5.15. Computation time for the FCH with three different scenarios vs. alpha value for network size of 50

Figures 5.14 and 5.15 show that the reductions of computation time are obvious from alpha values 0.0 to 0.7. The ratio of calculation time between alpha value 0.0 and 0.7 is 5:1 approximately. From alpha values 0.7 to 1.0, the reduction is unclear. Alpha values of 0.7 and 0.8 are tested for network size of 63 and 78 nodes. The results are shown in Tables 5.3a and 5.3b.

No. of Nodes	Scenario 1		Scenario 2		Scenario 3	
	0.7	0.8	0.7	0.8	0.7	0.8
63	99.66%	99.55%	99.62%	99.62%	99.71%	99.70%
78	99.39%	99.23%	99.77%	99.77%	99.75%	99.56%

Table 5.3a Percentage closeness to LB of the FCH with different alpha values for problem sizes of 63 and 78

No. of Nodes	Scenario 1		Scenario 2		Scenario 3	
	0.7	0.8	0.7	0.8	0.7	0.8
63	284	144	222	177	205	165
78	745	421	844	444	619	362

Table 5.3b Computation time of the FCH with different alpha values for problem sizes of 63 and 78

The results for problem sizes of 63 and 78 nodes indicate slight deficiency in solution quality, but demonstrate noticeable reductions in computation time. Thus, in performing the FCH, we use an alpha value of 0.8 for large size problems.

The results of the FCH using alpha value = 0.8 for problem instances larger than 50 nodes compared to the GH for the first scenario are as follows.

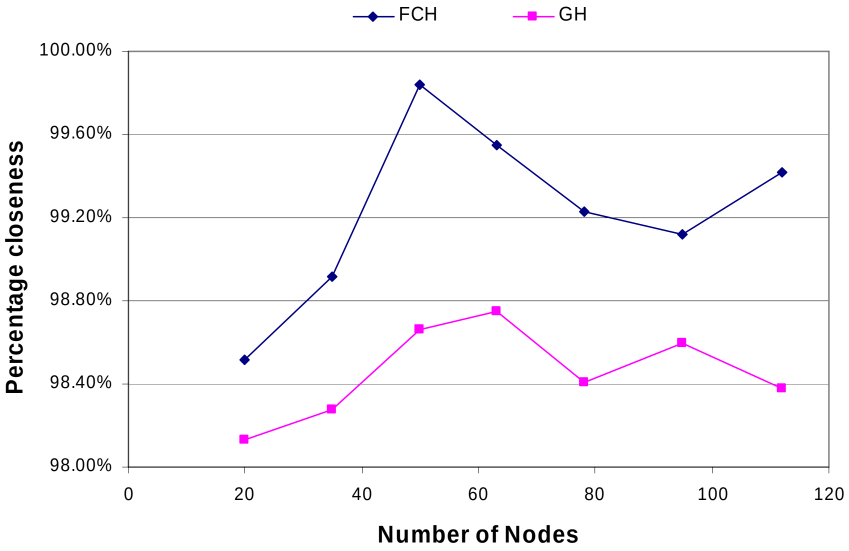

Figure 5.16. Percentage closeness to LB of FCH and GH vs. network size

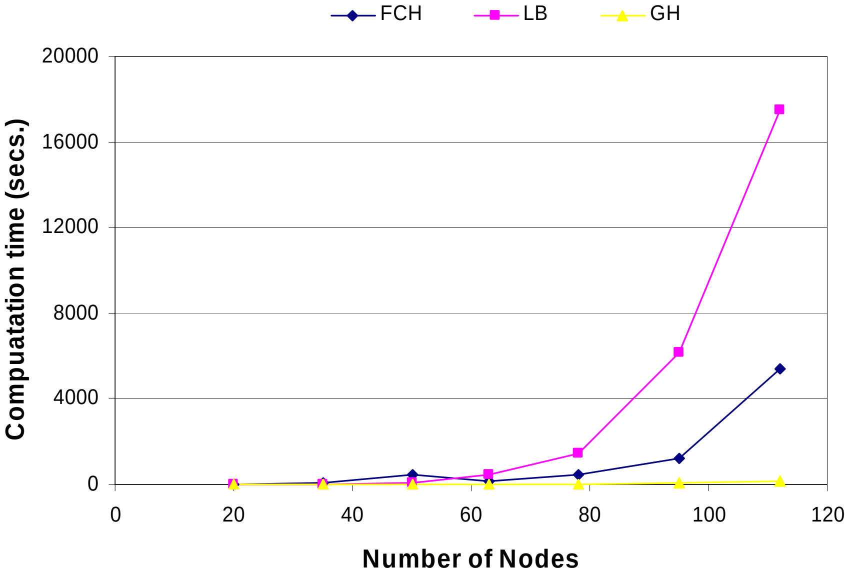

Figure 5.17. Computation times of the FCH, GH, and lower bound vs. network size

As shown in Figures 5.16 and 5.17, with the alpha value 0.8, the solution quality of the FCH is still better than the GH for all network sizes; however, the GH is much faster when the network size is larger than 50 nodes. The results are as expected; the GH is a speedy approach as the computation is only to solve the LP relaxation excluding bypass arcs. For the FCH, not only the LP relaxation with bypass arcs is to be solved, but a large set of forcing constraints are included. The solution quality of the FCH is better than the GH because the FCH accounts for sorting costs in determining routing sequences while the GH finds the routings based on flight costs alone. Using the alpha value to reduce problem size could be a promising technique as the solution quality is still satisfactory while computation time is significantly reduced.

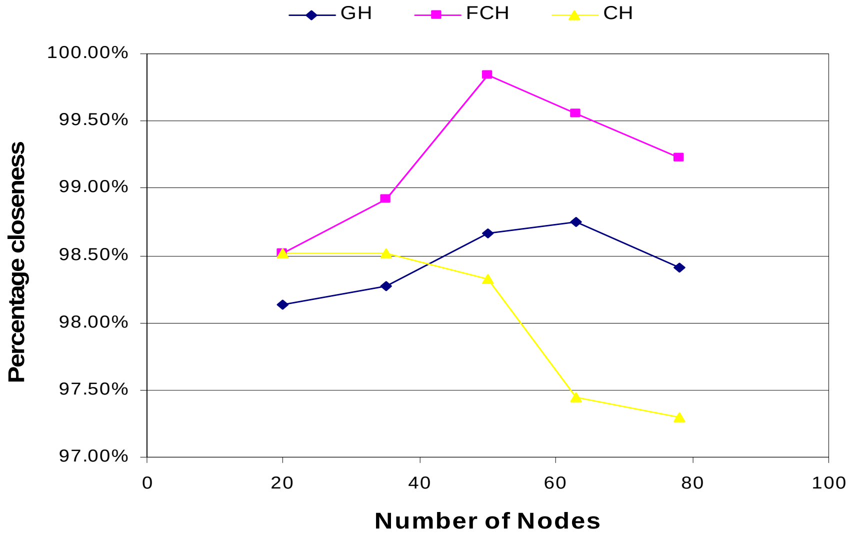

Figure 5.18. Percentage closeness to lower bound of the GH, FCH and CH vs. network size for the first scenario.

Note: Alpha value = 0.8 is used with the FCH for network size larger than 63 nodes.

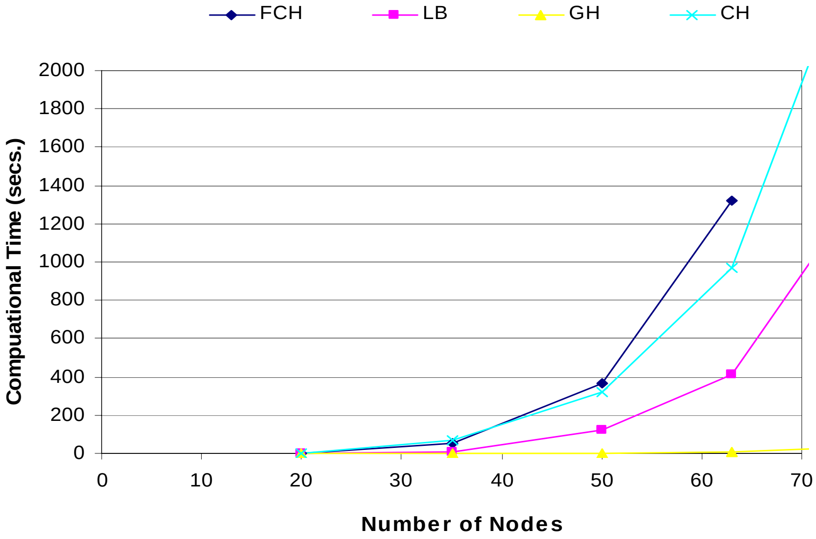

Figure 5.19. Computation times of the GH, FCH, CH, and lower bound vs. network size

5.2.3 Combined Heuristic

As shown in Figure 5.18, the solution quality of the CH deteriorates as the problem size increases. This trend also occurs for scenarios 2 and 3. This could be because, after the GH has been completed in the beginning steps of the CH, several optimal paths are eliminated and then the rest of the flows are distributed to different remaining paths as indicated by the solution obtained from the FCH. Since optimal paths are possibly eliminated, the solution quality is inferior to the solution obtained from the FCH. The deterioration is obvious when the problem size increases because a lot more commodities share the same flights and, thus, eliminating paths at the beginning will have significant effects on finding optimal routings. Also, according to the procedure, flows on any container group are obtainer from iteratively solving the LP relaxation. When no container group holds flows that greater than one container capacity, flows on different paths are not grouped so more bypass containers could not be formed. On the other hand, under the GH, grouping possible flows on different paths is possible and could result more sort savings.

Figure 5.19 shows calculation time of the CH compared to the FCH, the GH, and LB calculation. As expected, the CH calculation time lies between the FCH and the GH. However, as the solution quality is even worse than the GH when the problem size is larger than 50 nodes. The CH will be excluded from sensitivity analysis in the following section.

5.4 Sensitivity Analysis

The solutions obtained from the heuristics could be a reliable guide under a set of parameters. However, some model parameters could have significant effects on the performance of the heuristics. In this section, sensitivity analysis is performed to investigate the effect on the solution provided by the developed heuristics if the parameters change. Factors under consideration are container capacity and t/s ratio. The network size under examination is 35.

5.3.1 Container capacity

In these test problems, aircraft capacities, measured in number of shipments, remain the same. Thus, as we increase container capacity on each flight, the number of containers on each flight is reduced. As shown in Figure 5.20, an increase in container capacity does not significantly affect the performance of the FCH and the GH. In fact, larger container capacity possibly improves the performance of the GH because fewer flows can be grouped and the optimal solution will put more weight on routing costs. However, the results for scenarios 2 and 3 also show no sign of significant changes in the solution quality for both heuristics.

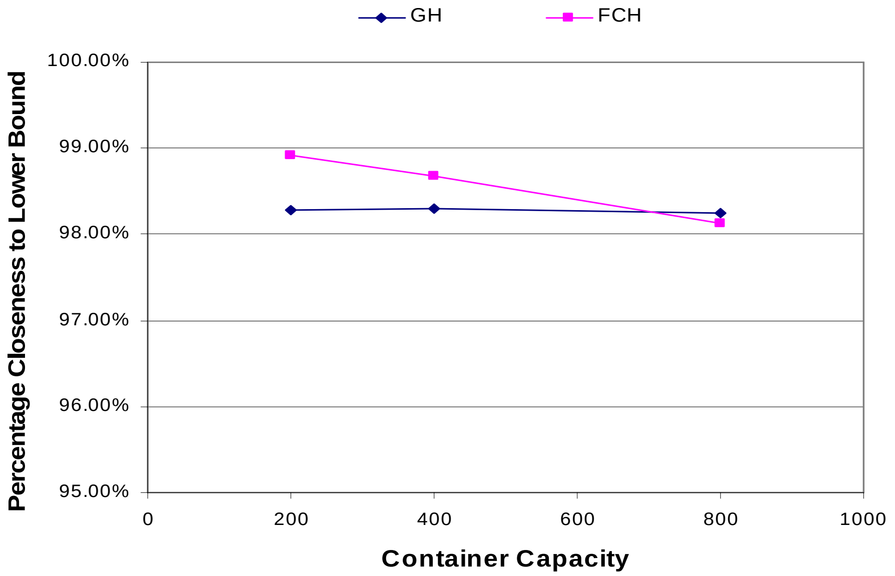

Figure 5.20. Percentage closeness to LB vs. container capacity for the first scenario

5.3.2 T/S ratio

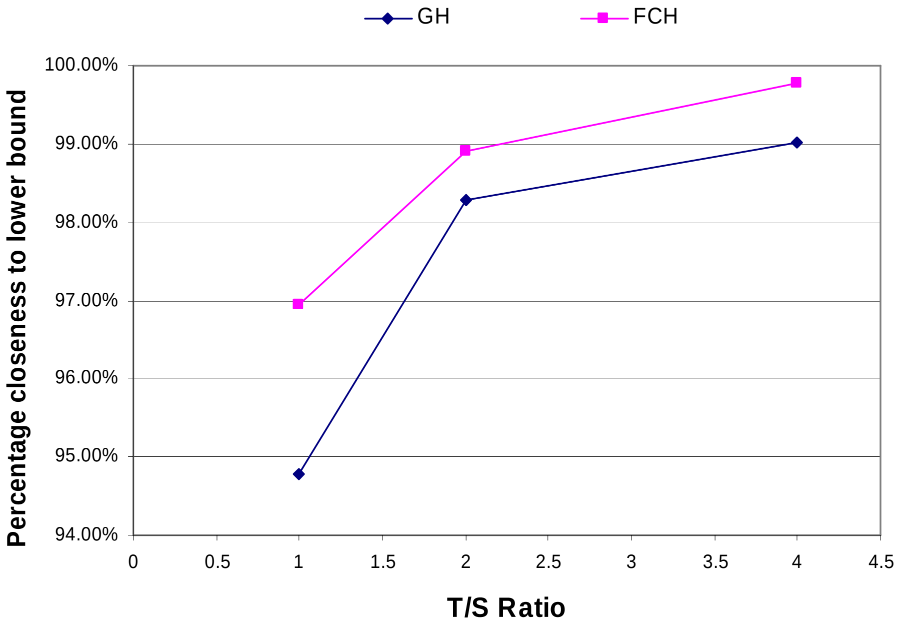

Figure 5.21. Percentage closeness to LB vs. t/s ratio for the first scenario

T/S ratio is defined as average transportation cost per shipment per arc/ average sorting cost per node. In our experiments, t/s ratio = 2.0 was used for all runs. Figure 5.21 presents the effects of t/s ratio on the solution quality of the GH and the FCH. With t/s ratio = 1.0, the performance of both heuristics are significantly worse. The deficiency of the performance can be explained by two reasons: the deterioration of percentage closeness of lower bounds and the characteristics of the heuristics themselves.

To illustrate the first rationale, we evaluated the percentage closeness to optimality of lower bounds by comparing the objective values with the objective values of optimal solutions obtained from CPLEX. The network size of 20 nodes was examined to acquire optimal solutions in reasonable time.

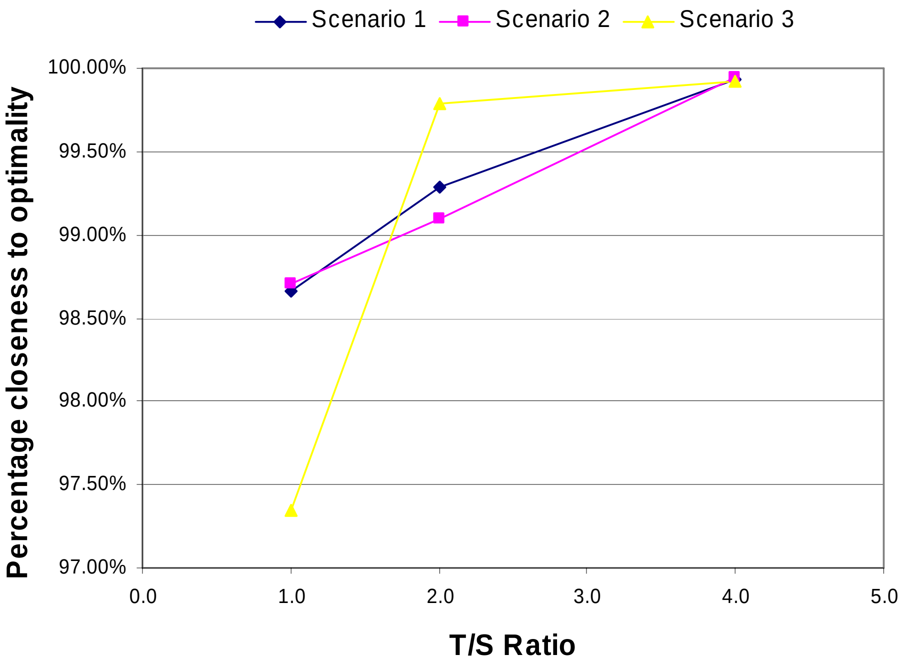

Figure 5.22. Percentage closeness to optimality of LB for network size of 20

As shown in Figure 5.22, the percentage closeness to lower bounds notably declines with lower t/s ratios. This indicates that the performance of the heuristics, if benchmarked with percentage closeness to optimality, would show better results for problems instances with low t/s ratios because the gaps to optimality are relatively smaller than the gaps to lower bounds with low t/s ratios.

Another reason for the performance deterioration is the characteristics of the heuristics themselves. For the GH, this is because the GH attempts to find the solutions based on transportation costs and, then, form container groups to make sort savings. The terminal sequences used in routing do not change after obtained from solving the LP relaxation. The change is container groups used for commodities in those terminal sequences. However, in the optimal solution with low t/s ratio, flows tend to use more expensive terminal sequences so as to avoid high sorting costs; these paths are not likely to be chosen in the beginning steps of the GH. Assuming sorting cost is zero and sorting capacity is unlimited, the solution obtained from the GH will be the optimal solution. When the sorting cost component increases, the GH will get further from the optimal solution. As for the FCH, the deficiency of the approach results from the fact that, in the optimal solution with low t/s ratio, flows will be distributed more to bypass arcs; thus, there are smaller number of container groups holding flows up to one container capacity. This reduces chances in assigning flows to bypass container groups and obtaining paths used in the optimal solution. The solution quality of both heuristics is noticeably improved as the ratio increases. It should be noted that, with t/s ratio of 4.0, the computation time of FCH is significantly improved from 50 seconds to 10 seconds on average under the same network size for three scenarios. In addition, under the same network size of 35 nodes, CPLEX can acquire optimal solutions for problems with t/s ratio of 4.0 in 24 seconds while it cannot obtain the optimal solution for the same problem with a t/s ratio of 1.0 within 12 hours.

In practice, t/s ratio can greatly vary, depending on package sizes. Envelopes, for example, tend to have a small t/s ratio since they can be transported in bulk which incurs low transportation cost per shipment but sorts for a bulk of these packages could be resource-intensive. On the other hand, oversized packages would occupy considerable space on vehicles but sorting involvement, compared to shipments with the same volume, would be relatively uncomplicated, leading to high t/s ratios. Thus, the t/s ratio could be a meaningful factor of consideration in opting for a suitable heuristic method for an individual scenario.

5.5 Summary

In this section, problem solving schema and three heuristics are presented as well as experimental results. The first heuristic, the GH, is based on a greedy algorithm. The second heuristic, the FCH, uses forcing constraints to acquire feasible solutions. Due to a large set of forcing constraints added to perform the FCH, an alpha value is introduced to balance computation time with solution quality. The third heuristic, the CH, is the combination of the first two heuristics.

Experiments were conducted to examine the algorithm performance in terms of solution quality and computation time. Experimental results show that, in terms of solution quality, the FCH outperforms the GH and the CH but, as expected, the computation time is the highest as a result of added forcing constraints. As for the computation time, the GH performs best among the three heuristics because the heuristic solves the LP relaxation excluding artificial arc. The speed of the CH lies in between the GH and the FCH but the solution quality drastically deteriorates for large problems and became the worst for test problems larger than 50 nodes. The alpha value was tested with the FCH. The results show that the computation time of the FCH declines considerably from alpha = 0.1 to 0.6, while the solution quality does not drop as much and is still better than that of the GH and the CH. Although the speed of the CH is between the two heuristics, the solution quality is the worst; therefore, it was excluded from sensitivity analysis. The sensitivity analysis was performed to identify the effect of changes in parameters. The results show that container capacity has little effect on the algorithm performance of the GH and the FCH. As for changes in transportation costs and sorting costs, the two heuristics evidently deteriorate as sorting costs increase.

6. Model Extensions

The original model assumes that sort capacity is a single aggregate value, measured in number of shipments. However, in reality, sort capacity is a rate measured in shipments per time unit and, thus, the number of shipments that can be sorted depends on shipment arrivals as a queuing process, as mentioned in Section 3. It is anticipated that sorting operations are congested only in some periods of a day after a cut-off time. During these congested periods, reducing the number of shipments to be sorted would be particularly beneficial because the process can be finished earlier, which enhances the flexibility of the work schedule. Therefore, containerization over peak and non-peak periods should produce different savings.

In addition, in practice, the very first terminals with no incoming shipments (e.g. local terminals, rural airports), referred to as the source terminals, have a particular characteristic in common in that they have two discrete options in processing incoming shipments: 1. to sort all shipments and route them to connected terminals based on their optimal routing, or 2. to sort no shipments and transfer them to a single connected terminal. The latter option causes some shipments to use more expensive routes; however, it enables quicker transfer as well as produces sort savings at the origin terminal.

To better conform to real world practice, we enhance the original model by capturing these characteristics in an extended model. In this section, the modified formulation and a problem solving approach are presented, followed by experimental results.

6.1. Extended Components

6.1.1. Capacity by time periods

In capturing the traffic congestion of a sorting facility, we classify operation time periods into non-peak and peak periods for each terminal. With the assumption of fixed schedules of arrivals, we can determine that a shipment will arrive at the subsequent terminal in peak or non-peak periods. Shipments that arrive in peak periods will produce more savings if containerized.

Sort capacity in peak periods is particularly critical to terminals that handle a large number of shipments, namely, national hubs and regional hubs. Therefore, we modify the network by duplicating all the hub nodes. The duplicated nodes represent terminals in peak periods with limited sort capacities and higher sorting costs. Non-peak period nodes are assumed to have unlimited capacities. The modification enlarges the network by the number of duplicated nodes; however, the mathematical formulation changes little.

6.1.2. All-sort/non-sort operations

The source terminals with no incoming shipments from other terminals have options to be a sort facility or act as a pure transshipment terminal. The benefit of sorting all shipments is that all shipments could be sorted and directed to their cheapest routes with potential bypasses over subsequent terminals. On the other hand, the non-sort option could save fixed operation costs and variable sorting costs in the amount of all shipments collected and also expedite shipment transfer.

To capture this aspect under the assumption that a source terminal could be connected to at most two terminals, we create, for each terminal, a dummy node with three outgoing arcs linked with the source terminal and the two connected optional terminals of the source terminal. The modification of the network is depicted as follows.

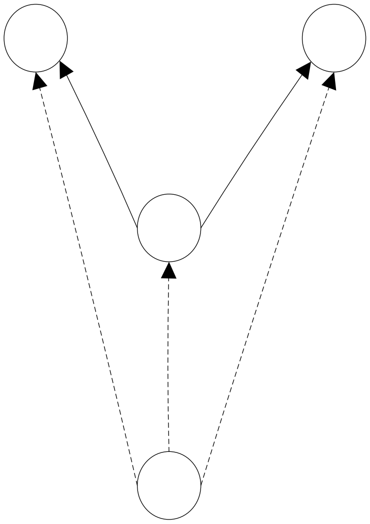

Figure 6.1. Dummy nodes and arcs

From Figure 6.1, node o is a source terminal connected to two terminals, nodes c_1 and c_2. Dummy node d and three artificial arcs are created for the terminal. The three arcs represent the options of sorting at the origin terminal. If the sort occurs at the source terminal, arc a_{do} will be used and sorting costs will be imposed. The two arcs that link the dummy node and the connected terminals incur transportation costs from the source terminal to the connected terminal. Only one of the three artificial arcs is chosen in a feasible solution.

Incorporating this feature into the model notably complicates the problem formulation because binary decision variables must represent the decision of using the terminals. The formulation can be modified as follows.
From the original formulation,

Min. $\sum_{k\in K}\sum_{i\in I}\sum_{q\in Q(k)} SC_q^k w_{iq} f_q^k + \sum_{k\in K}\sum_{a_{ij},g}\sum_{q\in Q(k)} RC_{a_{ij},g} u_{a_{ij},g}^q f_q^k$

s.t. $\sum_{g\in G(a_{ij})} c_{a_{ij},g} \le M(a_{ij}) \qquad \forall a \in A \qquad (1)$

$\sum_{k\in K}\sum_{q\in Q(k)} u_{a_{ij},g}^q f_q^k \le Bc_{a_{ij},g} \qquad \forall a \in A, \forall g \in G \qquad (2)$

$$\sum_{q \in Q(k)} f_q^k = d^k \qquad \forall k \in K \qquad (3)$$

$$\sum_{k \in K} \sum_{q \in Q(k)} w_{iq} f_q^k \leq S(i) \qquad \forall i \in I \qquad (4)$$

$$f_q^k \geq 0 \quad \forall q \in Q(k), \forall k \in K$$

$c_{a_{ij},g} \in I$ I = set of integer numbers, $\forall a_{ij} \in A(i,j), \forall g \in G(a_{ij})$

Where,
K = set of all commodities *k* identifying o-d pair.
I = set of all terminals in the network.
Let,
S = set of source terminals s where no incoming shipments from other terminals; $S \subset I$.
M = set of dummy nodes m associated with source terminals s.
N = set of optional terminals n associated with dummy nodes m; $N \subset I$.
K’ = set of commodities k’ which are originated from source terminals s.
A = set of artificial arcs a_{mi} connecting dummy nodes m to source terminals s or optional terminals n.
Q’(k’) = set of possible path q’ of commodities k’.

Commodities k’ originated from terminals s are moved to associated dummy nodes m as if they are originated at nodes m associated with terminals s and demands k’ at terminals s now become 0. Possible paths q’ for commodities k’ are generated using artificial arcs a_{mi}. Q’(k’) is the original set of possible paths, Q(k’), with relocated origin nodes, plus a set of additional possible paths for the all-sorting option.

Additional constraint sets are included as follows,

$$\sum_{\substack{k' \\ orig=s}} \sum_{q'} f_{q'}^{k'} \leq \sum_{\substack{k' \\ orig=s}} d^{k'} x_{mi} v_q^{a_{mi}} \qquad \forall s \in S$$

$$\sum_i x_{mi} = 1 \qquad \forall m \in M$$

x_{mi} = binary

where,

Parameters

$v_{q'}^{a_{mi}}$ = binary parameter specifies the association of arc a_{mi} on path *q’*. If arc a_{mi} is included in path *q’*, then $v_{q'}^{a_{mi}}$ = 1; otherwise, 0.

Decision variables
x_{mn} = set of binary variables indicating that artificial arcs *mi* are used.

Sorting capacity at source terminal s is assumed to be large enough to handle originating demand.

6.2. Problem Solving Approach

6.2.1 Initialization

A feasible solution can be acquired by solving the extended problem by relaxing binary integer restrictions using one of heuristics presented in Section 5. It can be anticipated that commodities k' will not use the arcs connected to terminals s because sorting costs are imposed. Most likely, demands at a source terminal s will be segmented into two groups and use the two optional connected arcs that give the cheapest routing costs. Also, it is possible that all commodities k' at a terminal will travel to the same terminal, which, in this case, binary integer constraints associated with the terminal are satisfied. However, the solution we obtain from solving the extended problem with relaxing binary integer restrictions could be viewed as if all flows go to the source terminal s and are divided after being sorted, meaning that the net objective value of the feasible solution is the objective value of the solution obtained plus sorting costs of flows that are originated from the same source terminal and use both two connected arcs.

6.2.2 Heuristic Approach to Improve Initial Solutions

From the initial solution, improvements can be made by avoiding sorting costs at the source terminals. However, changing routing from the sorting option to non-sort routings result in higher transportation costs and sorting costs at intermediate terminals and may affect some other flows to use more expensive routings. Thus, to make improvements over total costs, we need to examine sort cost savings at ultimate origin terminals and costs incurred from switching routings.

We developed a heuristic procedure based on examination of this tradeoff. For each terminal s, sorting cost that can be saved is known. Here, what we need to determine is transportation costs (routing costs plus sorting costs) that would be imposed if only one of the two connected arcs is used for all demands from terminal s. The cost changed in total costs can be estimated by moving flows from one arc to the other arc. The moved flows will be assigned new routings by using routings of flows that have the same destinations, at the node connected to the arc which the flows are moved to. New transportation costs will be estimated accordingly and compared to transportation costs of using the all sort option in the solution. The costs changed by moving flows is the sum of new total costs less the sum of original total costs of flows at dummy node m. The costs changed at node m can be written as follows.

$$\sum_{k'} \sum_{q'} RC_{q'} f_{q'}^{k'} + \sum_{i} \sum_{q'} SC_i w_{iq'} f_{q'}^{k'} - (\sum_{k'} \sum_{q} RC_q f_q^{k'} + \sum_{i} \sum_{q} SC_i w_{iq} f_q^{k'})$$

, where

$RC_{q'}$ = routing cost of using path q'.
SC_i = sorting cost at terminal i.
k' = commodities to be moved at node m.
q = original paths of commodity k' obtained from the solution.
q' = new paths for commodity k'.

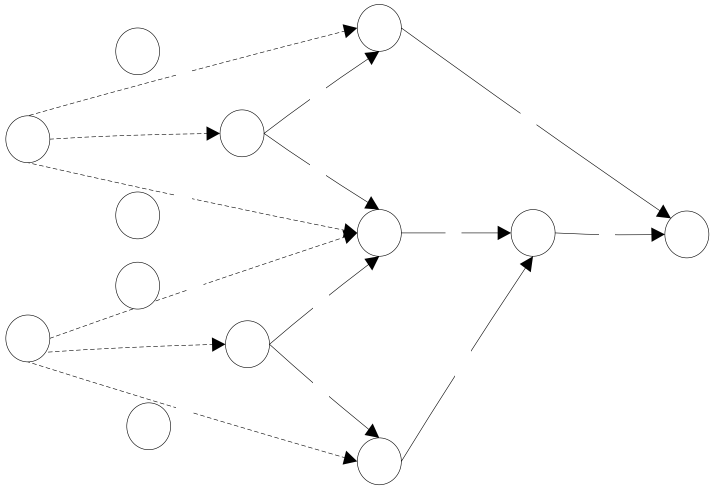

Figure 6.2. Routing costs changed

Figure 6.2 illustrates how to calculate the change in cost. Assume given demands are originated at source terminals 2 and 4. Nodes 8 and 9 are dummy nodes associated with source terminals 2 and 4. Commodities f87 and f86 actually are commodities originated from source terminal 2 and commodities f97 and f96 are originated from source terminal 4. Arc costs of arcs that link source terminals to connected terminals are equal to costs of the arcs that connect associated dummy nodes to the connected terminal, indicated as shown. Sorting costs are assumed to be 1 per shipment for all terminals. Suppose in the initial solution, f87 = 20 on path 8-1-7; 50 f86 = 50 on path 8-3-6; f97 = 30 on path 9-3-7; f96 = 20 on path 9-5-6 as shown in Figure 6.2. We can examine the changes in cost as follows.

Initial solution

Node 8:
f87: RC = 20*2+20*2 = 80; SC = 20*1 = 20.
f86: RC = 50*2+50*2 = 200; SC = 50*1 = 50.
Sorting cost at source terminal 2 = 70*1 = 70.
Total costs = 80+20+200+50+70 = 420.
Node 9:
f97: RC = 30*3 + 30*2 +30*1 = 180; SC = 30*1+30*1=60.
f96: RC = 20*1 + 20*1 = 40; SC = 20*1 = 20.
Sorting cost at source terminal = 50*1 = 50.

f87=20

f86=50 8

Total costs = 180+60+40+20+50 = 350.
Examination

Node 8:
To avoid sorting costs of 70 at source node 2, f87 could be moved to arc83 and use the same routing as that of f97, beginning from node 3, which is 8-3-6-7.
Moving f87 from path 8-1-7 to 8-3-6-7:
f87: RC = 20*2 + 20*2 +20*1 = 100; SC = 20*1+20*1=40.
f86: RC = 200; SC = 50
Total costs = 100+40+200+50 = 390.
Possible savings from switching to the non-sort option = 420-390 = 30.
Node 9:
To avoid sorting costs at source node 4, f96 could be moved to arc93 and use the same routing as that of f97, beginning from node 3, which is 9-3-6.
Moving f96 from path 9-5-6 to 9-3-6:
f96: RC = 20*3 +20*2 = 100; SC = 20*1=20.
f97: RC = 180; SC = 60.
Total costs = 100+20+180+60 = 360.

Since total costs of using the all-sort option are lower than the non-sort option, the all sort option will be selected for source terminal 4.

All source terminals will be examined if switching to the non-sort option can possibly reduce costs. For the source terminal that total costs can be reduced, the binary variables associated with the connected arc will be fixed to one. The heuristic schema is incorporated into the GH and FCH presented in Section 5, referred to as the EGH and the EFCH.

The steps of the procedure are as follows.
1. Apply GH/FCH to acquire a feasible solution.
2. For local terminals whose demands are split, if

$$.\ \sum_{k'} \sum_{q'} RC_{q'} f_{q'}^{k'} + \sum_{i} \sum_{q'} SC_i w_{iq'} f_{q'}^{k'} - \Big(\sum_{k'} \sum_{q} RC_q f_q^{k'} + \sum_{i} \sum_{q} SC_i w_{iq} f_q^{k'}\Big) \le 0$$

where,
$RC_{q'}$ = routing cost of using path q'.
SC_i = sorting cost at terminal i.
k' = commodities to be moved at node m.
q = original paths of commodities k' obtained from the solution.
q' = new paths for commodities k'.
, fix x_{mi} = 1 for terminal i which all flows are routed to.

The set of new paths *q'* could be derived as follows.

i). If there are commodities with the same destination at the terminal that flows are moved to, new paths are paths that those commodities use in the solution.

ii). If commodities with the same destination at the terminal do not exist in the solution, new paths are possible paths that skip all nodes.
3. Re-solve the problem using GH/FCH.

In summary, we obtain an initial feasible solution by solving the problem with relaxing binary restrictions using the GH or FCH. Then, sorting costs are imposed for source terminals whose demands use two connected terminals in their routings. From the initial solution, we examine whether potential savings can be produced by switching for the all-sort option to non-sort option. For source terminals with potential savings, binary variables associated with the arc used in routing all flows will be fixed to one. Then, the problem will be resolved again using the GH or the FCH.

6.3 Experimental Approach

As the all sort/non sort options are added, we need to incorporate this component into lower bound calculation so as to benchmark the heuristic. This section will describe how lower bounds are calculated for the model extension as well as test problems.

6.3.1 Lower Bound

Lower bounds are calculated using the LP relaxation with forcing constraints as done with the original model with a slight modification to capture the all sort/non sort options. The added component is treated by estimating the costs changes in a similar manner as in the heuristic for solution improvement, but without resolving the problem after examination.

First, the LP relaxation with forcing constraints is computed. Then, for each source node whose demands use both connected terminals in their routing the potential saving is calculated by estimating cost changes. If

$$\sum_{k'} \sum_{q'} RC_{q'} f_{q'}^{k'} + \sum_{i} \sum_{q'} SC_i w_{iq'} f_{q'}^{k'} - (\sum_{k'} \sum_{q} RC_q f_q^{k'} + \sum_{i} \sum_{q} SC_i w_{iq} f_q^{k'}) \leq 0$$

, the demands will use a single connected terminal for the source node so that sort costs at the source terminal will not be imposed.
LB =

$$O - \sum_{m} \max \{ (\sum_{k'} \sum_{q} RC_q f_q^{k'} + \sum_{i} \sum_{q} SC_i w_{iq} f_q^{k'} - \sum_{k'} \sum_{q'} RC_{q'} f_{q'}^{k'} - \sum_{i} \sum_{q'} SC_i w_{iq'} f_{q'}^{k'}), 0\}$$

where,
O = Objective value of the LP relaxation with forcing constraints.
$RC_{q'}$ = routing cost of using path q'.
SC_i = sorting cost at terminal i.
m = dummy nodes m associated with origin terminals s.
k' = commodities to be moved at node m.
q = original paths of commodity k' obtained from the solution.
q' = new paths for commodity k'.

6.3.2 Problem Instances

Test problems are created from the network used in the experiments for the core model in Section 5, but with the addition of duplicated nodes of hubs, dummy nodes of local terminals and dummy arcs as follows. To obtain the solutions within reasonable time, we tested the network size of 25 nodes to 108 nodes.

Network Size	# National Hub	#Regional Hubs	# Major Airports	# Local Airports	# Rural Airports	# of Local Terminals
25	1	2	2	15	0	5
45	1	2	3	19	10	10
65	1	2	4	25	18	15
83	1	2	5	30	25	20
108	1	3	7	35	32	30

Table 6.1. Number of nodes of each terminal type for different network sizes

6.4 Experimental Results

6.4.1 Solution Quality VS. Computation Time

The percentage closeness to lower bound and computation time of the EGH and the EFCH are shown in Figures 6.3 and 6.4.

As shown in Figure 6.3, the percentage closeness to lower bound of the EFCH and the EGH ranges from 94.35% to 96.38% and 92.66 to 94.25%, respectively. For computation time, the EGH performs much faster than the EFCH. These results are consistent with the results of the solution quality and speed of the GH and FCH experimented in Section 5. In terms of the solution quality, the EFCH and the EGH yield lower percentage closeness to lower bound than the FCH and GH. This is because the heuristic schema developed could produce errors in selecting nodes to use the non-sort option under the examination of moving flows in the beginning steps of the EFCH/EGH. The heuristic also degrades lower bound tightness, which, in turn, results in lower percentage closeness to lower bound. As for computation time, the EFCH and the EGH take approximately two times the calculation time used by the FCH and the GH under the same network size. The computation time is as expected because the EFCH and the EGH resolve the problems again after the examination step.

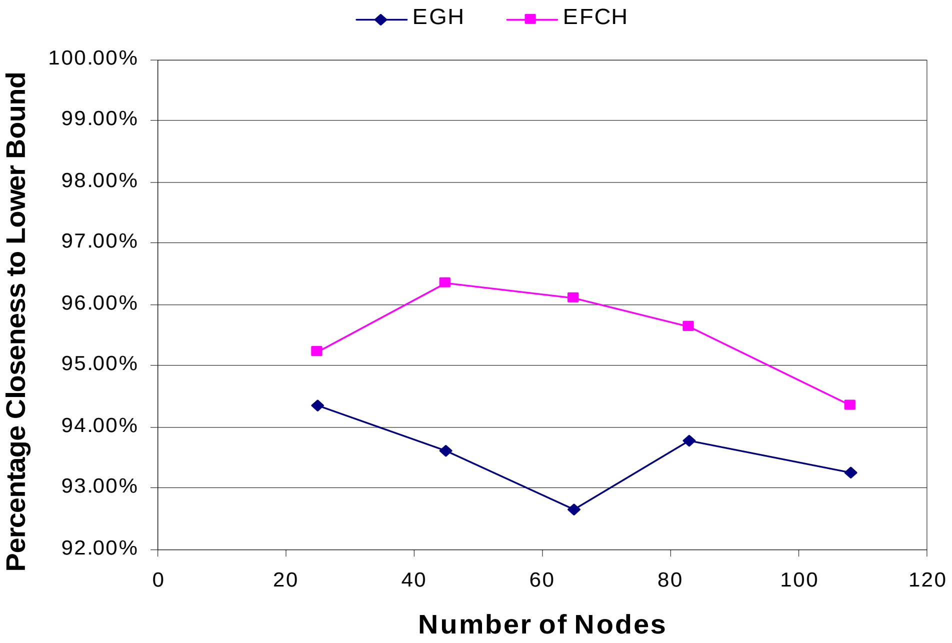

Figure 6.3. Percentage Closeness to LB of the EGH and the EFCH

Note: For the EFCH, alpha value of 0.8 was applied for network sizes larger than 50 nodes.

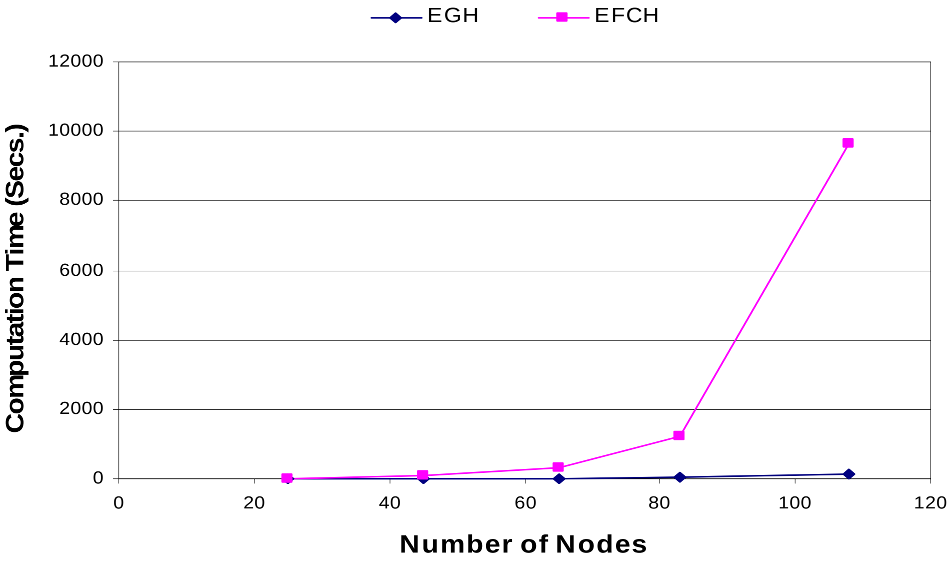

Figure 6.4. Computation Time of the EGH and EFCH

6.4.2 Sensitivity analysis

6.4.2.1 Sorting cost

The number of source terminals that use the non-sort option affects the algorithm performance. We evaluate the effects by increasing sorting costs at source terminals and comparing the percentage closeness to lower bounds among different percentage increments. The network size with 65 nodes was chosen to conduct the experiments.

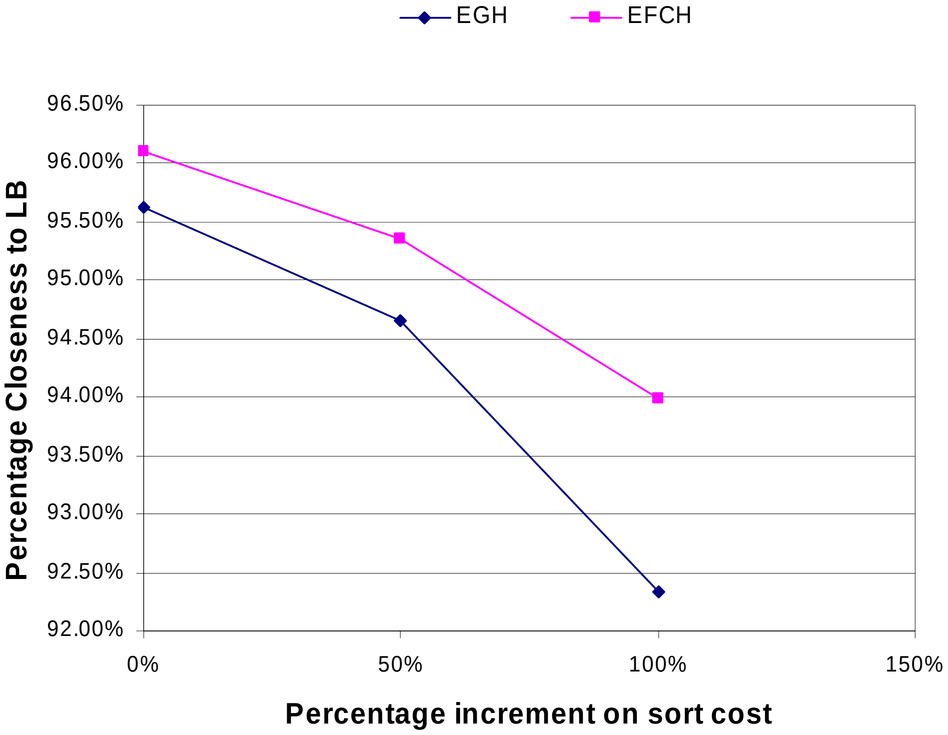

Figure 6.5. Percentage closeness to LB vs. percentage increment on sort cost at source nodes

As shown in Figure 6.5, percentage closeness to lower bound declines when the percentage increment increases. A reason for the deficiency of algorithm performance is that lower bound performance deteriorates when more source terminals use the non-sort option. The performance of lower bounds under different percentage increments are benchmarked with the objective values of optimal solutions obtained from CPLEX 8.0 under the network size of 25 nodes. The results are shown as below.

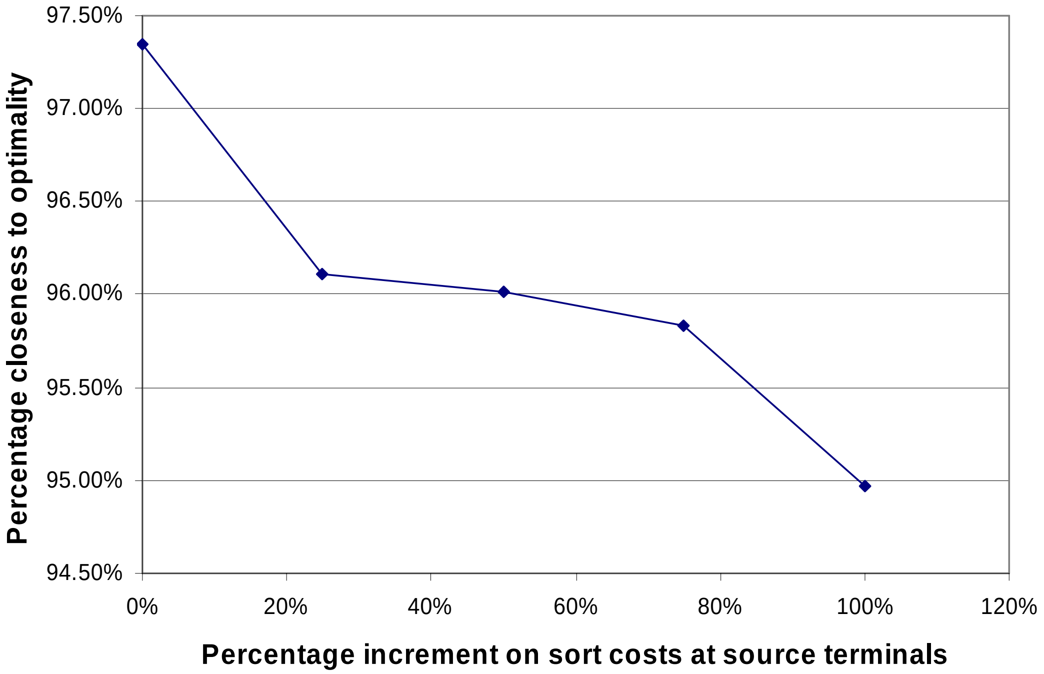

Figure 6.6. Percentage closeness to optimality vs. percentage increment on sort costs

From Figure 6.6, it can be seen that the lower bound performance deteriorates when the percentage increment increases because, when sort costs at source terminals increase, more source nodes tend to use the non-sort option. Since, in calculating lower bounds, the paths of moved flows for the non-sort option are not examined for feasibility of aircraft capacity. As a result, more source terminals using the non-sort option mean more paths will violate aircraft capacity constraints and make lower bounds further from optimality. Another reason of the deterioration of the heuristic performance is similar to the reason for the deficiency of lower bound performance. When sort costs increase, more source terminals will select the non-sort option. The number of nodes using the non-sort option for network size of 65 nodes, which has 15 local terminal nodes, is shown in Table 6.2.

Sort cost increment at source terminals (%)	EGH	EFCH	LB
0	3	3	3
50	5	6	6
100	9	11	12

Table 6.2. Sort cost increment and number of source terminals using the non-sort option

More source nodes, with the non-sort option, could produce larger errors in the examination of switching flows from the all-sort option to the non-sort option in that the number of nodes that use the non-sort option, obtained from the examination, may not be the same as in the optimal solution. These errors cause the deterioration of the heuristic performance.

6.5 Summary

In this section, we extended the original model by accounting for capacity by time periods and the all-sort/non-sort options for source terminals. Capacity by time period component is presented to capture the congestion of incoming shipments in peak and non-peak periods of sorting facility so as to maximize benefits of containerization. This component is treated by duplicating hub nodes to indicate peak and non-peak periods with different capacities and sort costs. The all-sort/non-sort options are presented to provide sorting options at source terminals. The all-sort option has the advantage of sorting incoming shipments so that they can be directed to their cheapest routes with potential bypasses over subsequent terminals. The non-sort option, however, could save fixed operation costs and variable sorting costs at the source terminals and also expedite shipment transfer. We captured this component by duplicating dummy nodes for each source terminal and dummy arcs associated with connected nodes of source terminals. A heuristic schema incorporated into the GH and FCH was developed to handle these components. The results show that the EFCH performs better than the EGH but the computation time of the EGH is much faster as in the experiments in Section 5. The performance of both heuristics deteriorates when sort costs at source terminals increase because more source nodes tends to use the non-sort option, leading to more errors under examination of switch flows from using the all-sort option to the non-sort option. In addition, lower bounds degrade when sort costs increase since more flows will violate aircraft capacity constraints, making the lower bounds further from optimal solutions. This also affects the percentage closeness to lower bounds of the EGH and EFCH.

7. Conclusion and Future Research

7.1 Conclusion

This dissertation addresses the package transportation problem that accounts for sorting costs and sorting capacity constraints. The problem we consider could be segmented into two components: transportation and sorting. The transportation component is a multi-commodity problem in which a large number of shipments are originated and destined to different locations over the whole network. These packages are transported in the form of containers. At each terminal visited by shipments, sorting costs are imposed on arrival shipments if the shipments in the same containers need to be directed to different locations, meaning that they use sorting facility at the terminal. Thus, the sorting component is a problem of how to containerize shipments to save sorting costs under limited sorting capacity. We consider an environment in which sorting capacity for each terminal is assumed to be a single aggregate value, measured in number of shipments. The objective is to minimize total costs of transportation costs and sorting costs.

The problem is modeled as a MIP path-based formulation in which continuous variables represent commodity flows on their possible routing paths and integer variables indicate container groups used for routing commodity flows. To account for sorting activities at terminals, the original network is modified by creating artificial arcs to represent bypass container groups. Based on this formulation, three heuristic approaches are constructed to minimize total transportation and sorting costs: the grouping heuristic (GH), the forcing constraint heuristic (FCH), and the combined heuristic (CH). The GH is based on a greedy algorithm in which shipment grouping decisions are made iteratively, based on highest sort savings. The FCH exploits forcing constraints to tighten the LP relaxation and acquire feasible MIP solutions. The CH is the combination of the GH and the FCH to balance computation time and solution quality.

A lower bound, derived by computing the LP relaxation with forcing constraints, is used to benchmark the developed heuristics. The lower bound is investigated by comparing lower bounds for small test problems with optimal solutions obtained from CPLEX 8.0. The results show that the lower bounds for based cases are within 1% of optimality.

The heuristics are assessed in terms of solution quality and computation time under different network sizes, ranged from 20 to 112 nodes. The benchmarking results show that solutions obtained from the GH are within 4% from lower bounds for base cased and computation time for the GH is comparable to solving the LP relaxation alone. The FCH is relatively slow due to a large set of forcing constraints added. To handle large size problems within reasonable time, we introduce an alpha value representing the proportion of container capacity. Container variables whose maximum flows are not up to container capacity multiplied by the alpha value are eliminated from the formulation since they not likely to be in optimal solutions. Incorporated with the alpha value for large size problems, the FCH outperforms the GH in terms of solution quality in that solutions obtained from the FCH are within 2% of lower bounds for most cases. For the CH, computation time lies in between the GH and the FCH as expected. However, solution quality is poorer than the first two heuristics and shows signs of deterioration for large size problems.

The GH can perform relatively fast because the network to be solved is the original network, which does not include a large set of artificial arcs created. Compared to the FCH and the CH, the speed of the GH is even more pronounced as the network size is larger than 50 nodes. The FCH yields the best solution quality among the three heuristics by virtue of forcing constraints inserted. Serving as additional cuts, forcing constraints drive commodity flows to take larger values so that most flows are large enough to be assigned to containers on their optimal paths.

The sensitivity analysis is provided to investigate effects of container capacity and increase in sorting costs on the GH and the FCH. Container capacity shows no signs of significant changes in solution quality for both heuristics. Increase in sorting costs is found to have major effects on the solution quality. The results show that the solution deteriorates as sorting costs increase.

The core model is extended to capture the congestion of hubs in peak periods and all-sort/non-sort options for source terminals where no shipments are allowed to be transshipped. Congested periods of hubs are modeled by duplicating hub nodes to represent peak period nodes with limited sort capacities and high sort costs. The sort options of source terminals are all-sort and non-sort options, indicating if originating shipments are all sorted or all routed to a single connected terminal without being sorted. This component is treated by duplicating source terminals and creating artificial arcs with associated binary integer variables to specify sort options.

Solving the extended model begins with acquiring an initial feasible solution. The extended problem with relaxing binary integer restrictions is first solved by the GH or the FCH. Then sort costs are imposed on source terminals in which originating flows are split and routed to two connected terminals. To improve the initial feasible solution, we develop a heuristic approach, based on examination of the sort-options at source terminals. For each terminal, we examine if switching from the all-sort to the non-sort option could make potential savings. With the associated binary variables fixed to 1 for the selected sort option, the problem is resolved again using the GH or FCH. The objective values are benchmarked with lower bounds calculated in the same manner but without resolving the problem. Experimental results correspond to the results with the core model in which the FCH outperforms the GH in terms of solution quality and the GH is relatively faster than the FCH. Lower bound performance deteriorates when the sort costs at source terminals increase because more source terminals will select the non-sort option leading to larger errors in costs changed at the examination step.

In summary, the GH is the fastest algorithm and provides solutions within 4% of lower bounds for test cases. The FCH outperforms the GH in terms of solution quality, in which solutions are within 2% of lower bounds for test cases, but it is slower than the GH. As for the CH, computation time is between the GH and the FCH but it yields inferior objective values relative to the FCH and the GH. For the model extension, the results correspond to the results with the core model in the sense of incorporation with the GH and the FCH. Increase in sorting costs at source terminals impacts the heuristic performance on solution quality; the performance deteriorates when the number of nodes that should be switched is increased, resulting in more error.

7.2 Directions of Future Research

Future research can be conceived in two main directions in modeling: extension on capacity by time period model and detailed model accounting for queuing process. The first direction could be accomplished by considering interactions between the non-sort option and the congestion at subsequent terminals. The latter option is a model considering queuing process of arrival shipments into the model.

In this research, our core model assumes that sort capacity for each terminal is a single aggregate value. In the model extension, we account for the congestion of processed shipments by modeling sort capacity by time periods at hubs and also consider the all sort/non-sort options at source terminals. In the model, the advantage of the non-sort option is captured only by producing sort savings at source terminals. Another advantage of the non-sort option is to expedite transfer to next terminals. This feature in fact could have effects on subsequent terminals in such a way that quicker transfer allows shipments to arrive the next terminals earlier and could be processed in non-peak periods to avoid the congestion in peak periods. This interaction could be captured in the model by enabling packages that are originated from source terminals with the non-sort option to have more possible routings such that congestion in peak periods at hubs could be allayed.

To better conform to real world scenarios, queuing process of arrival shipments could be taken into account. With this modeling, the assumption of prescheduled flights is no longer held since time incurred in containerization could affect departure schedules at the terminals processing shipments and arrival schedules at subsequent terminals. Decisions of containerization at terminals in the network would interact with arrivals at all other terminals. As shipment arrivals depend on the decisions of containerization, congested periods of sorting facilities become non-deterministic. Thus, sorting capacity should be in the form of a processing rate, measured in the number of shipments per time unit. The queue of incoming shipments would impact processing time and completion time.

9. References

Assad A. A. (1980), Models for Rail Transportation, *Transp. Res. A*, **14**, 205-220.

Assad A. A. (1983), Analysis of Rail Classification Policies, *INFOR*, **21**, 293-314.

Amiouny S.V., Bartholdi J. J., Vande Vate J. H., and Zhang J. (1992), Balanced loading, *Opns. Res.*, **40**, 2, 238-246.

Barnhart C., Boland N.L., Clarke L.W., Johnson E.L., Nemhauser G.L., and Shenoi R.G. (1998a), Flight String Models for Aircraft Fleeting and Routing, *Transp. Sci.*, **32**, 3, 208-220.

Barnhart C., Hane, C. A., and Vance P. H. (2000), Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems, *Opns. Res.*, **48**, 2, 318-326.

Barnhart C., Johnson E. L., Nemhauser G.L., Savelsbergh M.W.P., and Vance P. H. (1998b), Branch-and-Price: Column Generation for Solving Huge Integer Programs, *Opns. Res.*, **46**, 3, 316-329.

Barnhart C. and Schneur R.R. (1996), Air Network Design for Express Shipment Service, *Opns. Res.*, **44**, 852-863.

Barhart C. and Sheffi Y. (1993), A network-based primal-dual heuristic for the solution of multicommodity network flow problems, *Trans. Sci.*, **27**, 2, 102-117.

Bodin L. D., Golden B.L., and Schuster A.D. (1980), A Model for the Blocking of Trains, *Transp. Res.*, **14**, 115-121.

Bostel N. and Dejax P. (1998), Models and Algorithms for Container Allocation Problems on Trains in a Rapid Trasshipment Shunting Yard, *Transp. Sci.*, **32**, 4, 370-379.

Cheung R.K. and Muralidharan B. (2000), Dynamic Routing for Priority Shipments in LTL service Networks. *Transp. Sci.*, **34**, 1, 86-98.

Cheung W., Leung L.C., and Wong Y.M. (2001), Strategic Service Network Design for DHL Hong Kong, *Interfaces*, **31**, 1-14.

Cranic G. T., Ferland J. A., and Rousseau J. (1984), A Tactical Planning Model for Rail Freight Transportation, *Transp. Sci.*, **18**, 165-184.

Cranic G. T. and Rousseau J. (1986), The Column Generation Principle and The Airline Crew Scheduling Problem, *INFOR*, **25**, 2, 136-151.

Daganzo C.F. (1986), Static Blocking at Railyards: Sorting Implications and Track Requirements, *Transp. Sci*., **20**, 3, 189-199.

Dantzig G.B. and Wolfe P. (1960), Decomposition Principle for Linear Programs, *Opns. Res*., **8**, 101-111.

Desrosiers J., Soumis F., and Desrochers M. (1984), Routing with Time Windows by Column Generation, *NETWORKS*, **14**, 545-565.

Geinzer C.M. and Meszaros C.M. (1990), Modeling High Volume Conveyor Sorting Systems. Proceedings of the 1990 Winter Simulation Conference, O. Balci, R.P. Sadowski and R.E. Nance (eds.), 714-719.

Glickman, T. S, and Sherali, H. D. (1984), Large-Scale Network Distribution of Pooled Empty Freight Cars over Time, with Limited Substitution and Equitable Benefits, *Trans. Res. B*, **19**, 2, 85-94.

Grünert T. and Sebastian H. (2000), Planning Models for Long-haul Operations of Postal and Express Shipment Companies, *European Journal Of Operational Research* , **122**, 2, 289-309.

Haessler R. W. and Talbot F. B. (1990), Load Planning for Shipments of Low Density Products, *European Journal of Operations Research*, **44**, 289-299.

Haghani A. E. (1989), Formulation and Solution of a Combined Train Routing and Makeup, and Empty Car Distribution Model, *Trans. Res. B*, **23**, 6, 433-452.

Hall R. W. (1989), Configuration of an Overnight Package air network, *Transp. Res. A*, **23**, 2, 139-149.

Heidelberg K. R. (1998), Automated Air Loading, *Naval Research Logistics*, **45**, 751-768.

Keaton, M.H. (1989), Designing Optimal Railroad Operating Plans: Lagrangian Relaxation and Heuristic Approaches. *Transp. Res. B,* **23**,6, 415-431.

Keaton, M.H. (1992), Designing Optimal Railroad Operating Plans: A Dual Adjustment Method for Implementating Lagrangian Relaxation. *Transp. Sci*., **26**, 263-279.

Kim D., Barnhart C., Ware K., and Reinhardt (1999), Multimodal Express Package Delivery: A Service Network Design Application, *Transp. Sci.,* **33**, 4, 391-407.

Kuby M.J. and Gray R.G. (1993), The Hub Network Design Problem with Stopovers and Feeders: The Case of Federal Express, *Transp. Res. A*, **27**, 1, 1-12.

Laporte G., Louveaux F. and Mercure (1989), The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms, *European Journal of Operations Research*, **59**, 345-358.

Martinelli D. and Teng H. (1994), Neural Network Approach for Solving the Train Formation Problem, *Trans. Res. Record*, **1470**, 62-69.

Martinelli D. and Teng H. (1995), A Genetic Algorithm Approach for Solving the Train Formation Problem, *Trans. Res. Record*, **1470**, 38-46.

Newton H.N., Barnhart C., and Vance P. H.(1998), Constructing Railroad Bloking Plans to Minimize Handling Costs, *Transp. Sci*., **32**, 4, 330-345.

Nobert Y. and Roy J. (1998), Freight Handling Personnel Scheduling at Air Cargo Terminals, *Transp. Sci*., **32**, 2, 295-301.

Petersen E. R. and Fullerton H.V. (1975), The Railroad Network Model, *Report No. 75-11*, Queen's University, Kingston, Ontario, Canada.

Philip L.T. (1987), Air Carrier Activity at Major Hub Airports and Changing Interline Practices in the United States' Airline Industry. *Transp. Res. A*, **21**, 215-221.

Rexing B., Barnhart C., Kniker T., Jarrah A, and Krishnamurthy N. (2000), Airline Fleet Assignment with Time Windows, *Transp. Sci*., **34**, 1, 1-20.

Thomet M. A. (1971), A User-oriented Freight Railroad Operating Policy, *IEEE Transaction on Systems, Man and Cybernetics*, **1**, 4, 349-356.