| REPORT DOCUMENTATION PAGE | Form Approved OMB No. 0704-0188 |
|---|---|



| 1. AGENCY USE ONLY *(Leave blank)* | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | 9 Jan 97 | |

**4. TITLE AND SUBTITLE**
Nework design Under Budget Constraints With Application To The Railroad Blocking Problem

**5. FUNDING NUMBERS**

**6. AUTHOR(S)**
Harry N. Newton

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Auburn University

**8. PERFORMING ORGANIZATION REPORT NUMBER**
96-37D

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
DEPARTMENT OF THE AIR FORCE
AFIT/CIA
2950 P STREET
WPAFB OH 45433-7765

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**

**12a. DISTRIBUTION AVAILABILITY STATEMENT**
Unlimited

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** *(Maximum 200 words)*





| 14. SUBJECT TERMS | 15. NUMBER OF PAGES |
|---|---|
| | 106 |
| | 16. PRICE CODE |

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| | | | |

# Network Design Under Budget Constraints with Application to the Railroad Blocking Problem

Harry N. Newton

Certificate of Approval:

| | |
|---|---|
| Robert L. Bulfin<br>Professor<br>Industrial and Systems Engineering | Pamela H. Vance, Chair<br>Assistant Professor<br>Industrial and Systems Engineering |
| Carolyn L. McCreary<br>Associate Professor<br>Computer Sciences and Engineering | John F. Pritchett<br>Dean<br>Graduate School |

Style manual or journal used IIE TRANSACTIONS

Computer software used *The document preparation package TEX (specifically LATEX) together with the departmental style-file* `aums.sty`.

# Network Design Under Budget Constraints with Application to the Railroad Blocking Problem

Harry N. Newton

A Dissertation

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Doctor of Philosophy

Auburn, Alabama

December 13, 1996

## VITA

Harry Nelson Newton was born November 19, 1963, in Charleston, SC. In 1985, he completed dual Bachelor of Science degrees at Clemson University in Mathematical Sciences and Computer Science. At graduation, he was commissioned as an officer in the United States Air Force as a distinguished graduate from Clemson's Reserve Officer Training Corps. As a lieutenant, Harry served as an operations research analyst at the Air Force Logistics Management Agency from 1985-1989 and was selected by the Air Force for sponsorship of a Master's degree in Mathematics. In 1989, Harry was promoted to captain and received his Master's degree from University of Texas with research in an alternative system for the Air Force's inventory control under the direction of Professor Emeritus Abram Charnes. Harry was selected to teach in the Mathematical Sciences department at the United States Air Force Academy and subsequently selected for sponsorship of his doctorate in 1993. In 1996, he was selected for promotion to Major and completed the degree requirements for his doctorate in Industrial and Systems Engineering at Auburn University. He will return to the United States Air Force Academy as an assistant professor in August 1996.

DISSERTATION ABSTRACT

NETWORK DESIGN UNDER BUDGET CONSTRAINTS WITH

APPLICATION TO THE RAILROAD BLOCKING PROBLEM

Harry N. Newton

Doctor of Philosophy, December 13, 1996
(M.A., University of Texas, 1989)
(B.S., Clemson University, 1985)
(B.S., Clemson University, 1985)

115 Typed Pages

Directed by Pamela H. Vance

We develop a column generation-based, branch-and-bound algorithm for the directed network budget design problem (BDP), also known as the optimal network design problem, when additional budget constraints are placed on some nodes. Our pricing subproblems identify new paths for the commodities using a shortest path algorithm. We model the railroad blocking problem (RBP) as a BDP with constraints on the capacities at the nodes and restrictions on the legal paths for each commodity. In RBP, the physical rail network (the railroad terminals and tracks) is already defined. The "blocking network" to be constructed is a virtual network that is overlayed on the physical network. The blocks are virtual "express" arcs which a commodity may take to have uninterrupted service between two terminals that are not necessarily connected by a physical link. Solving RBP requires specifying the blocking network and assigning paths through the blocking network for each commodity. Our solution technique to RBP is unique in constraining the use of resources at the nodes and allowing multiple priority classes for

the commodities. Computational results for our algorithm show solutions within 2.3% of a known lower bound for a real-world RBP instance (with 150 nodes, 1300 commodities, and up to 6,800 candidate arcs after preprocessing) for a large domestic railroad and within 5.5% for randomly-generated instances of the same size.

## Acknowledgments

This work is dedicated to my wife whose support and encouragement have made it possible. My thanks go to Colonel Daniel W. Litwhiler and Colonel Steven C. Gordon for nominating me for sponsorship by the United States Air Force for a doctoral degree.

Several people have contributed to the success of the project which this dissertation represents. My advisor, Pamela Vance, deserves special note for supervising this research while on maternity leave. I thank CSX Transportation (CSXT) for providing the required data and especially thank CSXT's Dharma Acharya for his technical assistance and encouragement and David Rinker for his help interfacing with CSXT's computer systems.

My friends and family also have my gratitude for their encouragement and prayers.

TABLE OF CONTENTS

## List of Figures

# Chapter 1

# The Budget Design Problem

## 1.1 Introduction

In the network budget design problem (BDP) the cost of flowing a set of commodities through a network is minimized while observing budget constraints on the fixed costs of the links used. Applications of this problem are diverse and include problems such as deciding what transportation or communication infrastructure to build between cities or deciding on the topology of a computer network. BDP can be used to construct a new network or modify an existing one. The links may be either directed or undirected.

In its simplest form, BDP includes balance equations for the flow of each commodity, a budget constraint on the sum of fixed costs of the links selected, and "bundle" constraints to ensure that flow is allowed only on arcs which are built and that the maximum capacity of each arc is observed. Depending on the application, there may be other constraints, such as service level constraints or requirements on the reliability or survivability of the network.

Our motivation for studying BDP is an application in the rail industry referred to as the railroad blocking problem (RBP) where the fixed cost of offering direct service between two *yards* involves dedicating a sorting track at the origin yard to accumulate railcars into *blocks* which are headed next for a particular destination yard. In this application, there is a separate budget constraint for each yard (node) based on its sorting tracks. We refer to these constraints as *node-budget* constraints. Other side constraints are needed to model service constraints on the number of times a commodity

may be sorted and flow constraints at each node to model the number of railcars which may be sorted at each yard in any given period. The solution procedure we propose for BDP is flexible enough to allow these additional constraints. Other applications with similar node-budget constraints arise in the trucking and airline industries where there is a budget constraint at each city due to limited docks or departure gates.

BDP is related to several other problems, such as, the fixed-charge network design problem and the multi-commodity flow problem. The fixed charge network design problem is different in that the fixed costs for the arcs appear in the objective function (minimize the total fixed and variable costs) and not in the constraints. The multi-commodity flow problem is a BDP with the binary arc selection variables fixed.

The contribution of this thesis is modeling and developing a branch and price algorithm for the BDP and RBP. Our solution for the column generation subproblems generates attractive paths for each commodity by solving a shortest path problem. We implement our algorithm for the RBP and demonstrate its usefulness with computational results for problems with with 150 nodes, 6000 potential arcs, and 1300 commodities. Components of our algorithm for RBP include: an LP-based column-generation procedure, a primal heuristic to aid in finding integer solutions, a shortest path-based lower bound, and a branching rule. Our solution to RBP incorporates priority constraints on the number of arcs used in delivering each commodity which allows simultaneous solutions for the RBP for express and non-express traffic. Current practice is to solve RBP for the express traffic independently from the non-express traffic. For the test problems, we find solutions in several hours on a workstation-class computer that are within a few percent of a known lower bound.

## 1.2 Node-arc formulation

We first present a node-arc Mixed Integer Program (MIP) for BDP and show that the constraint matrix for this formulation is too large for the problems we wish to solve. Using Dantzig-Wolfe decomposition, we form an equivalent path-based formulation which has fewer constraints but more variables.

### Parameters

$G = (N, A)$ is the graph with node set $N$ and candidate arc set $A$.

$K$ is the set of all commodities $k$ designated by an origin-destination pair of nodes.

$v^k$ is the volume of commodity $k$ (in consistent units).

$orig(k)$ is the origin node for commodity $k$. $orig(a)$ is the origin of arc $a$.

$dest(k)$ is the destination node for commodity $k$. $dest(a)$ is the destination of arc $a$.

$u_a$ is the capacity of arc $a$.

$c_a \geq 0$ is the per unit cost of flow on arc $a$ (assumed equal for all commodities).

$e_a$ is the fixed cost for including arc $a$ in the network.

$B$ is the budget for fixed costs for the entire network.

$B(i)$ is the budget for fixed costs of arcs leaving $i$.

$\Delta(k)$ is the set of constraints on legal paths for commodity $k$.

$Q(k)$ is the set of extreme points of $\Delta(k)$.

### Variables

$x_a^k$ is the proportion of commodity $k$'s volume flowing on arc $a$.

$$y_a = \begin{cases} 1 & \text{if arc } a \text{ is included in the network,} \\ 0 & \text{otherwise.} \end{cases}$$

**Formulation**

$$\text{(NODE)} \quad min \quad \sum_{k \in K} \sum_{a \in A} c_a v_k x_a^k \tag{1.1a}$$

s.t.

$$\text{Bundle} \quad \sum_{k \in K} v_k x_a^k \leq u_a y_a \qquad \forall a \in A \tag{1.1b}$$

$$\text{Balance} \quad \sum_{\substack{a \in A \\ orig(a)=i}} x_a^k - \sum_{\substack{a \in A \\ dest(a)=i}} x_a^k = \begin{cases} 1 & orig(k) = i \\ -1 & dest(k) = i \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in N, k \in K \tag{1.1c}$$

$$\text{Budget} \quad \sum_{a \in A} e_a y_a \leq B \tag{1.1d}$$

$$\text{Node Budget} \quad \sum_{\substack{a \in A \\ orig(a)=i}} e_a y_a \leq B(i) \qquad \forall i \in N \tag{1.1e}$$

$$y_a \in \{0,1\} \qquad \forall a \in A$$

$$0 \leq x_a^k \leq 1 \qquad \forall a \in A, k \in K$$

The objective is to minimize the sum of the costs of delivering each commodity using the network formed by arcs for which $y_a = 1$. For each arc, constraints (1.1b) prevent flow on arcs which are not built and enforce the upper bound $u_a$ on flow for arcs which are built. For each node, constraints (1.1c) are balance equations for the flow of each commodity and constraints (1.1e) enforce the node-budget limit $B(i)$ for the sum of the fixed costs of the arcs which leave the node. The single budget constraint (1.1d) limits the sum of the fixed costs $e_a$ for all arcs selected for the network to the total budget $B$.

As is typical with node-arc formulations, NODE has a large number of constraints and variables. Let $|N|$ denote the number of nodes, $|K|$, the number of commodities, and $|A|$, the number of possible arcs. Since there is one mass balance constraint for each node-commodity pair, there will be $|N||K|$ equality constraints of form (1.1c) with

a non-zero coefficient for each potential arc which is adjacent to the node. There will be $|A|$ bundle constraints (1.1b), one for each arc, and $|N|$ node-budget constraints (1.1e), one for each node. Adding the single (1.1d) budget constraint gives a total of $|N||K|+|A|+|N|+1$ constraints, $|A||K|$ continuous flow variables $x_a^k$ (one for the flow of each commodity on each arc) and $|A|$ binary selection variables $y_a$. For a network with 100 nodes, 1000 commodities, and 5,000 arcs (half of the possible 10,000 node pairs), there would be:

- 105,101 constraints,
- 5,000,000 continuous variables, and
- 5,000 binary variables.

The railroad problem that we wish to solve is even larger: 150 nodes, 1,300 commodities, and 6,000 arcs. Other constraints that are needed in the railroad problem will further increase the size. Given the huge size of the constraint set, storing and manipulating this formulation becomes difficult on a workstation. Hence, we turn to a path-based formulation.

### 1.3 Path-based MIP formulation

In NODE, if we list the balance equations (1.1c) grouped by commodity with the $x_a^k$ variables also grouped by commodity, then the nonzero $x_a^k$ coefficients have a block angular structure as shown in Figure 1.1. This structure lends itself to a Dantzig-Wolfe decomposition scheme where we express the flow variables $x_a^k$ as convex combinations of the extreme points of the polyhedra formed by the balance equations for each commodity.

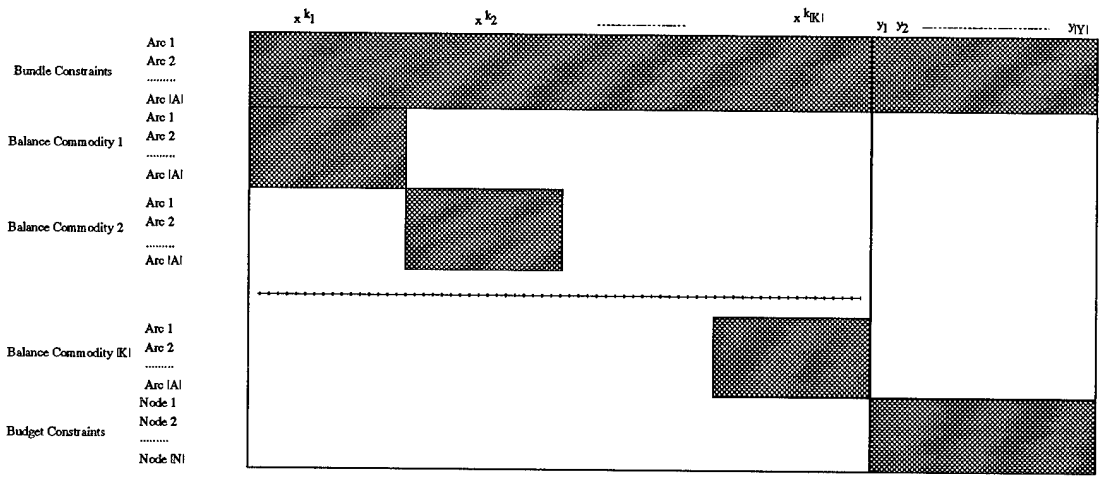


Figure 1.1: Structure of constraint coefficient matrix for NODE

In this section, we perform this decomposition, resulting in a path-based formulation equivalent to NODE but with many fewer constraints. However, the number of path variables in this new formulation will, in general, be exponential in the number of nodes.

We introduce the following additional notation to address the change to path variables.

### Parameters

$\Delta(k)$ is the set of constraints on legal paths for commodity $k$.

$Q(k)$ is the set of all extreme points of $\Delta(k)$.

$PC_q^k$ is the path cost for flowing one unit of commodity $k$ on path $q$.

### Decision Variables

$f_q^k$ proportion of commodity $k$ on path $q$, $\forall q \in Q(k), k \in K$

For each commodity $k$, we replace the flow balance constraints (1.1c) with a convex combination of their extreme points. Let $\delta^q$ be an extreme point of the polytope defined by

$$\Delta(k) = \{$$

$$\sum_{\substack{a \in A \\ orig(a)=i}} \delta_a - \sum_{\substack{a \in A \\ dest(a)=i}} \delta_a = \begin{cases} 1 & orig(k) = i \\ -1 & dest(k) = i \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in N \tag{1.2a}$$

$$0 \leq \delta_a \leq 1 \qquad \forall a \in A\}.$$

Note that this polytope has pure network structure, so the extreme points will be integral and represent paths for commodity $k$. Any set of arc flows $x_a^k$ which satisfy flow balance can be expressed as a convex combination of these path flows. Substituting $\sum_{q \in Q(k)} f_q^k \delta_a^q$ for $x_a^k$ and adding the convexity constraints $\sum_{q \in Q(k)} f_q^k = 1$, we express BDP as

$$\text{(PATH)} \qquad min \quad \sum_{k \in K} \sum_{q \in Q(k)} v^k PC_q^k f_q^k \tag{1.3a}$$

$$s.t.$$

$$\sum_{k \in K} \sum_{q \in Q(k)} v^k f_q^k \delta_a^q - u_a y_a \leq 0 \qquad \forall a \in A \tag{1.3b}$$

$$\sum_{q \in Q(k)} f_q^k = 1 \qquad \forall k \in K \tag{1.3c}$$

$$\sum_{a \in A} e_a y_a \leq B \tag{1.3d}$$

$$\sum_{\substack{a \in A \\ orig(a)=i}} e_a y_a \leq B(i) \qquad \forall i \in N \tag{1.3e}$$

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

$$y_a \in \{0,1\} \qquad \forall a \in A.$$

Constraints (1.3b) ensure that flow is only allowed on arcs included in the network. Constraints (1.3c) are commodity convexity constraints that ensure all of each commodity is delivered. (1.3d) is the overall budget constraint and (1.3e) are the budget constraints for each node.

## 1.4 Comparing NODE and PATH

Again, we let $|N|, |K|, |A|$ denote the sizes of the sets of nodes, commodities and arcs, respectively. The number of bundle constraints (1.3b), $|A|$, is unchanged from the node-arc formulation. However, the $|N||K|$ mass-balance equations (1.1c) are replaced by $|K|$ convexity constraints (1.3c). As we show in Chapter 2, the number of path variables is exponential in $|N|$. However, we can use column generation techniques to prevent explicitly including all of them. For the same size example network as before ($|N|$=100, $|K|$=1000, $|A|$=5000), there would be:

- 6,101 constraints
- 5,000 binary $y_a$ selection variables
- an exponential number of $f_q^k$ continuous path variables

PATH can accommodate some constraints which could not be easily incorporated into NODE. For instance, limiting the number of arcs used in delivering a commodity to max_arcs may be incorporated into PATH by adding the constraint $\sum_{a \in A} \delta_a^q \leq$ max_arcs to $\Delta(k)$. However, this constraint may not be written as linear inequalities in the $x_a^k$ flow variables of NODE. Constraints of this nature might arise in planning airline itineraries for passengers (commodities) unwilling to make more than two plane changes.

## 1.5 Complexity

Let UBDP denote the undirected version of BDP. For UBDP with the node budget constraints (1.1e or 1.3e) removed, Johnson, Lenstra and Rinnooy Kan [28] show that it is NP-complete even if all the arcs have equal fixed cost and the solutions are restricted to trees. Wong [42] shows that, even under the restriction of unit demands between all nodes, finding a solution within a factor of $n^{1-\epsilon}$ is NP-hard., where $n$ is the number of nodes and $\epsilon > 0$.

**Proposition 1.1** *The feasibility problem for UBDP with node budget constraints* (1.1e) *or* (1.3e) *on the fixed costs of the arcs incident to each node is NP-Complete even if all of the arcs' fixed costs are one.*

Let UBDP be defined by commodity set $K$, node set $N$, edge set $E$, edge capacities $U$, budget $B$, node budgets $B(i)$, and fixed costs for edges $e_a$. The Degree Constrained Spanning Tree (DCST) problem is defined as follows: given a graph $G = (V, E)$ and an integer $k \geq 2$, find a spanning tree for $G$ such that no node has degree greater than $k$. We can transform an arbitrary instance of DCST to UBDP on the same node set and edge set. Let $B(i) = k, B = \sum_{i \in N} B(i), e_a = 1, u_a = \infty$, and commodity set $K = \{(1, j) \mid j = 2, \ldots, n\}$ each with unit volume. If $G' = (N, E')$ is a solution to DCST, then $G'$ is also a solution to UBDP since the node budget constraints are met and the connectedness of $G'$ will ensure a delivery path for each commodity. Conversely, let $G'' = (N, E'')$ be a solution to the UBDP. $G''$ must be connected since there is a path from node 1 to each other node (to deliver a commodity). Construct $G'$ by finding a spanning tree in polynomial time by deleting edges from $G''$. The degree of each node

in $G'$ is at most the same as in $G''$, so the degree constraints are satisfied and $G'$ is a solution to DCST. Since DCST is NP-complete [20] and a solution to UBDP can be verified as feasible in polynomial time, the feasibility problem for UBDP is NP-complete.

□

**Proposition 1.2** *The feasibility problem for the directed BDP with budget constraints on the fixed costs for arcs out of each node is NP-complete.*

We show that finding a Hamiltonian cycle is a restriction of the BDP. Let an instance of BDP be defined by node set $N$, commodity set $K = \{(1,j) \mid j \in N\} \cup \{(2,1)\}$, arc set $A$, node budgets $B(i) = 1$, and fixed arc costs $e_a = 1$. Then in any solution, all nodes must be connected in a cycle and the out-degree of each node must be one. Thus any solution to BDP is a Hamiltonian cycle. Since the feasibility problem for the Hamiltonian cycle is NP-Complete, feasibility for BDP is NP-Complete. □

## 1.6 Previous work

In this section, we review papers which address BDP or which have introduced valid inequalities for the related fixed-charge problem.

Magnanti and Wong [33] and Minoux [34] survey network design problems and solution procedures. Polynomial time algorithms exist only for some special cases; e.g., if the solutions are restricted to spanning trees and the flow cost for each arc is one, Hu [26] provides an $O(n^4)$ algorithm based on maximal flow computations. Johnson, Lenstra, and Rinnooy Kan [28] and Wong [42] give the complexity results presented in §1.5.

For unit flows between $m$ pairs of $n$ nodes in an undirected network, Hoang [25] proposed an $O(mn^2)$ lower bound based on the effect of arc deletions. In 1979, Dionne and Florian [16] generalized Hoang's lower bound to include non-unit volumes. They also proposed a branch-and-bound procedure based on an approximation of their lower bound. Since this approximation may overestimate the optimal solution, the algorithm may fathom nodes erroneously. The exact Dionne-Florian bound is $O(n^4)$. Building on Dionne and Florian's work, Gallo [19] proposed a family of lower bounds of complexity $O(n^4)$ which dominate the previous bounds. In 1987, Ahuja and Murty [1] presented two lower bounds—an $O(n^4)$ bound which averaged 1 to 3 percent sharper than Gallo's bound but has larger ($O(n^2)$) space-complexity and an $O(n^3)$ bound which outperformed Hoang's bound.

Boffey and Hinxman [12] propose a construction heuristic with backtracking which manipulates three sets of links—excluded, included, and undecided. Their heuristic starts with allowable operators—rules to convert one feasible solution to another. They report results of tests on problems with up to 20 nodes. They also propose a branch-and-bound procedure and test it on networks with 10 and 20 nodes under budget constraints ranging from 10% to 90% of the combined fixed cost for all links. The most difficult problems were at the 20% budget level.

We next cover the research for the related fixed-charge network design problem. Rardin and Wolsey [37] propose a collection of *dicut* inequalities for fixed-charge network design to strengthen the formulation in the original variables to that of the multicommodity extended formulation proposed by Rardin and Choe [36]. Since there are

an exponential number of dicut inequalities, they recommend further research to find separation algorithms for them.

Hellstrand *et al.* [24] show that the polytope for the uncapacitated fixed-charge network design problem is quasi-integral and note that this property means that this problem may be solvable by a pivoting scheme such as the simplex method, if degeneracy can be avoided.

Padberg, Van Roy, and Wolsey [35] develop a number of facet inducing inequalities for the fixed-charge network flow problem. In particular, their "flow cover" inequalities apply to the budget design problem. Van Roy and Wolsey in [41] propose a valid inequality for the uncapacitated fixed-charge network problem motivated by work on the lot-sizing problem. They report successful computational experience for the more restricted lot-sizing and multi-level distribution network problems.

Balakrishnan [5] addresses the the fixed-charged version of PATH with the addition of the constraints (using our notation) $f_q^k \leq y_a$. These constraints ensure that the proportion of flow of commodity $k$ on path $q$ containing arc $a$ is less than or equal to the design variable $y_a$ for the arc. The number of these constraints is the number of $f_q^k$ variables (which is exponential in the number of nodes) times the number of arcs. He provides a characterization of fractional extreme points, two classes of cuts to eliminate some of them, and separation heuristics for these cuts.

For the undirected fixed-charge network design problem with the addition of survivability constraints (node-disjoint and/or arc-disjoint paths must exist for some traffic), Grotschel, Monma, and others have made a number of important contributions. See

[6] for references to these papers. Since our research does not include survivability constraints, we omit these papers.

# Chapter 2

## Column Generation for the Budget Design Problem

In PATH, the number of paths $\delta^q$ for each commodity is generally exponential in the number of nodes of the network unless the candidate arc set $A$ is small. Consequently, explicitly including all $f_q^k$ columns in PATH may result in a formulation which is too large to store or solve easily on a workstation. In this chapter, we develop a column generation procedure to solve the linear programming (LP) relaxation of PATH which implicitly considers the $f_q^k$ variables. Because the bound provided by the LP relaxation of PATH is weak, we suggest cuts to strengthen it and show the effect of adding them on the column generation procedure.

The LP relaxation of PATH-MP is presented below with the indicated assignment of dual variables.

$$\text{(MP)} \quad min \quad \sum_{k \in K} \sum_{q \in Q(k)} v^k \, PC_q^k \, f_q^k \qquad \text{DUALS} \qquad (2.1a)$$

s.t.

$$\sum_{k \in K} \sum_{q \in Q(k)} v^k \, f_q^k \, \delta_a^q - u_a y_a \leq 0 \qquad \forall a \in A \qquad (\beta_a) \qquad (2.1b)$$

$$\sum_{q \in Q(k)} f_q^k = 1 \qquad \forall k \in K \qquad (\chi_k) \qquad (2.1c)$$

$$\sum_{a \in A} e_a y_a \leq B \qquad (\eta) \qquad (2.1d)$$

$$\sum_{\substack{a \in A \\ orig(a)=i}} e_a y_a \leq B(i) \qquad \forall i \in N \qquad (\theta_i) \qquad (2.1e)$$

$$y_a \leq 1 \qquad \forall a \in A \qquad (\mu_a)$$

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

$$y_a \geq 0 \qquad \forall a \in A$$

The dual of MP is

$$\text{(DUAL)} \quad max \quad \sum_{k \in K} \chi_k + \eta B + \sum_{i \in N} B(i)\theta_i + \sum_{a \in A} \mu_a \qquad (2.2a)$$

s.t.

$$\sum_{a \in A} (v^k \, \delta_a^q)\beta_a + \chi_k - v^k \, PC_q^k \leq 0 \quad \forall k \in K, q \in Q(k) \qquad (2.2b)$$

$$-u_a \beta_a + e_a \eta + e_a \theta_{orig(a)} + \mu_a \leq 0 \quad \forall a \in A \qquad (2.2c)$$

$$\beta_a, \eta, \theta_i, \mu_a \leq 0$$

$$\chi_k \text{ free.}$$

Constraints (2.2c) correspond to the arc selection variables $y_a$ and constraints (2.2b) correspond to the commodity path variables $f_q^k$ . If some of the $f_q^k$ variables are missing from MP, then they will correspond to missing constraints (2.2b) in the dual problem.

Thus, if there is a column $f_q^k$ of MP not included in RMP with favorable reduced cost, it will correspond to a violated constraint in the dual of form (2.2b)

## 2.1 Pricing subproblem for commodity $k$

To find a violated constraint (2.2b), the objective of the column generation subproblem is

$$max \sum_{a \in A} v^k \, \delta_a^q \, \beta_a + \chi_k - v^k \, PC_q^k$$

or equivalently

$$min \sum_{a \in A} v^k \, \delta_a^q \, (-\beta_a) - \chi_k + PC_q^k \, v^k \, .$$

For each commodity, the constraints of the pricing subproblem (from the Dantzig-Wolfe decomposition) are defined by $\Delta(k)$. Using the minimization form for the objective, we obtain the following column generation subproblem.

$$(SP_k) \quad min \quad \sum_{a \in A} v^k \, \delta_a^q \, (-\beta_a) - \chi_k + PC_q^k \, v^k \tag{2.3a}$$

s.t.

$$\text{Balance} \quad \sum_{\substack{a \in A \\ orig(a)=i}} \delta_a^q - \sum_{\substack{a \in A \\ dest(a)=i}} \delta_a^q = \begin{cases} 1 & orig(k) = i \\ -1 & dest(k) = i \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in N \tag{2.3b}$$

$$0 \leq \delta_a^q \leq 1 \qquad \forall a \in A$$

$\beta_a \leq 0$ and $\chi_k$ free are fixed by the present solution to RMP. If the objective value of $SP_k$ is less than zero, then the solution identifies a path with negative reduced cost for commodity $k$ and we add this column to RMP.

## 2.2 Solution of $SP_k$ by shortest paths

We can solve each $SP_k$ as a shortest path problem where each arc's cost is a function of the original flow cost $c_a$ and dual variables $\beta_a$ for the arcs. The flow cost function for each arc is $(-\beta_a + c_a)$ and the length for any path is modified by the dual $(-\chi_k)$ on the convexity constraint of the commodity. For each arc, the cost function is non-negative since $\beta_a \leq 0$ and the original arc cost $c_a$ was assumed to be non-negative. Consequently, any shortest path algorithm may be used to solve these subproblems.

By complementary slackness, $\beta_a$ will be nonzero only when (2.1b) is tight, meaning that the flow on the arc is equal to $u_a y_a$. We can think of the $-\beta_a$ as a toll assessed on each unit of flow to discourage using arcs which are saturated for the current value of $y_a$. Since (2.1c) is an equality constraint, its dual $\chi_k$ may always be nonzero. We can think of this dual as a discount factor for the commodity that it represents. When it is sufficiently large, new paths for the commodity may allow the objective function of MP to be reduced.

## 2.3 Column generation algorithm

Instead of explicitly including all of the variables $f_q^k$ , we will initialize the formulation with a subset of the possible paths and use column generation to implicitly consider all paths and identify new paths which "price-out" (have favorable reduced cost). Once

we have a feasible solution over the columns currently included, we solve the pricing subproblems $SP_k$ to either generate new columns to add or prove that no such columns exits. Let RMP denote the restriction of MP where only a subset of the possible paths are included. Once we have an optimal solution for RMP and each of the $SP_k$ fail to generate a new column, the optimal solution for RMP is also an optimal solution to MP. This procedure is depicted graphically in Figure 2.1.

## 2.4 Strengthening MP with additional cuts

**MP is a weak LP relaxation.** Both the node-arc and path-based formulations which we introduced above have weak LP relaxations, especially if the arc capacities $u_a$ are large with respect to the possible flows. Using the notation for the node-arc formulation, the $y_a$ values are bounded below by

$$y_a \geq \frac{\sum_{k \in K} v^k x_a^k}{u_a} \tag{2.4}$$

where $x_a^k$ is the proportion of commodity $k$ on arc $a$. For large $u_a$ capacities these constraints provide little support to force the $y_a$ to be near one for arcs which are used. Fractional $y_a$ will allow arcs to have flow but only part of their fixed cost will be incurred in the budget constraint, so more arcs may have flow in the LP relaxation than in the MIP. Consequently, the bound provided by MP may significantly underestimate PATH and thus provide little help in fathoming nodes in any branch-and-bound procedure.

To strengthen the bound of MP, we propose adding valid inequalities which we call "forcing constraints." These constraints are a disaggregation of the bundle constraints and prevent the flow of an individual commodity on an arc unless the proportion of flow

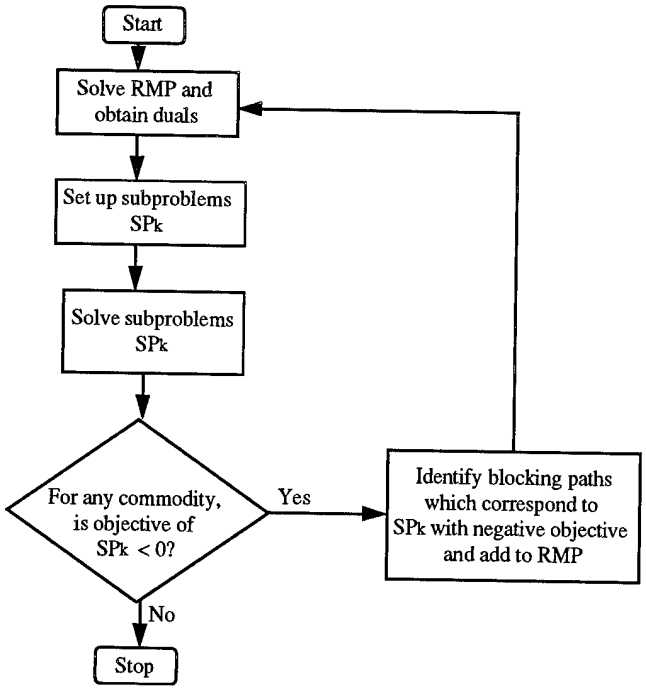


Figure 2.1: Flowchart for column generation

on the arc is less than or equal to the $y_a$ selection variable for the arc. The technique of using cuts which are a disaggregation of known constraints is a common practice. One reference which describes this procedure is Barnhart *et al.* [7]. The difficulty is usually that the problem formulation becomes much larger.

These forcing constraints for NODE (1.1) are

$$x_a^k - y_a \leq 0 \qquad \forall a \in A, k \in K \tag{1.1f}$$

and for PATH (1.3) are

$$\sum_{q \in Q(k)} f_q^k \, \delta_a^q - y_a \leq 0 \qquad \forall a \in A, k \in K. \tag{1.3f}$$

Adding all possible valid inequalities (1.3f) to one test instance (with six nodes and thirty commodities) reduced the number of nodes evaluated in the branch-and-bound tree from 1161 to 33. So, these cuts seem to be strong and may be high dimensional faces of the convex hull of feasible solutions.

Since there will be one forcing constraint for each arc-commodity combination, there are $|A||K|$ inequalities of this form. Clearly, for large networks with many commodities, adding all of these constraints may result in a formulation which is too large to store. In cases where including all of the forcing constraints is not practical, a simple separation algorithm can be used to identify violated cuts. We present a separation algorithm below. However, we should note first that since these constraints involve the $f_q^k$ variables which are in $SP_k$ , the pricing subproblem will have to be modified to include the dual variables for the added cuts (even if the forcing constraints are only added at the root node of the search tree).

**Separation Algorithm** Starting with a fractional $y_a$, examine the non-zeros in the coefficient matrix for the (1.3b) bundle constraint associated with arc $a$. Of these non-zeros $f_q^k$, any individual $f_q^k > y_a$ or collection of $f_q^k$ for the same commodity where $\sum_{q \in Q(k)} f_q^k > y_a$ identify a violated forcing constraint. This test for violated forcing constraints is polynomial, so finding them is straight-forward; however, since these constraints do not define the convex hull of extreme points, branching or adding other cuts will probably still be necessary.

**Modification of $SP_k$ for the Forcing Constraints** If we add the forcing constraints (1.3f) to MP and let $\phi_a^k \leq 0$ denote the corresponding dual variable for commodity $k$ and arc $a$, then constraint (2.2b) of DUAL will become

$$\sum_{a \in A} \{\delta_a^q (v^k \beta_a + \phi_a^k)\} + \chi_k - v^k PC_q^k \leq 0 \quad \forall k \in K, q \in Q(k) \qquad (2.2b')$$

Correspondingly, the objective of $SP_k$ will become

$$min \quad \sum_{a \in A} \{\delta_a^q (-v^k \beta_a - \phi_a^k)\} - \chi_k + PC_q^k v^k .$$

Since the new term introduced to the objective is also nonnegative, $SP_k$ can still be solved by any shortest path algorithm. The flow cost of each arc will now be

$$-\beta_a - \phi_a^k + c_a. \qquad (2.5)$$

Before adding the forcing constraints (1.3f), the arc cost functions were not dependent on the commodity, so all $SP_k$ for commodities which shared a common origin could

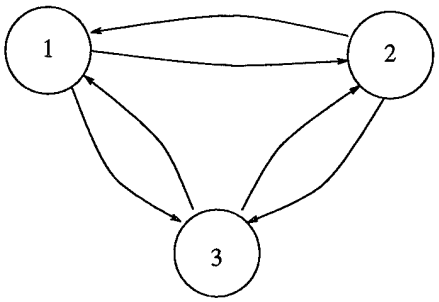


Figure 2.2: Node and arc set for example

be solved simultaneously by finding the shortest path tree from this origin to all other nodes. However, now each $SP_k$ must be solved separately.

## 2.5 Branch-and-price algorithm

We propose implementing our column generation procedure to solve the LP-relaxation of PATH in a branch-and-bound search for an integer solution to PATH. Branching on the (fractional) $y_a$ can be accommodated in $SP_k$ by deleting arcs $a$ for which $y_a$ is fixed at zero. A labeling shortest path algorithm used to solve $SP_k$ would simply ignore these arcs when computing distance labels. The bundle constraints (2.1b) of MP would prevent solutions which include $f_q^k$ variables corresponding to paths which use a deleted arc. Consequently, MP would not require modification during the search. Thus, we may add new columns which "price-out" throughout the branch-and-bound search. Branch-and-bound algorithms where column generation is allowed within the search are described in Barnhart *et al.* [9] where the term "branch-and-price" is introduced to describe them.

## 2.6 Example

In this example, we demonstrate the solution of MP strengthened by the disaggregated forcing constraints (1.3f) via the column generation strategy described above. The LP relaxation gives an integer solution for this example, but, in general, branching will be required to reach an integer solution.

The nodes are $N = \{1, 2, 3\}$. All node-pairs are potential arcs;
i.e., $A = \{(1,2), (1,3), (2,1), (2,3), (3,1), (3,2)\}$. The flow cost and fixed cost of each arc is one, $c_a = 1, e_a = 1$ and the arc capacities are $u_{12} = 2, u_{13} = 3, u_{21} = 1, u_{23} = 2, u_{31} = 0, u_{32} = 1$. The commodities (each with unit volume) are $\{(1,2), (1,3), (2,3)\}$ and node budget capacities $B(i)$ are all one. The overall budget $B = 3$. The node and arc set are depicted in Figure 2.2. Since all of the flow costs are one, the objective is equivalent to minimizing the sum of the number of arcs used to deliver each commodity. Since $u_{31} = 0$, $y_{31}$ may be deleted from the formulation.

We initially populate RMP with paths for each commodity that use the single arc from the commodity's origin to its destination. In order to use column generation, a feasible solution must exist among the columns in RBP, so we introduce the artificial variables $a_t$ with high objective cost to the budget constraints in order to guarantee that a solution for RMP always exists. Let $f^{ij}_{i-\ldots-j}$ represent the path $(i, \ldots, j)$ for commodity with origin $i$ and destination $j$.

For this example, the initial RMP is

$$
\begin{array}{lllllllllll}
min & f^{12}_{1-2} + f^{13}_{1-3} + f^{23}_{2-3} & & & & & & & & +\mathrm{M}\sum_{i=0}^{3} a_i & \\
\text{s.t.} & & & & & & & & & & \\
\text{bundle}_{12} & f^{12}_{1-2} & & & -2y_{12} & & & & & & \leq 0 \\
\text{bundle}_{13} & & f^{13}_{1-3} & & & -3y_{13} & & & & & \leq 0 \\
\text{bundle}_{21} & & & & & & -y_{21} & & & & \leq 0 \\
\text{bundle}_{23} & & & f^{23}_{2-3} & & & & -2y_{23} & & & \leq 0 \\
\text{bundle}_{32} & & & & & & & & -y_{32} & & \leq 0 \\
\text{convex}^{12} & f^{12}_{1-2} & & & & & & & & & = 1 \\
\text{convex}^{13} & & f^{13}_{1-3} & & & & & & & & = 1 \\
\text{convex}^{23} & & & f^{23}_{2-3} & & & & & & & = 1 \\
\text{budget}_1 & & & & y_{12} & +y_{13} & & & & -a_1 & \leq 1 \\
\text{budget}_2 & & & & & & y_{21} & +y_{23} & & -a_2 & \leq 1 \\
\text{budget}_3 & & & & & & & & y_{32} & -a_3 & \leq 1 \\
\text{budget} & & & & y_{12} & +y_{13} & +y_{21} & +y_{23} & +y_{32} & -a_0 & \leq 3 \\
\text{forcing}^{12}_{12} & f^{12}_{1-2} & & & -y_{12} & & & & & & \leq 0 \\
\text{forcing}^{13}_{13} & & f^{13}_{1-3} & & & -y_{13} & & & & & \leq 0 \\
\text{forcing}^{23}_{23} & & & f^{23}_{2-3} & & & & -y_{23} & & & \leq 0 \\
 & & & & & & & & & 0 \leq y_a & \leq 1 \\
 & & & & & & & & & f^k_q & \geq 0.
\end{array}
$$

Let M = 100000000. Solving RMP gives the following solution with an objective value of 500,003.

$$y_{12} = y_{13} = y_{23} = 1.0$$

$$f^{12}_{1-2} = f^{13}_{1-3} = f^{23}_{2-3} = 1$$

$$a_1 = 0.005$$

With these nonzero duals

$$\chi_{12} = \chi_{13} = 500001$$

$$\chi_{23} = 1$$

$$\phi_{12}^{12} = -500000$$

Solving $SP_k$ for commodity (1,2) identifies the shortest path (1-3-2). The objective value for $SP_k$ is

$$(-\beta_{13}) + c_{13}v^{12} - \phi_{13}^{12} + (-\beta_{32}) + c_{32}v^{12} - \phi_{32}^{12} - \chi_{12}$$
$$= 1 + 1 - 500001 = -499999$$

Since the objective is less than zero, the path (1-3-2) prices out for commodity (1,2) and will be added to RMP. Solving $SP_k$ for commodity (1,3) identifies the shortest path (1-2-3) for which the objective value of $SP_k$ is

$$(-\beta_{12}) + c_{12}v^{13} - \phi_{12}^{13} + (-\beta_{23}) + c_{23}v^{13} - \phi_{23}^{13} - \chi_{13}$$
$$= 1 + 1 - 500001 = -499999$$

and the path (1-2-3) prices out for commodity (1,3) and will be added to RMP. The objective for $SP_k$ for commodity (2,3) is zero, so no columns are added for this commodity.

Adding the two paths which priced out and the corresponding forcing constraints to RMP gives the following formulation.

$$
\begin{array}{lllllllllllll}
\textit{min} & f^{12}_{1-2} + f^{13}_{1-3} + f^{23}_{2-3} + f^{12}_{1-3-2} + f^{13}_{1-2-3} & & & & & & & & & +\mathrm{M}\sum_{i=0}^{3} a_i & & \\
\text{s.t.} & & & & & & & & & & & & \\
\text{bundle}_{12} & f^{12}_{1-2} & & & + f^{13}_{1-2-3} & -2y_{12} & & & & & & & \leq 0 \\
\text{bundle}_{13} & & f^{13}_{1-3} & & + f^{12}_{1-3-2} & & -3y_{13} & & & & & & \leq 0 \\
\text{bundle}_{21} & & & & & & & -y_{21} & & & & & \leq 0 \\
\text{bundle}_{23} & & & f^{23}_{2-3} & & + f^{13}_{1-2-3} & & & -2y_{23} & & & & \leq 0 \\
\text{bundle}_{32} & & & & & & & & & -y_{32} & & & \leq 0 \\
\text{convex}^{12} & f^{12}_{1-2} & & & + f^{12}_{1-3-2} & & & & & & & & = 1 \\
\text{convex}^{13} & & f^{13}_{1-3} & & & + f^{13}_{1-2-3} & & & & & & & = 1 \\
\text{convex}^{23} & & & f^{23}_{2-3} & & & & & & & & & = 1 \\
\text{budget}_1 & & & & & & y_{12} & + y_{13} & & & & -a_1 & \leq 1 \\
\text{budget}_2 & & & & & & & & y_{21} & + y_{23} & & -a_2 & \leq 1 \\
\text{budget}_3 & & & & & & & & & & y_{32} & -a_3 & \leq 1 \\
\text{budget} & & & & & & y_{12} & + y_{13} & + y_{21} & + y_{23} & + y_{32} & -a_0 & \leq 3 \\
\text{forcing}^{12}_{12} & f^{12}_{1-2} & & & & & -y_{12} & & & & & & \leq 0 \\
\text{forcing}^{12}_{32} & & & & f^{12}_{1-3-2} & & & & & & -y_{32} & & \leq 0 \\
\text{forcing}^{13}_{12} & & & & & f^{13}_{1-2-3} & & -y_{13} & & & & & \leq 0 \\
\text{forcing}^{13}_{13} & & f^{13}_{1-3} & & & & & -y_{13} & & & & & \leq 0 \\
\text{forcing}^{23}_{23} & & & f^{23}_{2-3} & & & & & & -y_{23} & & & \leq 0 \\
 & & & & & & & & & & & 0 \leq y_a & \leq 1 \\
 & & & & & & & & & & & f^k_q & \geq 0.
\end{array}
$$

Solving RMP with these new columns gives an objective value of 4. Solving each $SP_k$ with the new dual values does not identified any blocking paths with favorable reduced cost. So the solution to RMP is also optimal for MP. In general, branching will be necessary before identifying an integral LP solution; however, in this particular case, all of the $y_a$ are integer. Since the solution to MP is integral, it is also an optimal for solution for PATH. The optimal solution is:

$$y_{12} = y_{23} = 1.0$$

$$f^{12}_{1-2} = f^{23}_{2-3} = f^{13}_{1-2-3} = 1.0$$

# Chapter 3

# The Railroad Blocking Problem (RBP)

## 3.1 Introduction

As we mentioned in the introduction, our motivation for studying BDP is an application in the railroad industry. In the railroad blocking problem (RBP), the physical rail network (the railroad terminals and tracks) is already defined. The "blocking network" to be constructed is a virtual network that is overlayed on the physical network. The blocks are virtual "express" arcs which a commodity may take to have uninterrupted service between two terminals that are not necessarily connected by a physical link. In this chapter, we describe RBP and rail operations in general. In the following chapters, we introduce a solution procedure for RBP and provide computational results.

On major domestic railroads, a typical general merchandise shipment may pass through many classification yards on its route from origin to destination (OD). At these yards, the incoming traffic is *reclassified* (sorted and grouped together) to be placed on outgoing trains. On average, each reclassification results in a one day delay for the shipment. In addition, the classification process is labor and capital intensive since many workers and large quantities of equipment are needed to sort the traffic and construction and maintenance of large yards is necessary to handle the sorting task. To prevent shipments from being reclassified at every yard they pass through, several shipments may be grouped together to form a *block*. A block has associated with it an OD pair which may or may not be the OD pair of any of the individual cars contained in the

block. Once a shipment is placed in a block, it is not reclassified until it reaches the destination of that block.

Over the years, railroads have developed blocking plans that dictate which blocks should be built at each individual yard and which traffic should be assigned to each block. For the most part, these plans have evolved through incremental refinement of an existing plan. These refinements are inherently local in nature since they only consider the effects of changing a small number of blocks in the existing plan. Thus, these local improvement approaches may fail to recognize opportunities for improvement that require more significant changes to the current blocking plan. Also, these local approaches may not be sophisticated enough to capture the effects of changes to the plan on the flow of traffic over the entire rail network.

The objective of the RBP is to develop a feasible blocking plan which in conjunction with other operating policies (such as train routes and frequencies of service) minimize the total cost of delivering the commodities. These costs are usually broken down into "car-handling" costs associated with handling (or blocking) a car and "car-miles" costs associated with the movement (conveyance) of a car.

To solve RBP, we must decide which blocks to include in the blocking plan and which blocks to use to deliver each commodity. We provide an example RBP in this section, describe RBP and its context in railroad operations in §3.2 and formalize our terminology in §3.3. The solutions to RBP proposed in the literature are reviewed in §3.4; however, none of the optimization-based heuristic or exact solutions proposed for the blocking plan have gained widespread use among domestic railroads [22]. We discuss methods for dealing with the large amount of data from a real-world application in §3.5.

**An example** of a rail system with 4 terminals is shown in Figure 3.1. We define commodities as origin-destination pairs of terminals. While there would be "four-choose-two" possible commodities for this example, we assume that the only commodities are: A→B with 100 cars, A→C with 80 cars, and A→D with 90 cars. Terminal A has volume capacity of 270 cars which is its minimum since 270 cars originate there. Terminal B and C can each block 90 cars. Terminal A can build 2 blocks whereas each of the other terminals can only build one block.

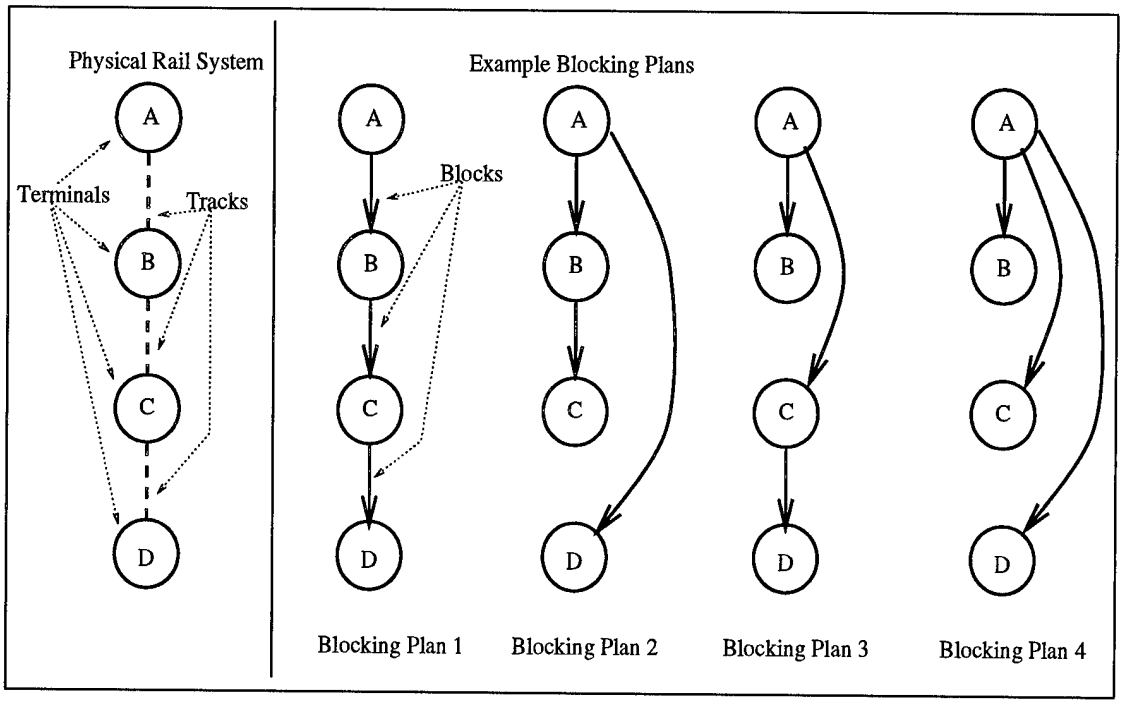


Figure 3.1: Physical rail system and four blocking plans

At each terminal, limited classification resources restrict the number of cars (or *car volume*) which can be classified and the number of blocks. In terms of our graph the car volume constraints would be upper bounds on the flow through each node and

the number of blocks constraints would be limits on the out-degree of each node. One performance metric of a blocking plan is the total number of classifications necessary to deliver all commodities. We consider four different blocking plans for this example.

With blocking plan 1, at terminal A all three commodities are sorted to move to terminal B. The resulting string (block) of cars may have the commodities intermingled. At terminal B, commodity A→B has reached its destination and leaves the system. Commodities A→C and A→D are blocked to be move to terminal C. At terminal C, commodity A→C has reached its destination and leaves the system, and commodity A→D is blocked to move to terminal D. Using this blocking plan, the 100 cars for commodity A→B each use one block, (A,B), the 80 cars for commodity A→C each use two blocks, (A,B) and (B,C), and the 90 cars for commodity A→D each use three blocks, (A,B),(B,C),(C,D). So, for the blocking plan 1 there are 530 classifications; however, this plan is not feasible because it requires blocking 170 cars at terminal B which has a maximum volume of 90 cars. The strategy of building blocks only between adjacent terminals, as in blocking plan 1, is referred to as "short blocking."

Blocking plan 2 provides a "direct" block from terminal A to terminal D. Cars which travel in this block will still move along the physical track through terminals B and C, but since they are pre-sorted for terminal D, they do not require classification resources at B or C. To deliver the three commodities using this blocking plan requires 350 classifications, since the 90 cars of Commodity A→D now require only one block, (A,D). This blocking plan is feasible since it observes the limits on number and total volume of the blocks built at each terminal.

Similar calculations show that blocking plans 3 and 4 require 360 and 270 classifications, respectively. However, blocking plan 4 is infeasible since it requires three blocks for terminal A which has a maximum of two blocks. Always blocking commodities directly to their destination, as in blocking plan 4, is referred to as "long blocking."

So blocking plan 2 with 350 handlings is optimal with respect to the total number of classifications required. However, if the number of cars which could be blocked at B is reduced to less than 80 (or equivalently, the volume of commodity A→C were more than 80), then blocking plan 2 would be infeasible and Blocking Plan 3 with 360 handlings would be optimal.

## 3.2 A description of the blocking problem

To further provide the reader with a context of the overall operation of a railroad, we include the following excerpt on rail operations from [15].

First, an order for a number of empty cars is issued by a client. At the concerned yard, the cars are selected, inspected and then delivered to the loading point. Once loaded, the cars are moved to the *origin* yard where they are sorted, or *classified*, and grouped into a *block*. A block is a group of cars, with possibly different final destinations, arbitrarily considered as a single unit for handling purposes from the yard where it is made up to its destination yard where its component cars are to be re-sorted. Rail companies use blocks as a means to take advantage of some of the economies of scale related to full train loads and to the manipulation of longer car strings in yards.

The block is eventually put on a *train* and this signals the beginning of the cars' journey. During the long haul part of this journey, the train may overtake and be overtaken by other trains with different speeds and priorities. When the train travels on single-track lines, it may also meet trains traveling in the opposite direction. Then, the train with the lowest priority has to give way and wait on a side line, or *siding*, that the train with the higher priority passes by.

At the yards where the train stops, cars and engines are regularly inspected. Also at yards, the blocks of cars may be *transfered*: taken off one train and put on another. When the block finally arrives at its destination, it is taken off the train, its cars are sorted and those arrived at their final destination are directed to the unloading station. Once empty, the cars are prepared (cleaned, inspected, etc.) for a new assignment: either a loaded trip or an empty movement.

. . .

When a train arrives for classification at a yard, it is first directed to the receiving area, where engines are disconnected, inspected and taken away, blocks are separated and cars are inspected. The cars are then directed to the sorting area, each car being assigned to a classification track according to its outbound (or classification, or blocking) destination, together with other cars sharing the same classification destination. Physically, this operation may be performed in two ways: (i) by a switching engine, in a *flat yard*, (ii) by rolling cars down an incline (or a hump) and automatically switching them to the rail track allocated to the block, in a *hump yard*. Most modern, large classification yards are hump yards. Once sorted, cars on the same classification tracks form blocks that next have to wait for the designated outbound train. Following one last inspection of the whole train, the journey may begin. It is noteworthy that not all trains arriving at a yard are thus totally classified. Trains may stop at a yard for inspections, to change crews, to refuel, to leave or to pick up blocks of cars. Thus, the total workload of a yard is larger than that defined only by its main operations: the classification of cars and the forming of trains.

**The Physical Rail System** The physical rail system consists of undirected links (one or more railroad tracks) connecting nodes (rail yards). For example, the rail network for CSX Transportation which covers roughly the eastern third of the United States is shown in Figure 3.2. A subset of the nodes represent classification yards at which blocking operations are predominantly performed. We refer to these nodes as terminals. The terminals for this rail network are shown in Figure 3.3.

Terminals can be divided into two groups based on the switching technology used:

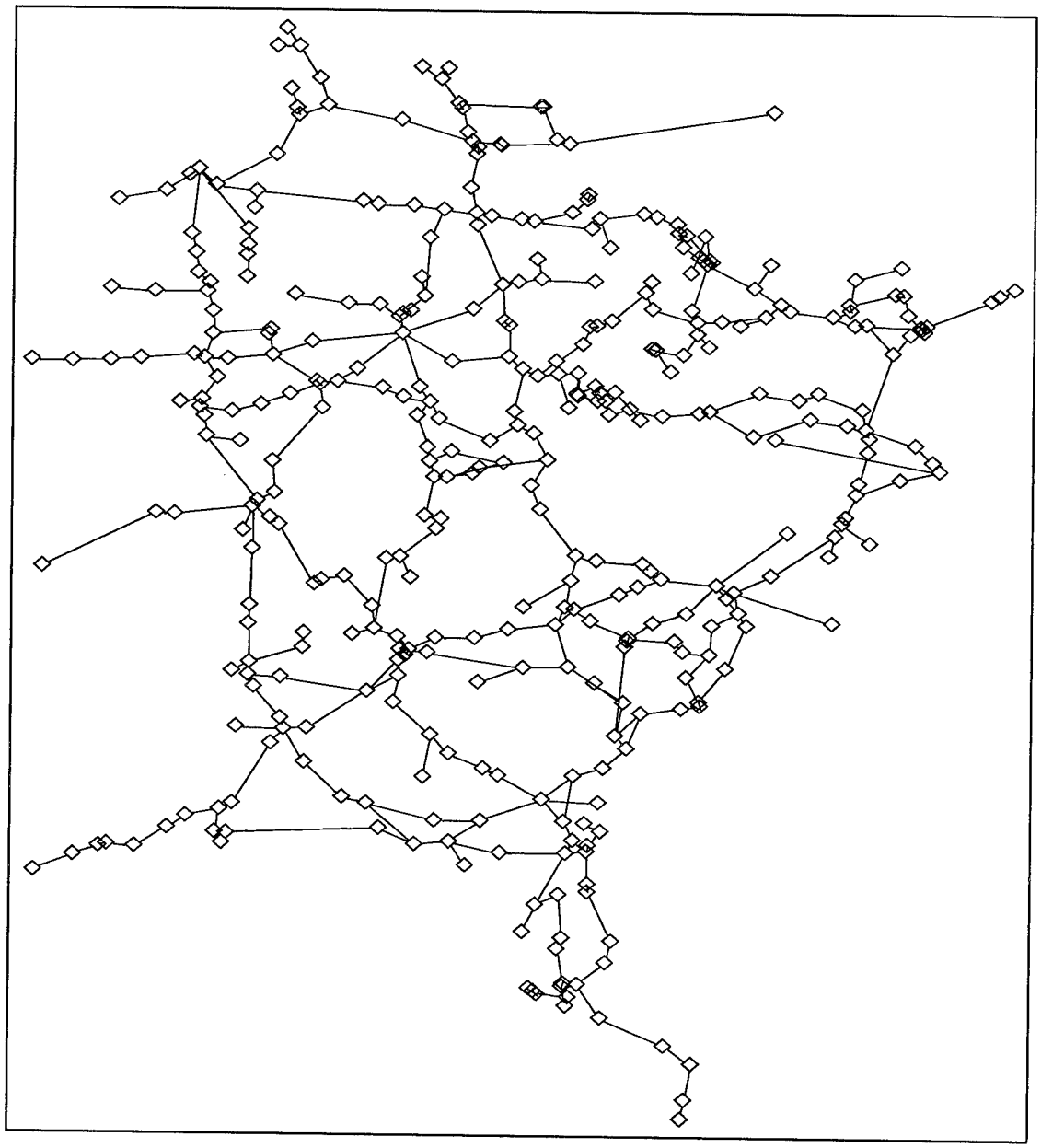

Figure 3.2: Physical rail network of links and nodes

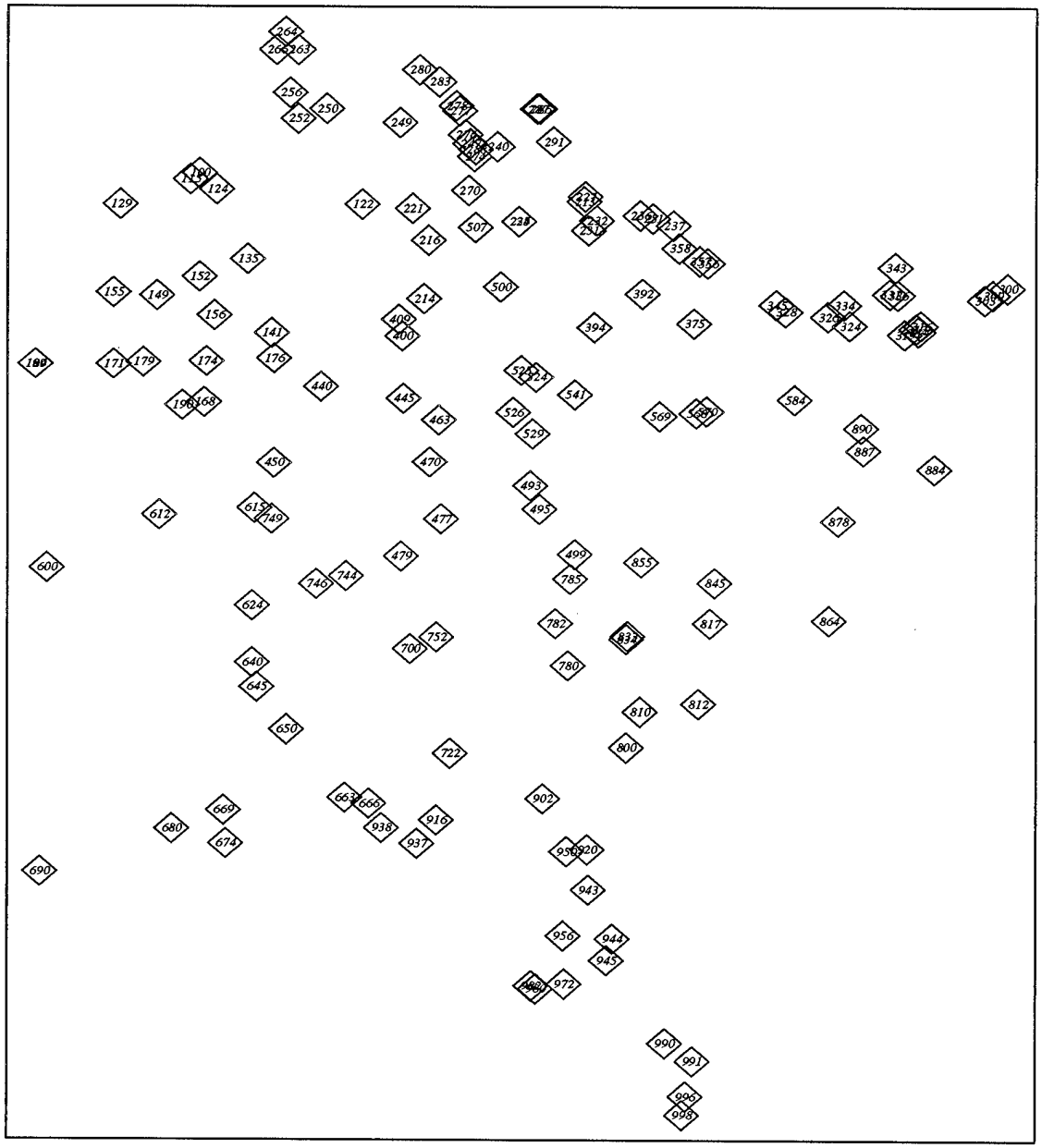


Figure 3.3: Locations of terminals in the rail network

- *Hump yards* at which cars are pushed up a hump and as they fall down the other side are automatically "switched" to the proper classification track; e.g., CSXT's terminal at Cincinnati, OH. See Figure 3.4 for a generic hump yard layout as depicted in Armacost (1995).

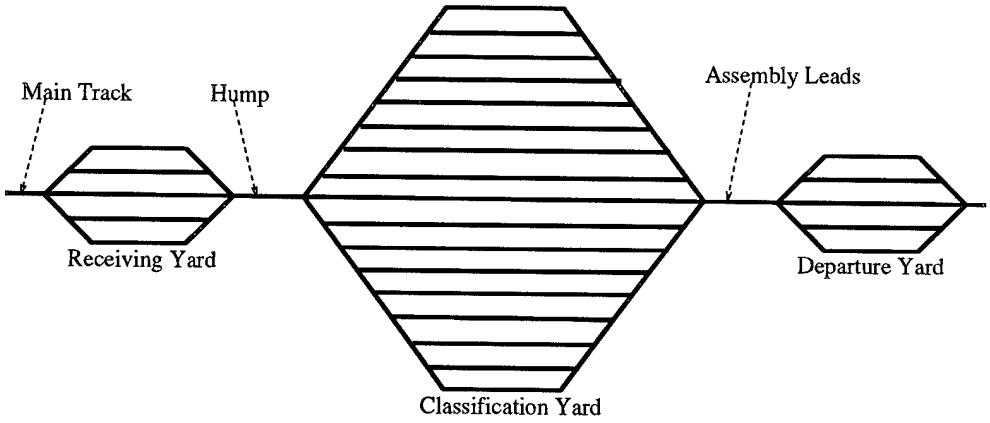


Figure 3.4: Generic hump yard layout

- *Flat yards* at which classification is done by moving the cars between classification tracks using locomotives, called *switch engines*; e.g., CSXT's terminal at Chicago, IL.

For the rail system pictured in Figure 3.2 there are:

- 590 undirected *links* (many are multi-lane) connecting the nodes
- 336 nodes rail yards (including terminals)
- 150 terminals.

Figure 3.5: Assad's taxonomy of rail decisions

**Strategic Decisions** with long time horizons and major capital investments.

- Network design and improvement. Track abandonment.
- Location of terminals and major classification facilities within large terminals.
- Highly aggregate routing decisions. Long-term planning of train services.

**Tactical decisions** with medium-term planning horizons and focus on effective allocation of existing resources rather than major acquisitions.

- Train selection and traffic routing: What trains should run and what should the required frequency of each train be to accommodate traffic demand?
- Train makeup: What groups (or blocks) of traffic should a train be allowed to carry (its take-list at a given terminal of its itinerary)?
- Terminal classification policy: Into what groups or blocks should the incoming traffic to the terminal be consolidated?
- Allocation of classification work among terminals: What is the total amount of classification work performed in the system? How should this workload be distributed among the various terminals to account for the fact that they might have different technological capabilities?

**Operational decisions** with day-to-day activities in a fairly detailed and dynamic environment.

- Train timetables: Determining arrival/departure times of each train at any intermediate station of its itinerary.
- Track scheduling and priority policy.
- Engine Scheduling
- Empty car distribution
- Yard receiving and dispatching policies.
- Line-haul and yard maintenance operations.

**How the Blocking Plan is Used** For each terminal, the blocking plan provides a menu of blocks that can be built. Which blocks are actually built on a given day will depend on what traffic arrives, but most blocks will be built each day. Using Assad's (1980) classification system (included in Figure 3.5), creating a blocking plan occurs at the tactical level of decision-making. Our research focuses on both developing the blocking plan and assigning blocking paths to each commodity and thus covers Assad's last two tactical decisions.

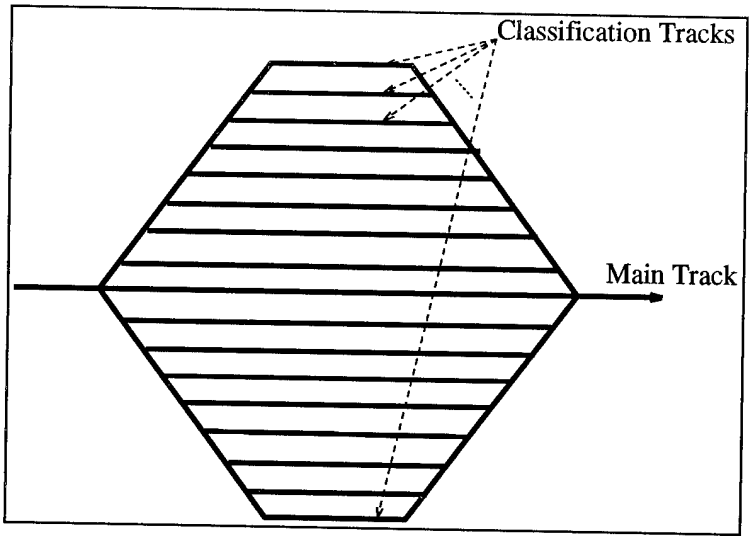


Figure 3.6: Classification yard

## A Discussion of the Constraints

**The Volume and Number of Blocks Constraints** Classification usually is done on a group of classification tracks that together roughly resemble a hexagon (or half-hexagon) with different length tracks (see Figure 3.6). The actual layouts are somewhat

different to accommodate constraints imposed by the switching technology. Researchers on the blocking plan have traditionally assumed that an acceptable approximation of the classification capacity is a rectangle with the same area as the hexagon. The capacity of the resulting rectangle can be then approximated by its width (the number of tracks) and area (the number of cars that can fit in the rectangle). Hence, the constraints on number of blocks and total volume of the blocks.

Larger volume blocks would be assembled on the longer tracks or several tracks. It is possible by scheduling train arrivals or by manipulating the cars on a classification track to build more than one block per classification track. Assigning blocks to the classification tracks or "track scheduling and priority policy," using Assad's taxonomy, is an operational decision. At the tactical decision level, this leads to a constraint on the maximum number of blocks for the terminal. Sometimes, factors of around 1.25 are used to convert the number of classification tracks into the maximum number of blocks.

**Priority Constraints** Express traffic (e.g. automotive parts or intermodal) must be delivered in a few days (usually two or three days) and non-express traffic (e.g. general merchandise) must be delivered in one to two weeks, as specified by the commitment to the customer. Since each reclassification can incur a delay of around a day, the number of reclassifications for both types of traffic must be limited. Consequently, we include constraints on the maximum number of times each commodity is reclassified. By including these constraints, we can simultaneously solve the blocking problem for both the express and the non-express traffic. Current practice in the industry is to first decide on the blocking plan for the express traffic and then to add blocks as necessary to accommodate the non-express traffic, using the left-over resources.

## 3.3 Terminology

We formalize the terminology for the RBP in this section.

**A Block** is completely specified by its origin and destination. Cars using the block will be sorted at the block's origin and then not sorted again until reaching the block's destination. An *inter-terminal* block has a terminal origin and a terminal destination whereas *local* blocks originate or terminate at a node which is not a terminal.

**Classification, Blocking, or Handling** is the process of sorting cars into different blocks.

**A Terminal** is a node of the rail network at which classification may be performed. An *End Terminal* is a terminal which may not be used for intermediate classifications for any commodity.

**A Car-move** is a one-way repositioning of a car to either pickup or deliver a commodity.

**An OD pair** is a group of cars with the same origin terminal and destination terminal. We aggregate each car-move to an OD pair as discussed in Section 3.5.

**A Priority Class** is the number of intermediate reclassifications that are permitted. If no intermediate reclassifications are permitted, then the commodity must be blocked for the destination terminal at the origin terminal.

**A Commodity** is a subgroup of an OD pair which has the same priority class.

**A Routing** for a commodity is a sequence of terminals, starting with the commodity's origin and ending with the commodity's destination, that a commodity may visit

as it moves through the network. Classification will occur at some of the terminals along one of the commodity's routings. We will assume that routings are non-circuitous; i.e., no terminal may occur more than once in the routing.

**A Blocking Assignment or Blocking Path** for a commodity is a sequence of blocks on which a commodity will travel as it moves along its routing. Classification for the commodity will be performed at the starting terminal for each of these blocks. The endpoints of the blocks in the blocking assignment will be a subsequence of the terminals in one of the commodity's routings.

**The Blocking Plan** is a listing of all blocks. The *Inter-Terminal Blocking Plan* is the listing of all blocks which have a terminal origin and a terminal destination. The *Local Blocking Plan* lists blocks for which at least one endpoint is not a terminal.

**A Train** consists of one or more blocks, a consist of locomotives, and an end-of-train device (formerly, the caboose).

### 3.4 Literature review

In the literature, the railroad blocking problem has been considered both independently and in conjunction with other problems. Consequently, this chapter will be divided into these sections: *Blocking Models*, *Routing Models*, and *Composite Models.* Also included is a section on *Related Literature.* Because our research does not address the routing problem, the *Routing Models* section is not comprehensive.

**Blocking Models** Bodin *et al.* [11] propose a large MIP formulation with constraints on the number and length of blocks at each yard and the volume of cars that each

yard may classify with the nonlinear objective of minimizing the sum of the delays incurred delivering each commodity. Their formulation has an extremely large number of binary variables. For each candidate block, a binary variable is used for each candidate routing of the block. For each possible path flow of a commodity, a binary variable is used for each candidate routing and each possible blocking path. Their model requires potential blocking paths for all commodities as input. With the 33 classification yards (terminals) that Norfolk and Western Railroad had at the time and with a restricted number of blocking paths for each commodity, this model had 6500 constraints and 11,000 variables (with 6000 binary). For this problem instance, they found a feasible solution (with extensive manual intervention) within 3% of a known lower bound.

Van Dyke [40] presents the Automatic Blocking Model (ABM) micro-computer based-system that became the heart of the early blocking software marketed by ALK Associates, Inc. The ABM system includes a heuristic based on shortest paths to assign traffic to a pre-defined set of blocks and a greedy heuristic to add blocks to an existing blocking plan.

**Routing and Makeup Models** These models address the selection and frequency of routes to be serviced by trains (train routing) and which blocks to assign to each train (train makeup).

Assad [3], in addition to giving the taxonomy in Figure 3.5, proposes an MIP formulation for routing and makeup based on a network model. His formulation decides which pairs of terminals should have direct train service between them. He ignores the block-transfer operations of set-offs (where block(s) are dropped off at intermediate terminals) and pick-ups (where block(s) are attached to the train at intermediate terminals). He

does not constrain the classification work done at each yard based on its resources (although he does allow for a different cost function at each yard) and does not constrain the length of the resulting trains. No solution procedure is given.

Assad [4] presents a greedy algorithm (and conditions under which it's optimal) for developing itineraries for multiple trains where the objective is minimizing the total hours for all car-moves. Also presented is a branch-and-bound algorithm to find exact solutions for small problems to compare to the heuristic solutions.

**Composite Models** Crainic *et al.* [14] propose a large non-linear MIP formulation to determine the blocking plan, train routing plan, and makeup plan simultaneously. Their model starts with a list of services (potential trains) and the set of possible itineraries of blocks and services for each commodity. They seek to minimize a non-linear objective of the sum of the costs of average yard and line delays for delivering each commodity by deciding on the proportion which will travel on each itinerary and the frequency (number of trains per day) for each service. The average yard and line delays for each itinerary are estimated based on queuing theory and the congestion on the lines and at the yards used. They assume that the capacity of each train is known and constant. Since the number of itineraries will be large, they use column generation based on a set of potential classification and transfer yards specified in advance for each commodity. They do not model restrictions on the number or volume of the blocks assembled at each terminal, but instead add an "accumulation delay" (based on a minimum block size specified *a priori*) in the objective which is larger for blocks with smaller volume and thus discourages the formation of small blocks. They suggest a heuristic solution for the relaxation of their model with the train capacity constraints incorporated in the objective function via a

penalty function. The quality of their solutions is difficult to assess since tight lower bounds are not available; however, Crainic [15] reports successful experiments of their model on Canadian and French railway data.

Haghani [23] proposes an MIP with non-linear objective to model the train routing and makeup decisions combined with empty car redistribution decisions. He proposes a heuristic solution which for computational tests provided solutions ranging from 2 to 19% from a lower bound.

Keaton [29] proposes an MIP formulation to select train routes and frequencies from a given superset and select a block sequence (from a few possible sequences specified *a priori*) to assign each commodity. Like Crainic, Keaton uses a service network but Keaton's does not include the block swap operations of set off and pick up. Ignoring constraints on the number of blocks formed at each yard and the length of each train, he is able to solve (to within 1.7%) a hypothetical problem with 26 terminals, 515 possible trains, and 333 commodities using a Lagrangian-based solution procedure. Keaton [31] proposes an IP (Integer Programming) formulation for deciding on the blocking plan and the routings and frequencies for trains. He suggests a heuristic for solving the formulation and obtains solutions with various constraints (such as number of blocks at each terminal or the length of trains) relaxed, but does not obtain solutions for the full formulation. In [30], Keaton quantifies trade-offs of adding more train connections to reduce transit times and concludes that significant reductions in transit time will require a large increase in the number of train connections and operating cost.

Huntley [27] describes the CARS (Computer Aided Routing and Scheduling) software which decides on the blocking plan and train routing and makeup. Simulated

annealing is used to generate blocking plans. This system provides "pert-like" visualization of the movement of each batch (cars with same origination, destination, and ready time). Currently supported platforms are the Macintosh and IBM/RS6000 (without graphics).

Gorman [21] proposes a genetic algorithm to solve a formulation similar to that of Keaton (1992) for the combined blocking and routing problem. He specifies a menu of possible train routes and exact schedules (time tables) as input. His model produced an operating plan based on data from a major domestic railroad which would have resulted in a 4% cost reduction and 6% reduction in late deliveries versus the operations plan actually used.

Aramacost [2] provides a comprehensive explanation of yard operations and proposes a two-machine sequencing model to plan yard activities.

**Related Literature** Klincewicz [32] gives an MIP formulation and heuristic solution to the general freight transportation problem of deciding whether to ship direct or via a single consolidation facility using a facility location formulation. He assumes unrestricted capacities at the consolidation facilities.

Several problems in the Less-than-TruckLoad (LTL) freight operations are similar to railroad problems. In LTL, a commodity is shipped from its origin to destination, stopping at various consolidation points where it is combined with other commodities into truckloads. The decision of what truckloads to use for each commodity in LTL is similar to the decision of what blocks to use for each commodity in our problem. However, blocks are relatively uncapacitated whereas truckloads have binding constraints on volume and

weight. The limited number of bays for consolidations at each LTL terminal may impose a node budget constraint similar to the one on the number of blocks in RBP.

Farvolden *et al.* [17] develop a primal partitioning algorithm for solving the multicommodity flow problem (path-based formulation) and demonstrate its use on real problems from the LTL industry. They do not model restrictions on the number of trucks connecting to each terminal. This model is intended to address the day-to-day operational decisions of an LTL company where operations at the terminals can be varied daily. They enforce service constraints via an underlying space-time service network. They present heuristics for this algorithm in [18]. A delete heuristic selects a profitable arc (truckload) to be dropped, and an add heuristic selects potential arcs to add.

Barnhart and Sheffi [8] propose a primal-dual heuristic to solve very large multicommodity flow problems. This approach is applied to the consolidation problem of LTL shipping. Barnhart *et al.* [10] propose a cycle-based formulation for the large-scale multi-commodity flow problem. They propose a solution based on solving a series of relaxations which give improved basic dual feasible solutions. Barring degeneracy they show that their algorithm converges in a finite number of iterations. Test cases show that this solution procedure outperforms existing methods.

**Comparison of Our Work to Previous Work in the Literature** Of the RBP solutions in the literature, our model is most similar to Bodin *et al.* —both models are MIPs which include constraints on the number and total volume of the blocks assembled at each terminal. However, their non-linear objective estimates the effect of increased congestion at each yard and on each link whereas our simpler, linear objective does not.

Our model accommodates different priority classes of traffic through the inclusion of priority constraints and has only a fraction of the binary variables of their model.

The model of Crainic *et al.* which includes decisions on the blocking plan, captures several other important planning decisions—train routing and the assignment of blocks to trains. However, their model is necessarily larger and their objective function is non-linear. Rather than including constraints on yard blocking capacities, their objective includes a queuing-based estimate of delays caused by congestion at the yards.

Compared to both the Bodin *et al.* and Crainic *et al.* models, we expect our model to be more tractable since we avoid non-linearities in the objective function.

### 3.5 Data aggregation schemes

To assess the practicality of our model, we obtained data from a major domestic railroad. The data for a peak month is extensive. It includes a database of over 50 Megabytes of records detailing each repositioning or *car-move* of each of the roughly 100,000 rail cars in their rail network for one month. These car-move records include the yards and blocks used and the priority class. The data also includes estimates on the blocking capacities—the number of blocks and total volume—for each terminal. In this section, we describe our methodology to aggregate the data to the level where only commodity flows and blocks between terminals are addressed.

The number of distinct nodes in the rail network is 336. Using this network for a one month period, there over 11,000 different OD pairs with positive flow. A BDP with this number of nodes and commodities and even a fraction of the possible $336^2$ arcs (representing blocks) is much larger than any BDPs which have been solved in

the literature. Furthermore, many of these nodes have only limited or no classification resources. In the peak month's data, there are fewer than 100 nodes which classify more than 100 cars—roughly one train—per day. These factors motivate us to define a subset of larger nodes—ones having some minimum level of classification resources—which we call terminals. In cooperation with the railroad, we identified 150 of these terminals.

In much of the literature on rail operations, an assumption is made that a subset of terminals are designated for reclassifications only; i.e., no commodity can originate or terminate at these terminals. We choose instead to identify a subset of "end" terminals where no reclassifications may occur. We have adopted this assumption in order to accommodate a policy decision by the railroad to restrict some terminals to exclusively servicing the traffic to or from their nearby area. Either the more common restriction of reclassification terminals or our restriction of "end" terminals serves to simplify the problem since the number of possible blocking paths is reduced by these assumptions. For our problem instance, there are 107 regular terminals at which classifications or reclassifications may be performed and a set of 43 "end" terminals at which reclassifications are not permitted.

Once we have reduced the network by deleting nodes which are not terminals, some commodities may not have an origin or destination in the network. For these commodities, endpoints which are not terminals must be replaced by terminals. We have considered two methods for making these assignments—one based on shortest paths and another based on historical practice by the railroad.

Using shortest paths, the terminal OD pair for the flow between each pair of nodes, can either be assigned based on the closest terminal to each endpoint node or as the

first and last terminal along the shortest path routing between endpoint nodes. In either case, the flows could be assigned to terminal OD pairs by using an efficient shortest path algorithm. However, since the links adjacent to smaller nodes sometimes have weight, height, or width restrictions, any assignment based on shortest paths would have to account for these restrictions. Further, if we assign the terminal OD pair based on the closest terminal to each node endpoint, then it is possible to assign a terminal which is in an opposite direction to the commodity's final destination node.

Another method for assigning the terminal OD pair is to use the historical practice of the railroad. Using the car-moves data, the terminal OD pair can be assigned as the first terminal and last terminal at which blocking has historically occurred. This type of aggregation will be more time consuming than using shortest paths because each of the several hundred thousand car-moves records must be examined; however, it avoids the problem of assigning traffic to links which can not accomodate it. On the other hand, it could bias the solution to the historical decisions while better solutions might be possible if the assignment to terminal OD's was allowed to vary.

Since we do not have data on the capacities of the links, we have chosen to use the latter method of assigning the same terminal OD that the railroad has historically used. Based on the aggregation of commodities to terminal OD pairs, the number of commodities is reduced from around 11,000 to around 1,300.

In the next chapter, we present a model and solution procedure for the blocking problem resulting from the aggregation described.

# Chapter 4

## Modeling the Railroad Blocking Problem

Based on the presentation of the last chapter, we may define the the RBP as:

> Minimize the costs of delivering all commodities by deciding which inter-terminal blocks to build and specifying the assignment of commodities to these blocks, while observing limits on the number and aggregate volume of the blocks assembled at each terminal and limits on the number of blocks used to deliver a commodity.

In this chapter, we show that the Railroad Blocking Problem (RBP) can be modeled as a BDP with side constraints. In the next section, we describe these side constraints and present an MIP formulation (BLOCK) obtained by adding them to PATH (presented on page 7) for BDP. As with PATH, the number of variables for this model is exponential, so we develop a column generation procedure which parallels the one we developed for PATH. We conclude the chapter by proposing a "branch-and-price" solution for RBP. In Chapter 5, we present computational experience for our solution approach.

### 4.1 RBP may be modeled as a BDP with side constraints

The RBP can be modeled as a network design problem where the nodes represent the railroad terminals and the arcs represent potential blocks. As with the general BDP, the RBP seeks to minimize the flow costs of delivering all commodities. The RBP constraint on the maximum number of blocks originating at a terminal has the

same form as the node-budget constraints which we included in the NODE (1.1) and PATH (1.3) formulations for the BDP. In RBP, the fixed cost $e_a = 1$ for all arcs. The balance equations (1.2a) on the paths in PATH can be used to model the identical balance requirements for BLOCK. However, unlike BDP, the maximum number of arcs (blocks) which may be used in a commodity's path is restricted by a priority constraint. The possibility of having different service constraints for some of the traffic from an OD pair requires adding a priority class to the description of each commodity; i.e., now commodities are identified by origin, destination, and a maximum number of handlings. Since blocks are assumed to be uncapacitated, the coefficients $u_a$ in PATH are replaced by the maximum possible flow for arc $a$. However, where the node flows are assumed uncapacitated in BDP, in RBP the nodes (terminals) have a flow volume constraint to model the limit on the total number of cars which can be classified. Finally, there is no overall budget for the blocking problem, so constraint (1.3d) from PATH may be omitted. We present an MIP formulation BLOCK for the RBP which applies the changes listed above to PATH and uses the following notation.

**Parameters**

$G = (N, A)$ is the graph with node set $N$ and candidate arc (block) set $A$.

$K$ is the set of all commodities $k$ designated by an origin-destination pair of nodes and the number of intermediate handlings allowed.

$v^k$ is the volume of commodity $k$ (in consistent units).

$orig(k)$ is the origin node for commodity $k$. $orig(a)$ is the origin of arc $a$.

$dest(k)$ is the destination node for commodity $k$. $dest(a)$ is the destination of arc $a$.

$\Delta(k)$ is the set of constraints on legal paths for commodity $k$.

$Q(k)$ is the set of extreme points of $\Delta(k)$.

$E(k)$ is the maximum number of extra (intermediate) handlings which may be performed on commodity $k$.

$R(k)$ is the set of terminal sequences representing candidate routings for commodity $k$.

$S(k)$ is the set of vectors representing subsequences of the routings in $R(k)$. These subsequences must start with $orig(k)$ and end with $dest(k)$. For $s \in S(k)$, each $s_a$ corresponds to a block which could be taken by commodity $k$.

$c_a \geq 0$ is the per unit cost of flow on arc $a$ (assumed equal for all commodities).

$u_a$ is the capacity of arc $a$.

$B(i)$ is the number of blocks which may originated at node $i$.

$V(i)$ is the volume which may be classified at node $i$.

$PC_q^k$ path cost for flowing one unit of commodity $k$ on path $q$.

**Decision variables**

$f_q^k$ proportion of commodity $k$ on path $q$, $\forall q \in Q(k), k \in K$.

$y_a$ is the binary design variable for including arc (block) $a$.

$$= \begin{cases} 1 & \text{if arc } a \text{ is selected,} \\ 0 & \text{otherwise.} \end{cases}$$

Based on this notation, we model RBP by the following MIP.

**Formulation**

$$\text{(BLOCK)} \quad min \quad \sum_{k \in K} \sum_{q \in Q(k)} PC_q^k \, v^k \, f_q^k \tag{4.1a}$$

*s.t.*

$$\sum_{k \in K} \sum_{q \in Q(k)} v^k \, f_q^k \, \delta_a^q - u_a y_a \leq 0 \qquad \forall a \in A \tag{4.1b}$$

$$\sum_{q \in Q(k)} f_q^k = 1 \qquad \forall k \in K \tag{4.1c}$$

$$\sum_{\substack{a \in A \\ orig(a)=i}} y_a \leq B(i) \qquad \forall i \in N \tag{4.1d}$$

$$\sum_{k \in K} \sum_{q \in Q(k)} \sum_{\substack{a \in A \\ orig(a)=i}} v^k \, f_q^k \, \delta_a^q \leq V(i) \qquad \forall i \in N \tag{4.1e}$$

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

$$y_a \in \{0,1\} \qquad \forall a \in A$$

The parameters $\delta^q$ are the extreme points of the convex hull of vectors that satisfy the set $\Delta(k)$ of constraints on legal blocking paths for each commodity. These constraints include the balance equations and priority constraints.

$$\Delta(k) = \{$$

$$\sum_{\substack{a \in A \\ orig(a)=i}} \delta_a - \sum_{\substack{a \in A \\ dest(a)=i}} \delta_a = \begin{cases} 1 & i = orig(k) \\ -1 & i = dest(k) \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in N \tag{4.2a}$$

$$\sum_{a \in A} \delta_a \leq E(k) \qquad \forall a \in A \tag{4.2b}$$

$$\delta \in S(k) \tag{4.2c}$$

$$\delta_a \in \{0,1\} \qquad \forall a \in A$$

$$\}$$

To summarize, the limit on the number of blocks which may be built at each node is modeled by the node-budget constraints (4.1d), the volume of cars which may be classified at each terminal is modeled by (4.1e) and the balance equations and constraints on the maximum number of handlings are included in the set of constraints $\Delta(k)$ on legal blocking paths for commodity $k$. $\Delta(k)$ also includes the restriction that blocking paths be consistent with the set of candidate routings $R(k)$ by the definition of the set $S(k)$.

Which costs to capture in the path cost expression $PC_q^k$ of the objective function is a modeling decision. Factors included could reflect costs for car-miles (miles that a car following this blocking path will travel) and costs for car-hours (hours of delay for classifications or block swaps incurred along this blocking path). Other costs which could be included are usage fees for links belonging to other railroads and premiums for using routings which are serviced by older, less efficient locomotives. Car-miles cost might vary depending on the grades involved since steeper grades require more fuel and possibly additional locomotive power. Labor and equipment costs per car might be different depending on the links and blocks for a commodity and could also be captured in $PC_q^k$ . Using the formulation presented, all of the costs captured in the expression $PC_q^k$ must reflect per car costs.

In order to include both the car-mile and car-hour costs, the tradeoff between mileage and handlings must be defined. We believe that this tradeoff will be difficult to quantify without large quantities of link-specific data. Since on average each handling incurs a one day delay, even if this tradeoff is captured, we anticipate that the costs for an extra handling will dominate the mileage costs among the shortest several routings for a commodity. Consequently, we chose to minimize the number of handlings required to

deliver all commodities and, rather than explicitly include the mileage costs, restrict the legal paths for each commodity via $\Delta(k)$ to be ones for which the car-miles are shortest. Thus $PC_q^k = v^k \sum_{a \in A} (c_a \delta_a^q)$ with $c_a = 1$.

We can exploit the sparsity of the rail network. The rail network is relatively sparse due to the high cost of installing new links—roughly \$1,000,000 per mile. For a major domestic railroad, there are on average fewer than two undirected links incident to each node. Consequently, there are likely to be only a few attractive routings for each commodity. For our solution to BLOCK we will assume that the set $R(k)$ of candidate routings for each commodity can be identified *a priori*. Since commodities may be split, different cars from the same commodity may take different blocking paths. The different blocking paths may or may not correspond to different underlying routings. Identifying the candidate routings in advance will facilitate the solution procedure that we propose for BLOCK and also implicitly limit the car-mile costs of our solutions.

We define *routings* to be paths through the physical network. It is convenient to describe a routing by the sequence of terminals visited. The terms *blocking path or commodity blocking assignment* on the other hand describe the path through the blocking network. If the blocking path is also identified by the sequence of terminals, a blocking path for a railcar will be a subsequence of the routing that it followed since blocking will be done at a subset of the terminals visited. We denote the candidate routes for commodity $k$ by $R(k)$ and the set of vectors representing possible blocking paths as $S(k)$.

In Figure 3.2 on page 33, we showed the physical rail network that underlies RBP for a major domestic railroad. Figure 4.1 contains a solution to RBP for the terminals identified in Figure 3.3 by solving BLOCK for this physical rail network.

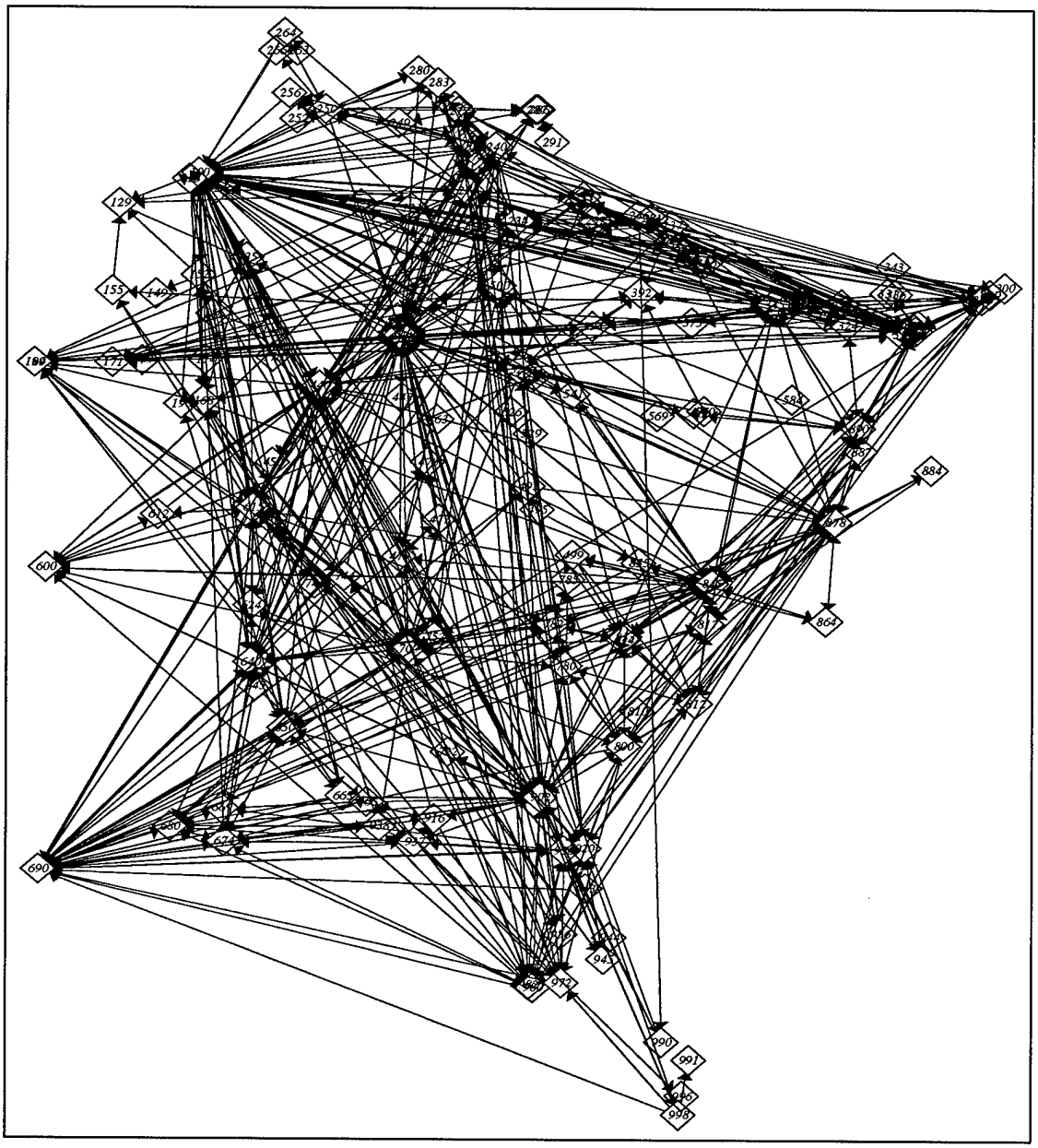


Figure 4.1: Solution for RBP for rail system of Figures 3.2 and 3.3

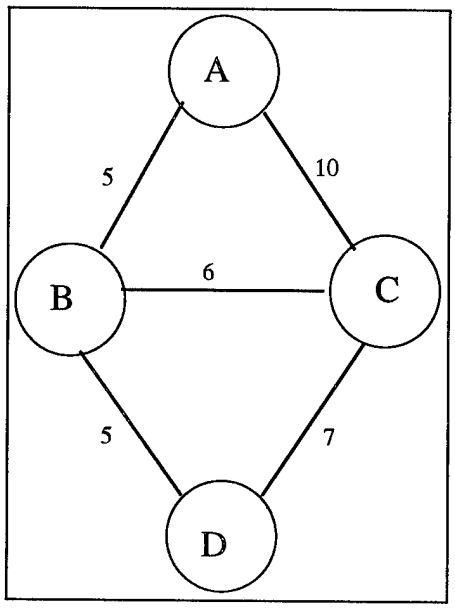


Figure 4.2: Physical network

We show the relationship between routings and blocking paths using the following example. Consider commodity A→D in the physical network of Figure 4.2 (for now we ignore the possibility of different priority classes). The shortest four routings are A-B-D (length 10), A-C-D (length 17), A-B-C-D (length 18) and A-C-B-D (length 21). We can represent all of the possible blocking paths that are consistent with these routings in the "commodity network" depicted in Figure 4.3. Any path in the commodity network from the source node to the destination node represents a blocking path for A→D. However, this blocking path may not be legal if it violates the priority constraints on the number of handlings.

While it may seem wasteful to use different nodes for each occurrence of the same terminal, duplicating nodes prevents us from forming a blocking path that is not a subsequence of a particular routing. Having separate nodes in each routing also allows

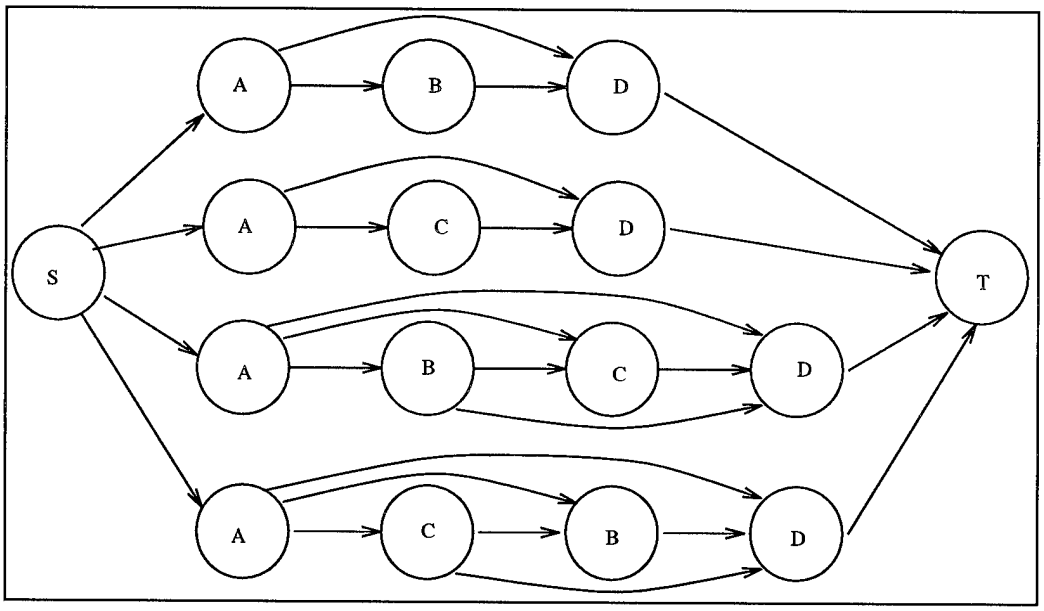


Figure 4.3: Commodity network for shortest 4 paths for commodity A→D

route specific penalties to be incorporated. Later, this network structure will allow us to accommodate the service constraint on the maximum number of blockings for the commodity.

## 4.2 Column generation to solve BLOCK

In BLOCK, the number of $f_q^k$ blocking path variables will be exponential. We show this next and then develop a column generation procedure to solve the LP relaxation of BLOCK analogous to the procedure for the LP relaxation of PATH in Chapter 2.

**Lemma 4.1** *The number of possible blocking paths for each commodity routing is exponential in the number of nodes in the routing.*

Let $r$ be the number of nodes in routing $R$ and represent this blocking path as a binary sequence with the *ith* digit corresponding to the *ith* node in $R$. There is a one-to-one correspondence between all binary sequences of length $r$ with the first and the last digit being one and the paths along routing $R$. There are $2^{r-2}$ such binary sequences, so there are an $2^{r-2}$ distinct blocking paths for the commodity which use routing $R$. □

**BLOCK - Master Problem** The BLOCK-MP master problem is the LP relaxation of BLOCK. We make the assignment of dual variables as indicated below.

$$\text{(BLOCK-MP)}\,min \quad \sum_{k\in K}\sum_{q\in Q(k)} v^k\, PC_q^k\, f_q^k \qquad DUALS \tag{4.3a}$$

s.t.

$$\sum_{k\in K}\sum_{q\in Q(k)} v^k\, f_q^k\, \delta_a^q - u_a y_a \le 0 \qquad \forall a \in A \qquad (\beta_a) \tag{4.3b}$$

$$\sum_{q\in Q(k)} f_q^k = 1 \qquad \forall k \in K \qquad (\chi_k) \tag{4.3c}$$

$$\sum_{\substack{a\in A\\ orig(a)=i}} y_a \le B(i) \qquad \forall i \in N \qquad (\theta_i) \tag{4.3d}$$

$$\sum_{k\in K}\sum_{q\in Q(k)}\sum_{\substack{a\in A\\ orig(a)=i}} v^k\, f_q^k\, \delta_a^q \le V(i) \qquad \forall i \in N \qquad (\nu_i) \tag{4.3e}$$

$$y_a \le 1 \qquad \forall a \in A \qquad (\mu_a) \tag{4.3f}$$

$$f_q^k \ge 0 \qquad \forall q \in Q(k), \forall k \in K$$

$$y_a \ge 0 \qquad \forall a \in A$$

The dual of (BLOCK-MP) is

$$\text{(DUAL)}\ max \sum_{k \in K} \chi_k + \sum_{i \in N} (B(i)\theta_i + V(i)\nu_i) + \sum_{a \in A} \mu_a \tag{4.4a}$$

s.t.

$$\sum_{a \in A} [(v^k \delta_a^q)(\beta_a + \nu_{orig(a)})] + \chi_k - v^k PC_q^k \leq 0 \quad \forall k \in K, q \in Q(k) \tag{4.4b}$$

$$\sum_{a \in A} \Big( \mu_a - u_a \beta_a \Big) + \sum_{i \in N} \theta_i \leq 0 \tag{4.4c}$$

$$\beta_a, \theta_i, \mu_a, \nu_i \leq 0$$

$$\chi_k \text{ free.}$$

In DUAL, constraints (4.4b) correspond to the variables $f_q^k$ and (4.4c) correspond to the variables $y_a$. Since the number of possible blocking paths $\delta^q$ is exponential we will populate the master problem with a subset of them. Omitted columns $\delta^q$ with favorable reduced cost will correspond to violated constraints (4.4b). After solving BLOCK-MP over a subset of the blocking paths and obtaining values for the dual variables, we solve a separation problem to identify violated constraints (4.4b). These violated constraints identify new paths with negative reduced cost to add to BLOCK-MP. See Figure 2.1 on page 19 for a flowchart of the column generation procedure.

**BLOCK - Pricing Subproblem for Commodity k** The objective for the pricing subproblem $SP_k$ is chosen to identify paths $\delta^q$ for which dual constraint (4.4b) is violated. The set of constraints for $SP_k$ is the set of restrictions on legal blocking paths which is contained in $\Delta(k)$. Given duals $\beta_a, \chi_k, \nu_i, \mu_a$ from the solution of the restricted master problem (RMP), $SP_k$ is defined as

$$(SP_k) \quad min \quad \sum_{a \in A} v^k \, \delta_a^q \, (-\beta_a - \nu_{orig(a)}) - \chi_k + PC_q^k \, v^k \tag{4.5a}$$

s.t. (4.5b)

$$\text{Balance} \quad \sum_{\substack{a \in A \\ orig(a)=i}} \delta_a^q - \sum_{\substack{a \in A \\ dest(a)=i}} \delta_a^q = \begin{cases} 1 & orig(k) = i \\ -1 & dest(k) = i \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in N \tag{4.5c}$$

$$\text{Service} \quad \sum_{a \in A} \delta_a^q \leq E(k) \tag{4.5d}$$

$$\delta^q \in S(k) \tag{4.5e}$$

$$\delta_a^q \in \{0,1\} \qquad \forall a \in A$$

If the objective value of $SP_k$ is less than zero, then the solution identifies a path with negative reduced cost for commodity $k$ and we add this column to RMP. Next, we present a solution procedure using shortest paths to solve the $SP_k$ subproblems.

## 4.3 Solving $SP_k$ by shortest paths

If the service constraint (4.5d) on the number of blocks used to deliver a commodity were not present then $SP_k$ could be solved, as suggested in Chapter 2 for the BDP, by finding a shortest path on the graph with nodes (terminals) and arcs (blocks) having a cost function dictated by the objective of $SP_k$ . We prefer network-based solution algorithms rather than general IP techniques to solve the $SP_k$ problems since they will be solved frequently. Two solutions which can accommodate the service priority constraints are described below. Assuming that the path cost is defined as $PC_q^k = v^k \sum_{a \in A} (c_a \delta_a^q)$, we can minimize the objective by solving for the shortest path using the arc cost function $(-\beta_a) + c_a$. The two options below enforce the constraint on the maximum number of handlings differently.

**Using a generic network and a label-based shortest-path algorithm.** Let $G = (N, A)$ be the (non-acyclic) graph with nodes representing terminals and arcs representing possible blocks. For this problem, non-dominated paths refer to paths for which the distance to the node is the minimum for all paths with at least the same number of arcs; i.e., there are no other paths to this node with both a smaller distance and fewer handlings. We must track all of the non-dominated paths to each node since we will not know until we reach the end of each path whether the path is feasible with respect to the number of handlings. Consequently, any labeling algorithm used to solve $SP_k$ on this network must maintain the following information for each non-dominated path to each node.

- distance: sum of the flow costs for the arcs used in path,
- handlings: number of arcs used in path, and
- predecessor in this path.

This procedure has the advantages of not requiring the identification of candidate routings *a priori* and the same graph can be used for all commodities. However, the graph is cyclic and fairly large—$|N|$ nodes and $|A|$ arcs with each node requiring information on the distance, number of handlings, and predecessor for as many as $E(k)$ non-dominated paths.

**Using Commodity Networks** Let $G$ be the commodity network as shown in Figure 4.3 on page 57. As for the generic network described above, we can generate new blocking paths for a commodity which adhere to the priority constraints by using a shortest path, labeling algorithm and tracking information on all non-dominated paths. Forming

commodity networks, however, requires identifying the permissible routings for each commodity *a priori*. But, each such commodity network is acyclic and will likely be smaller than the generic network, so solving for the shortest path will be faster if we use special acyclic shortest path algorithms. The storage required for all of these commodity networks (and the associated information on the non-dominated paths to each node) will be larger.

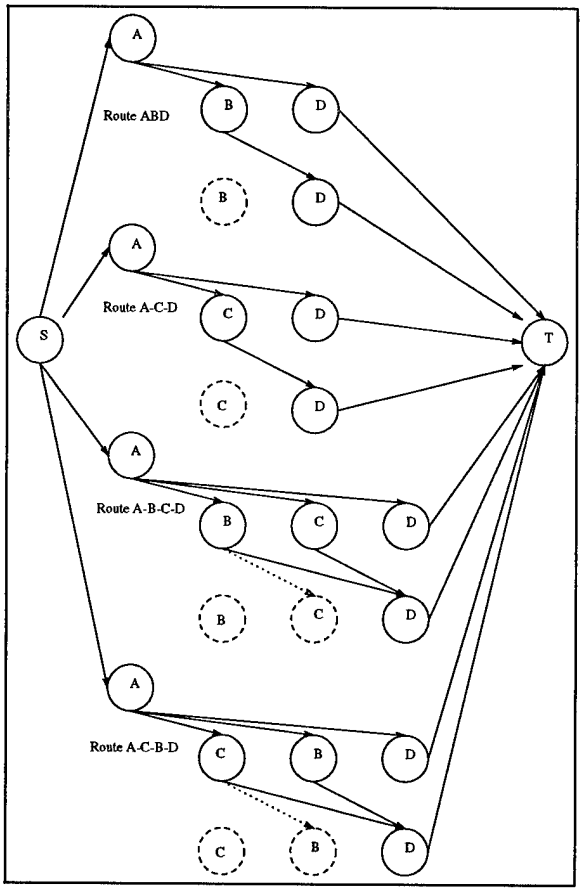


Figure 4.4: Space-time commodity net commodity A→D with $E(k) = 1$

We next describe the construction of a space-time version of the commodity network for which the single shortest path always adheres to the priority constraint without the need to track all non-dominated paths. We thus trade storing networks with more nodes for the ability to use a simpler shortest path algorithm. Whether the space-time version of the commodity networks will require more memory than the commodity networks plus the information on the non-dominated paths will depend on the implementations of both.

Let $G$ be the commodity network for commodity $k$ with a maximum number of intermediate handlings of $E(k)$ or equivalently a maximum number of handlings of $E(k)+1$. Replicate each node except those representing the commodity's origin $E(k)$ times. (Since $E(k)+1$ bounds the total number of handlings to the destination, no paths to other nodes which require more than $E(k)$ handlings need be considered.) The $j$th copy of each node will denote that the associated terminal was the $j$th terminal at which the commodity was blocked. Next add arcs to reflect the possible blocks and join the copies of the sink node to a supersink for the commodity. An example space-time commodity network with $E(k) = 1$ is shown in Figure 4.4. Note that some copies of nodes cannot be used in any path from the supersource to the supersink and can be deleted. These are depicted by dashed circles.

For our computational work, we have chosen the last option—using the space-time commodity networks. Thus, we trade storing a larger network which will only be constructed once for a faster algorithm which will be executed frequently.

Another advantage of using either the commodity networks or the commodity space-time networks is that some additional path costs can be modeled easily. For example,

the costs for car-miles for each routing could be placed on the arcs connecting the source node for the commodity to the first node for each routing. (The car-mile costs could also be put on each arc in the generic network, but the car-mile cost for the routing will be the same regardless of the blocking path, so it makes sense to just store the total once.) In a similar manner, the commodity networks can capture costs which depend on the underlying routing but which can not be disaggregated to the individual links involved. For instance we might want to attach a penalty to a routing which is serviced by older less-efficient locomotives. If these locomotives only service cars using all of the links in the routing, then the penalty may not be assigned to the individual links and, therefore, may not be modeled in the generic network.

## 4.4 Strengthening BLOCK-MP with additional cuts

In an analogous manner to §2.4, we add "forcing constraints" which are a dissagregation of the bundle constraints (4.3b) to BLOCK-MP to strengthen its LP relaxation. The "forcing constraints" have the form:

$$\sum_{q \in Q(k)} f_q^k \, \delta_a^q - y_a \leq 0 \quad \forall a \in A, k \in K. \tag{4.3g}$$

and require that the proportion of any commodity using block $a$ must be smaller than the binary design variable $y_a$ for the block. These cuts may be added via the polynomial separation algorithm of §2.4; however, in our computational work, we have been able to add all such cuts in advance.

Let $\phi_a^k \leq 0$ be the dual associated with the forcing constraint for commodity $k$ and block $a$ then, adding the forcing constraints to BLOCK-MP changes the dual constraint

(4.4b) corresponding to the blocking path variables $f_q^k$ to

$$\sum_{a \in A} \{\delta_a^q (v^k \beta_a + v^k \nu_{orig(a)} + \phi_a^k)\} - \chi_k - v^k PC_q^k \leq 0 \quad \forall k \in K, q \in Q(k) \qquad (4.4b')$$

Correspondingly, the objective of $SP_k$ becomes

$$min \quad \sum_{a \in A} \{\delta_a^q (-v^k \beta_a - v^k \nu_{orig(a)} - \phi_a^k)\} - \chi_k + v^k PC_q^k$$

Assuming the path cost expression $PC_q^k = \sum_{a \in A} c_a \delta_a^q$, this objective may be minimized by solving for a shortest path with arc cost

$$-v^k \beta_a - v^k \nu_{orig(a)} - \phi_a^k + v^k c_a. \qquad (4.6)$$

**Solving $SP_k$ simultaneously via a supergraph** Let $SG$ be the supergraph with a component for each commodity's space-time network (Figure 4.4) and a supersource node added which is connected by a zero cost arc to the source node of each of these components. Then we can solve all of the $SP_k$ simultaneously by solving for the shortest path from the supersource to the destination node of each component. Since there is a different copy of the arc representing each block in each commodity's component of $SG$, we may include the duals for the forcing constraints when calculating the distance labels. Consequently, even with the forcing constraints added, we can generate a new legal blocking path for each commodity by a single execution of an acyclic shortest path algorithm on $SG$. While this simultaneous solution has the same complexity as solving each $SP_k$ separately, it reduces the number of function calls to the shortest-path function from $|K|$ to one.

## 4.5 Branch-and-price algorithm

Since integrality of the $y_a$ is required to solve the original problem, we embed the column generation procedure described in §4.2 into a branch-and-bound search to obtain integral solutions. Elements of this branch-and-price algorithm include preprocessing, a rounding heuristic, a branching rule, and a non-LP based lower bound, as well as, the column-generation LP bound. The overall flow of our algorithm can be seen in the flowchart of Figure 4.5.

### 4.5.1 Preprocessing

Based on the candidate routings for each commodity, we can identify some arcs which must be included. Commodities for which no extra handlings are permitted ($E(k) = 0$) force the corresponding $y_a = 1$ for the direct arc from their origin to their destination. Similarly, commodities for which the only candidate routing has two terminals force $y_a = 1$ for the arc (block) between these two terminals. We preprocess to identify arcs meeting either of these conditions for any commodity and then fix the $y_a = 1$ for these arcs. Additionally, based on the candidate routings for each commodity, we calculate the maximum flow for each arc (block) and reduce the arc capacity bound $u_a$ (which we assumed was infinite) accordingly.

### 4.5.2 Rounding heuristic

As with most integer programs solved by branch-and-bound, we expect that a rounding heuristic to convert fractional solutions to valid primal (integer) solutions will be important. Tighter primal solutions will allow fathoming nodes more quickly since we

can fathom any node whose LP lower bound is higher than a known integer solution. However, since this heuristic is computationally expensive, it will likely only be used occasionally in the search. We describe our primal heuristic, provide the algorithm, and describe the effect of including the primal algorithm on a test case.

The arcs for which $y_a$ is positive in the LP relaxation are more likely to be present in a good IP solution, so we eliminate any arcs $a$ for which $y_a = 0$ in the LP solution. Then, we perform a branch-and-bound search for an integer solution as follows. At each branch-and-bound node, we reoptimize RMP and select a fractional $y_a$ variable on which to branch. We perform a depth first search examining the sub-tree where $y_a$ is excluded from the solution first. Each branching eliminates one more fractional $y_a$. We fathom any branch-and-bound node if the objective function value is more than 10% higher than the LP objective value for the current LP bound from the larger branch-and-price algorithm. We terminate after a finite number of branch-and-bound nodes, whether an integer solution was found or not. If we choose to branch on a fractional $y_a > 0.8$, we skip the sub-tree where $y_a = 0$. We terminate the search after examining 150 nodes, since there is no guarantee that this search will lead to an integer solution.

**Intuition** Consider a fractional $y_a$ and the paths $q_1, ..., q_m$ which include arc $a$. The flows of any commodities using these paths must be fractional or the forcing constraints (4.3g) would require $y_a = 1$. Fixing $y_a$ to zero will prevent flows of any commodity on $q_1, ..., q_m$, so the $y_a$ corresponding to any other arcs on these paths may now be reduced to zero in the LP solution. Thus, we expect that by setting $y_a$ to zero in the left branch, several other $y_a$ will also be reduced to zero when the LP is re-optimized. Setting $y_a$ to one in the right branch will reduce the amount of the node-budget capacity $B(i)$ which is

allocated among the remaining fractional $y_a$ and may cause some of these $y_a$ to become zero when the LP is re-optimized.

**Pseudo Code** We give the pseudo-code for the primal heuristic below and the flowchart for it in Figure 4.6.

**Main**

1. Count the number of fractional $y_a$. If over 5% stop.
2. Fix all $y_a$ currently equal to zero at zero.
3. Call DFS_primal()

**DFS_primal** (Depth First Search Primal Heuristic)

1. If all $y_a$ are integer return primal solution.
2. If depth $> 150$ return failure
3. Reoptimize using dual simplex
4. If current solution is 10% from LP bound or worse than a primal solution, return to calling function.
5. Pick a fractional $y_a$
6. If $y_a < 0.8$
   (a) Set $y_a = 0$
   (b) Call DFS_primal recursively
7. Set $y_a = 1$
8. Call DFS_primal recursively

**Effectiveness** A test problem, without the primal heuristic, required 500 branch-and-bound nodes and 265 minutes to close the integrality gap to 0.18%. With the primal heuristic added, this problem required 41 branch-and-bound nodes and 88 minutes (including five minutes spent in the primal heuristic) to close the integrality gap to 0.17%. The primal heuristic was able to identify an integer solution in about five minutes among the columns present in RMP after the execution of the root node of the branch-and-price algorithm. Because this heuristic is computationally expensive, requiring the re-optimization of an LP solution for each of its branching decision, and may not find an integer solution, it is used sparingly. For this problem instance, the primal heuristic was only executed once. In our computational work reported in Chapter 5, we execute the primal heuristic at the root node and every ten nodes thereafter, if the gap between the best integer solution found and the LP bound from the current node is greater than 0.5%.

### 4.5.3 Branching rule

If we branch on the $y_a$ variables, we may generate columns after branching using the shortest path solution for $SP_k$ if we delete arcs which have been excluded from the solution by branching. Since we use a labeling-algorithm for solving $SP_k$ , we simply ignore these excluded arcs when calculating the distance labels. So, generating additional columns within the branch-and-bound tree is no more difficult than generating them at the root node. The standard approach of branching on the fractional binary variable closest to 0.5 will work. However, we chose to branch on the fractional $y_a$ variable for which $i = orig(a)$ has the minimum node-budget value $B(i)$.

### 4.5.4 Lower bounds

The LP-relaxation provides a fairly tight bound for each node in the branch-and-bound tree; however, it must be solved to optimality before a node can be fathomed. Since we are using column generation, this requires solving the pricing subproblems until no new columns are generated. Consequently, at each node, we compute a faster lower bound which may allow fathoming before computing the LP bound. Below, we describe two polynomial lower bounds (due to Dionne & Florian and Gallo) for the generic BDP which are also valid for RBP. For the blocking problem, these bounds can be computed by solving for two shortest-path trees and partially sorting a vector with an entry for each potential block.

**Dionne-Florian Lower Bound** Let $d_a$ be a lower bound on the cost incurred in the objective if arc $a$ is removed. Then a lower bound is

$$LB = \sum_{a \notin I} d_a$$

where $I$ is any set of arcs such that the node-budget constraints (4.1d) of RBP are met. Let $L(k, G)$ be the shortest path distance from $orig(k)$ to $dest(k)$ in graph $G$. For the Dionne-Florian bound adapted for multiple commodities for each $OD$ pair,

$$d_a = \sum_{\substack{k \in K \\ orig(k)=orig(a) \\ dest(k)=dest(a)}} [L(k, G-a) - L(k, G)]v^k$$

We can simultaneously generate all of the $L(k, G)$ by finding the shortest path tree rooted at the supersource in the supergraph $SG$ described in §4.4. Arcs representing

blocks which have been excluded from the solution via branching decisions are ignored. Next we generate all of the $L(k, G-a)$ by solving for the shortest path on the same network where, for each commodity, we also delete the arc which represents a direct blocking. $L(k, G-a)$ where $a = (orig(k), dest(k))$ is then the distance to the supersink for commodity $k$. Summing $(L(k, G-a) - L(k, G)$ over commodities with origin and destination matching the arc $a$ provides $d_a$.

Once the $d_a$ are known, the Dionne-Florian Lower Bound ($DF_{LB}$) requires solving this knapsack problem for each terminal.

$$max \sum_j d_a y_a \tag{4.7}$$

$$s.t. \sum_j e_a y_a \leq B(orig(a)) \tag{4.8}$$

$$y_a \in \{0, 1\} \tag{4.9}$$

Since in RBP, the coefficients $e_a$ are all one, this problem can be solved by sorting the $d_a$ for blocks which originate at the terminal, so that, $d_1 > d_2 > d_3 \ldots$. Then ($DF_{LB}$) is solved exactly by choosing to include the first $B(i)$ blocks; i.e., the blocks whose exclusion would be the most costly. Once the $y_a$ are thus determined,

$$DF_{LB} = \sum_{k \in K} v^k + \sum_{a \in A} (1 - y_a) d_a$$

So, the Dionne-Florian bound includes one handling for every commodity and adds the lowest valued penalty terms for the collection of blocks from each terminal which may not be included because of the budget constraint.

At the root node, all $L(k, G) = 1$ since each commodity will have the shortest path of a direct blocking from origin to destination. When this direct blocking is deleted for the calculation of $L(k, G-a)$, then if there is a routing with any intermediate terminals, the new shortest path will have length 2, corresponding to a blocking path that stops at exactly one intermediate terminal. Any other blocking paths would have length greater than 2. If there no candidate routings which have an intermediate terminal, then $L(k, G-a)$ will be infinity. Additionally, if $E(k) = 0$ for a commodity, then no blocking paths other than direct blocking are feasible, so $L(k, G-a) = \infty$. Let $vol(o, d)$ be the sum of the volumes for commodities with origin $o$ and destination $d$. Then, $d_a$ will either be infinite or $vol(orig(a), dest(a))$ and by including the blocks with the highest $d_a$, $DF_{LB}$ will have the interpretation of direct blocking the highest volume $OD$ traffic and traffic with only one possible blocking and blockings with one reclassification for the lowest volume $OD$ traffic.

At the root node, this bound simply states that we can do no better than having one extra handling for the lowest volume traffic which doesn't have to be blocked directly because of a priority service constraint or because there are no other possible blocking paths using the candidate routings.

The bound at the root node which we just discussed depends on the candidate routings. We can modify this bound so that it's independent of any routing assumptions as follows. Replace any $L(k, G-a) = \infty$ with $L(k, G-a) = 2$; i.e., we assume that there is some alternate routing with an intermediate terminal for each commodity. This bound is based on direct blockings for the commodities which have priority constraints requiring direct blocking, $E(k) = 0$, and the highest volume traffic and on blockings with

one reclassification for the remaining lowest-volume commodities. This bound provides an absolute lower bound for solutions to RBP. We refer to this bound as DF-independent and use it to gauge the quality of solutions in our computational work.

As we mentioned in the literature review of BDP, one of the lower bounds proposed by Gallo [19] is very similar to the Dionne-Florian bound and dominates it. The procedure above can be adapted as follows to compute this Gallo lower bound.

Let $s(k,a) = 1$ if arc (block) $a$ is selected for exclusion from commodity $k$'s delivery and zero otherwise. Exactly one arc from the current shortest path is selected for deletion for each commodity; however, this selection is heuristic. Then,

$$d_a = \sum_{k \in K} [L(k, G-a) - L(k,G)] v^k \, s(k,a).$$

In the Dionne-Florian bound only commodities whose origin and destination were the same as arc (block) $a$ were included in $d_a$. Consequently, commodity $k$ would not participate in any $d_a$ once the direct block (arc) for it had been fixed out of the solution by branching. In contrast, for the Gallo bound every commodity is included in the estimate of one $d_a$. Gallo recommends assigning the arc in the current shortest path with the highest fixed cost. For RBP (with equal fixed costs $e_a$), we recommend selecting the arc $a = (i,j)$ in the current shortest path for which $B(i)$ is minimum.

At the root node, this bound will have the same value as the Dionne-Florian bound. At all nodes, the Gallo bound dominates the Dionne-Florian bound [19]. The trade-off is the extra time required to assign an arc (block) to each commodity.

**A bound from the Root LP** The solution of the LP relaxation of the root node also provides a lower bound which is based on the candidate routing set for each commodity. Allowing larger candidate routing sets (or all routings), our branch-and-price algorithm may not identify an integer solution in a reasonable amount of time. However, the final LP bound for the root node (with additional routings) could be compared to our best integer solution with fewer routings to give a bound on the maximum improvement possible for adding the additional routings.

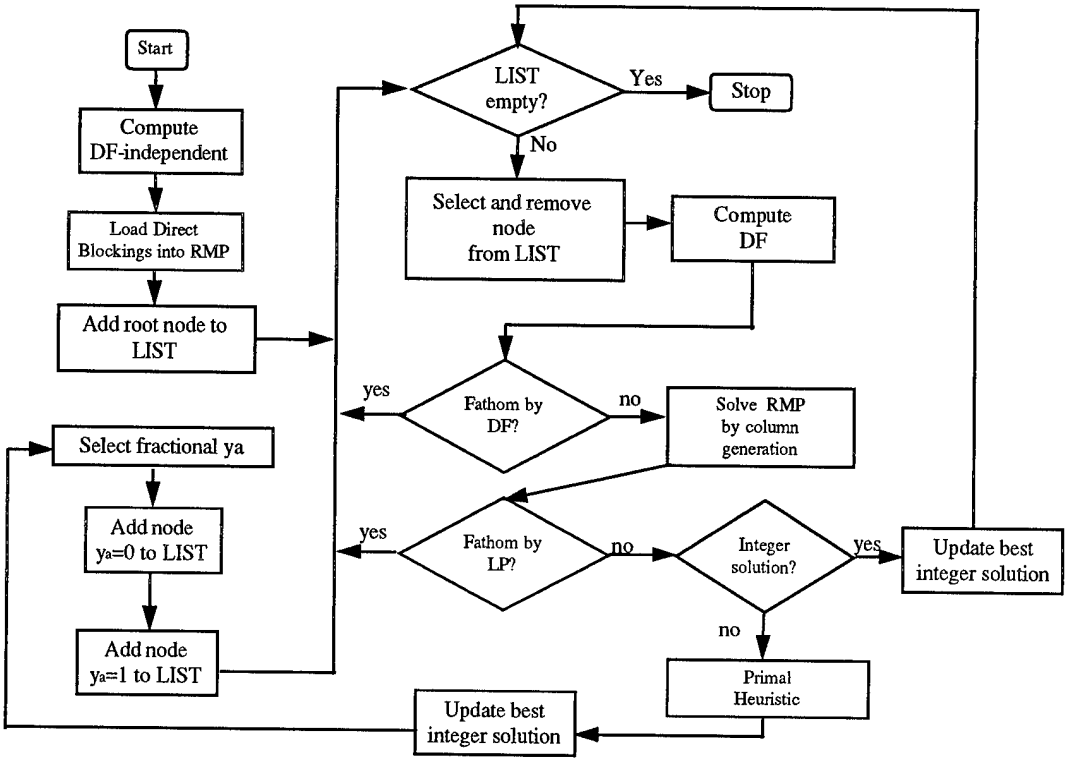


Figure 4.5: Branch-and-price algorithm flowchart

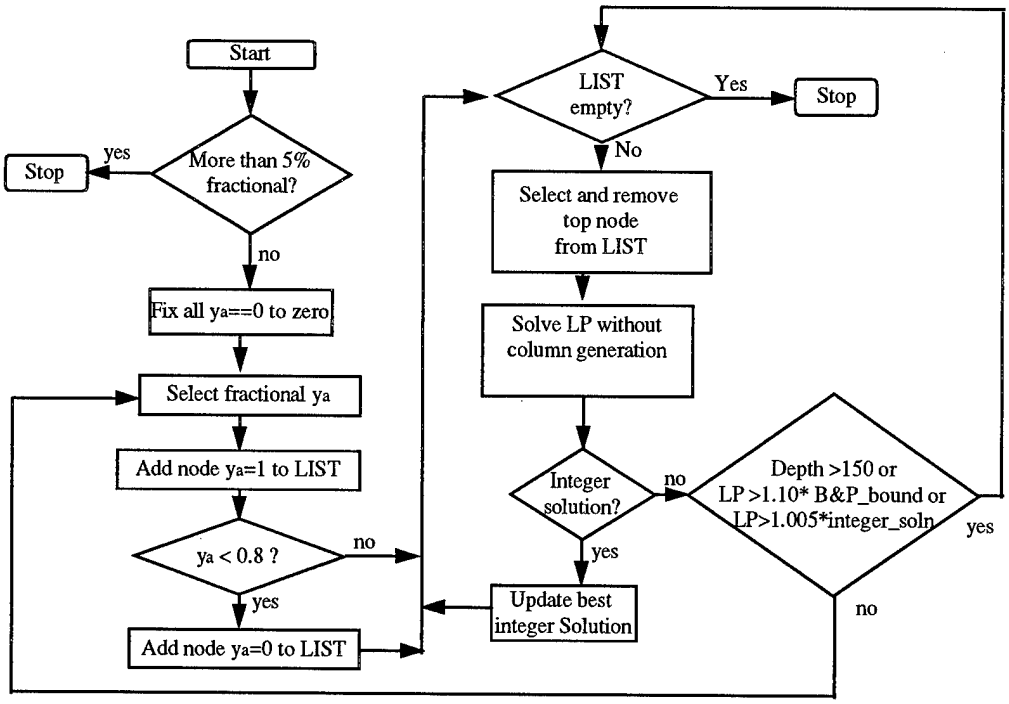


Figure 4.6: Flowchart for primal heuristic

# Chapter 5

## Computational Results

In this chapter, we provide computational results from an implementation of our branch-and-price algorithm. These tests provide proof of concept for our algorithm and help identify the input parameters to which our algorithm is most sensitive. The solution times reported are for a Sparc 20 workstation with a single 75 MHz processor and 128 Megabytes of RAM running SunOS 5.5. Our code uses the MINTO library (Mixed INTeger Optimizer – version 2.2a) by M.W.P. Savelsbergh [38] and the CPLEX version 4.0 callable library [13]. This chapter is organized into sections describing our computational testing, results on various problem sets, a statistical analysis of the results of random test instances, and ideas which may improve the execution speed of our algorithm.

### 5.1 Description of the computational tests

We implement our branch-and-price algorithm by restricting the routings for each commodity and adding an early stopping rule. Our restriction on routings implies that the integer solutions found are upper bounds on the solution to the problem without these restrictions. We stop our algorithm when the integrality gap is within a specified tolerance. Also, for convenience in conducting the tests, we limit the execution time to four hours.

As discussed in §4.1, two important costs for implementing a blocking plan are the costs for car-handlings and costs for car-miles. By restricting the candidate routes

for each commodity to ones with the shortest distance, the car-miles cost is implicitly limited. We then define the path cost $PC_q^k$ so that the objective function minimizes the costs for car-handlings. We assume that the costs for car-handlings are equal at all terminals and let $PC_q^k = v^k \sum_{a \in A} c_a \delta_a^q$ where $c_a = 1$ for all arcs. This definition of $PC_q^k$ gives our objective function the interpretation of minimizing the total number of handlings required to deliver all commodities. As discussed in §4.1, we can include other types of costs as long as they can be expressed as flow costs within the commodity networks shown in Figure 4.4.

Let $\boldsymbol{R}(k) = (R_1^k, R_2^k, ...)$ be the ordered set of candidate routings for commodity $k$. $R_1^k$ is the railroad's current routing and the remaining routings are the shortest path routings in increasing order of their distances. We include only routings that have a distance within 50% of the shortest routing. For each problem instance, we define the input parameter RTS which denotes that legal blocking paths are based on the first RTS routings in $\boldsymbol{R}(k)$ for each commodity $k$.

We initially populate the restricted master problem RMP (4.1) with direct blockings for each commodity. We add separate artificial variables with a high objective coefficient for each terminal to the blocking capacity constraint (4.1d) and the volume capacity constraint (4.1e) so that a feasible solution may always be obtained for the restricted master. We add all the forcing constraints at the root branch-and-bound node before generating any new blocking paths. The primal heuristic is executed at the root node and every ten nodes thereafter, if the gap between the best integer solution found and the current LP bound exceeds 0.5%.

| RTS | Poss_Paths | $\|A\|$ | Fixed | FCs | Rows |
|---|---|---|---|---|---|
| 1 | 8,126 | 2,332 | 337 | 4,919 | 8,785 |
| 2 | 127,083 | 5,776 | 226 | 24,696 | 32,006 |
| 3 | 259,392 | 6,415 | 217 | 30,452 | 38,401 |
| 4 | 389,661 | 6,802 | 215 | 34,850 | 43,186 |

Table 5.1: Problem sizes as a function of the number of candidate routings

For all the problem instances, we first aggregate the real-world data to the 150 largest terminals as explained in §3.5. Some of the statistics on the problem instances only differ with respect to RTS—the number of candidate routes selected from the set $\boldsymbol{R}(k)$ for each commodity. We provide these statistics on the problem sizes in Table 5.1. Note that since we restrict the routings to ones within 50% of the shortest paths, some commodities will have fewer than RTS routings included. The column "Poss_Paths" is the total number of blocking paths for all commodities along any of their candidate routings. The number of possible blocks for these problems is the number of terminal pairs, $150^2$. By eliminating the terminal pairs that cannot occur as a subsequence of any of the commodity's candidate routings and thus cannot occur in any legal blocking path, we may reduce the number of potential blocks to $|A|$. A binary $y_a$ variable corresponds to each of these potential blocks. By preprocessing the candidate paths and priority constraints, some blocks may be identified which must be included in the blocking plan. The number of these blocks is reported in the column "Fixed." The fixed $y_a$ correspond to a commodity with either a priority constraint of zero intermediate handlings ($E(k) = 0$) or without a candidate routing with an intermediate terminal. The number of forcing constraints is listed in the column "FCs" and the total number of rows (including the forcing constraints) for the master problem is listed under "Rows."

Our first set of problem instances is derived from the real world RBP for data provided from a major domestic railroad with different sizes for the candidate route sets. Another set of problem instances (which we refer to as "scaled problems") is derived by scaling the blocking resources for the real world data by constant factors. Our final set of test problems is generated by selecting the classification resources at each terminal from a random distribution. We explain the methodology for arriving at the scaled and random problem instances below.

We can test the sensitivity of our solution procedure to changes in the tightness of the blocking capacity constraints by either varying the commodity data or the terminal data. For example, to test the sensitivity to changes in the tightness of the volume constraints (4.1e), we can either change the terminal volume capacities $V(i)$, vary the volumes $v^k$ of the existing commodities, or add new commodities. Since every commodity must be blocked at its origin, any RBP instance is infeasible if $V(i)$ is lower than the total volume of commodities with origin $i$. If we vary the commodity volumes or add new commodities between problem instances, then this floor value for feasible $V(i)$ will have to be recomputed for each problem instance. Similarly, we may tighten the constraints on blocking capacity (4.1d) by either reducing the maximum number of blocks for each terminal or by adding additional commodities. If we add new commodities, then the set of $y_a$ that are fixed by the preprocessing step may change. The increased complexity for solving an MIP with more binary variables could mask the effect caused by tightening the blocking capacity constraints. Consequently, we choose to test for sensitivity to changes in the tightness of both capacity constraints by varying the terminal capacities $B(i)$ and $V(i)$.

| RTS | Megabytes |
|---|---|
| 1 | 0.6 |
| 2 | 1.9 |
| 3 | 3.1 |
| 4 | 4.2 |
| ... | ... |
| 10 | 6.5 |

Table 5.2: Megabytes required to store the commodity networks

Each problem instance for the test set using scaled capacities is defined by the real world data, the number of candidate routes RTS for each commodity, and two scaling factors—BLK and VOL. BLK is the proportion of the original blocking capacity $B(i)$ to use for each terminal. VOL is the proportion of the original volume capacity $V(i)$ to use for each terminal. If the $V(i)$ for any terminal falls below the floor feasible value of the volume of commodities originating at $i$, then it is reset to the floor value. Similarly, if $B(i)$ for any terminal falls below three, then we reset it to three. Otherwise $i$ would no longer meet our definition that a terminal has significant classification resources. We use scaling factors as low as 0.75 that match a reduction in capacities of 25% from the original data.

We convert each of the scaled test instances to several random instances by selecting each $B(i)$ and $V(i)$ from a random distribution. Let (BLK,VOL,RTS) describe a scaled test instance and $B(i)$ and $V(i)$ be the original capacities for each terminal. Random test instances are defined by drawing $B(i)$ and $V(i)$ from the discrete uniform distribution with mean BLK times $B(i)$ and VOL times $V(i)$, respectively, and variation of ten percent of the mean. As with the scaled test instances, we reset any terminal capacities that fall below the respective minimum value to the minimum value.

The number of binary variables and the number of possible blocking paths for our model does not depend on the terminal blocking capacities but only on the candidate routes for the commodities. So, for given sizes of the candidate route set for each commodity, the formulations for the problem instances in all three sets of test problems will have the same candidate block set ($A$) and the effects of preprocessing will be the same. Similarly, the supergraph $SG$ constructed to solve the subproblems $SP_k$ will only depend on the number of candidate routings. For different RTS levels we provide the storage required for $SG$ in Table 5.2. These sizes would change if the routings were not restricted to ones within 50% of the shortest routings. However, with this pragmatic restriction these network are not too large for efficient processing on a workstation.

## 5.2 Results for the sets of problem instances

### 5.2.1 Problem instances for the real world data

Results for solving the real world problem with up to four candidate routes for each commodity are given in Table 5.3. These problem instances differ only in the number of candidate routings RTS allowed for each commodity. We set the required integrality gap to 0.25%. Our algorithm will terminate when an integer solution is found within this tolerance of the LP lower bound which is based on the restricted routings for the problem instance. The number of $f_q^k$ variables which are generated when solving the LP relaxation of the root node and the total number of columns generated are listed next. The number of columns generated for each problem instance may be compared to the maximum number possible in Table 5.1. The most dramatic comparison is for the case

| RTS | Columns Generated Root | Total | Minutes Root LP | Primal | Total | B&B | LPs | PR/LP | Solution Quality I-Gap | DF-Gap |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2577 | 2577 | 1.07 | 0.12 | 1.19 | 2 | 12 | PR | 0.05% | 2.33% |
| 2 | 1237 | 3224 | 19.09 | 0 | 19.09 | 1 | 20 | LP | 0.00% | 1.57% |
| 3 | 3354 | 3354 | 43.09 | 0.93 | 44.02 | 2 | 28 | PR | 0.00% | 1.52% |
| 4 | 1237 | 3424 | 67.27 | 0 | 67.27 | 1 | 29 | LP | 0.00% | 1.46% |

Table 5.3: Results for the problem instances with $r = 1, \ldots, 4$ for the real world problem

RTS=4 where less than one percent of the possible columns are generated (3,424 versus 389,661).

The number of minutes required to solve the root LP and the primal heuristics are listed along with the total execution time. The number of branch-and-bound nodes (listed under "B&B") and the number of LPs solved (listed under "LPs"), as well as the execution times, provides an indication of how difficult the problem instance is. As expected the instances with larger candidate routings sets are more difficult to solve since more iterations of column generation will generally be required and each column generation subproblem takes longer since the commodity networks are larger. The final integer solution may be the result of an LP solution that is integral or the result of a tight solution by the primal heuristic. The column "LP/PR" indicates whether the final integer solution is the result of an integral LP or a solution from the primal heuristic.

The quality of the solutions obtained is measured by the integrality gap "I-Gap" and "DF-Gap" based on the known lower bound by Dionne and Florian. The integrality gap is the percentage difference of the final integer solution with respect to the lowest LP bound for any unexplored branch-and-bound node. Thus, I-Gap represents how closely the final integer solution solves the instance of the RBP given the candidate routings. For three of the four test problems, the I-Gap is less than one one-hundredth of one percent. DF-Gap on the other hand measures the percentage difference between our integer solution and the Dionne-Florian lower bound (denoted DF-independent) described in §4.5.4 which is independent of any routing assumptions. The DF-independent bound does not consider the volume or priority constraints, so it may not be a tight bound. For example, in the DF-independent bound a commodity with both an origin and destination in Florida

could have a routing using a terminal in Maine. Thus, the DF-gap is a bound on the maximum improvement that would be possible if we were able to include all routings and then solve our model to minimize the number of car-handlings.

The integer solutions (with very small integrality gaps) are found without resorting to branching which gives empirical evidence that our LP lower bounds and heuristic upper bounds are tight.

For each of these real world problems our algorithm identifies an integer solution within 0.05% of optimality for the restricted candidate routings and no further than 2.33% away from a known lower bound which is independent of any routing restrictions. However, reaching this independent lower bound on car-handles will require an increase in car-miles costs. For example, when RTS=3, reducing the system-wide car-handles below our solution will require at least one commodity to use a routing which is not one of the three "best" routes. Since the candidate routing sets implicitly limit the car-miles costs associated with the blocking plan, increases in the car-miles costs for a solution which uses different routings may offset any savings of car-handling costs.

### 5.2.2 Problem instances with terminal capacities scaled

For the problems reported in this section we scale the terminal blocking capacities $B(i)$ by the proportion BLK and the terminal volume capacities by the proportion VOL as explained above. We set the BLK and VOL factors between 0.75 and 1.0 so that we have test cases with 75% to 100% of the terminal capacities represented in the data for the real problem. By testing our algorithm on cases with tighter capacities, we provide

empirical evidence that our model is robust and gain insight into which problem instances are more difficult for our solution procedure.

For these test cases, we set the required integrality gap for the integer solutions at 0.25%. Further, we enforce an upper bound of four hours execution time on the solution for each instance. If an integer solution within the integrality gap has not been found at four hours, we report the best integer solution found so far and halt the solution process. The four hour limit on execution is enforced simply as a convenience for running these experiments.

All of the problem instances which were stopped because the execution time reached four hours occur when the blocking capacity at each terminal is reduced by 25%. Over the ranges covered by the factor level combinations, the effect of reducing the blocking capacity at each terminal is the strongest. The average execution times for different factor level combinations are shown in Tables 5.5 and 5.6. Within each BLK level, increased candidate routings (RTS) correspond to more difficult problems. These larger RTS values will result in larger space-time networks for each commodity. Since the shortest path for these networks must be found for each new column generated, the execution time for each iteration of column generation will be longer. We expected to find that within each BLK level, increased VOL would correspond to easier problem instances. Higher volume capacities should allow more of the commodities to take the most attractive blocking path since the corresponding $f_q^k$ variables are not restricted by the volume constraint (4.1e). Larger $f_q^k$ values would push the $y_a$ variables to integer

| Problem Instance | | | Columns | | Minutes | | | | | | Solution Quality | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BLK | VOL | RTS | Root | Total | Root LP | Primal | Total | B&B | LPs | PR/LP | I-Gap | DF-Gap |
| 1 | 0.75 | 3 | 3,386 | 3,386 | 42.89 | 1.09 | 43.98 | 2 | 28 | PR | 0.000% | 1.522% |
| 1 | 0.75 | 2 | 3,229 | 3,229 | 19.83 | 0.7 | 20.53 | 2 | 23 | PR | 0.007% | 1.579% |
| 1 | 0.75 | 1 | 2,562 | 2,562 | 1.05 | 0.12 | 1.16 | 2 | 12 | PR | 0.048% | 2.326% |
| 0.88 | 1 | 3 | 3,854 | 3,908 | 49.13 | 2.89 | 52.02 | 19 | 68 | LP | 0.054% | 2.037% |
| 0.88 | 1 | 2 | 3,648 | 3,671 | 31.04 | 2.21 | 42.87 | 11 | 57 | LP | 0.042% | 2.121% |
| 0.88 | 1 | 1 | 2,752 | 2,752 | 1.17 | 0.12 | 1.3 | 2 | 12 | PR | 0.021% | 3.080% |
| 0.88 | 0.75 | 3 | 3,833 | 3,833 | 53.93 | 3.23 | 57.16 | 2 | 36 | PR | 0.111% | 2.145% |
| 0.88 | 0.75 | 2 | 3,637 | 3,673 | 22.91 | 0.96 | 38.79 | 13 | 50 | PR | 0.075% | 2.161% |
| 0.88 | 0.75 | 1 | 2,757 | 2,757 | 1.21 | 0.14 | 1.35 | 2 | 13 | PR | 0.030% | 3.092% |
| 0.75 | 1 | 3 | 5,046 | 5,311 | 80.94 | 4.13 | 240.56 | 24 | 171 | PR | 0.279% | 3.813% |
| 0.75 | 1 | 2 | 4,646 | 5,129 | 40.1 | 2.64 | 240.06 | 106 | 368 | PR | 0.354% | 4.074% |
| 0.75 | 1 | 1 | 3,105 | 3,105 | 1.55 | 0.19 | 1.74 | 2 | 15 | PR | 0.159% | 5.267% |
| 0.75 | 0.75 | 3 | 5,086 | 5,373 | 56.05 | 4.43 | 240.47 | 36 | 172 | PR | 0.330% | 3.901% |
| 0.75 | 0.75 | 2 | 4,678 | 5,272 | 31.59 | 3.22 | 240.35 | 99 | 362 | PR | 0.314% | 4.064% |
| 0.75 | 0.75 | 1 | 3,106 | 3,106 | 1.72 | 0.25 | 1.97 | 2 | 17 | PR | 0.212% | 5.341% |

Table 5.4: Results for the problem instances with scaled capacities

| | VOL | | |
|---|---|---|---|
| BLK | 0.75 | 1 | Row Average |
| 0.75 | 160.93 | 160.79 | 160.86 |
| 0.875 | 32.43 | 32.06 | 32.25 |
| 1 | 21.89 | 21.43 | 21.66 |
| Column Average | 71.75 | 71.43 | 71.59 |

Table 5.5: Average execution times as a function of VOL and BLK

values because of the forcing constraints

$$\sum_{q \in Q(k)} f_q^k \, \delta_a^q - y_a \leq 0 \; \forall a \in A, k \in K. \tag{4.3g}$$

However, the results for this set of test problems do not support this expectation.

The test cases where RTS=1, meaning that the only candidate routing for each commodity is the routing historically used by the railroad, are particularly easy to solve even with a 25% reduction in both the volume and blocking capacities of each terminal. For these problems, the preprocessing step is able to set all but 2,332 binary variables which is less than half of the number left un-fixed for all other routing restrictions. Our analysis of the data from a major domestic railroad indicates that they use the same routing for each commodity almost without exception. Consequently, only allowing one candidate routing may be satisfactory in many cases. For these test cases, the longest solution takes under two minutes. As expected the cases where only one routing is allowed for each commodity have the largest DF-gap which bounds the maximum improvement which could be gained by adding all other routings. Even for these cases, the DF-gap is under 5.4%.

| | RTS | | | |
|---|---|---|---|---|
| BLK | 1 | 2 | 3 | Row Average |
| 0.75 | 1.86 | 240.21 | 240.52 | 160.86 |
| 0.875 | 1.33 | 40.83 | 54.59 | 32.25 |
| 1 | 1.18 | 19.81 | 44.00 | 21.66 |
| Column Average | 1.45 | 100.28 | 113.03 | 71.59 |

Table 5.6: Average execution times as a function of RTS and BLK

### 5.2.3 Problem instances for random terminal capacities

We report the results for the problem instances based on random capacities in Table 5.7. The methodology for constructing these random test instances is explained in §5.1. As for the scaled test instances, we set the required integrality gap to 0.25% and enforce a four hour limit on executions for convenience. Again, it is evident that lower settings for BLK and higher settings for RTS correspond to more difficult instances. We present a statistical analysis based on these random instances below.

## 5.3 Statistical analysis

We may assess the effect of each of the factors BLK, VOL, RTS and their interactions on the total execution time of our algorithm using ANOVA (analysis of variance). We want to explain the effects on the total execution time by changes in the factors. Let BLK*VOL represent the two-way interaction between the factors BLK and VOL and similar notation represent the remaining interaction terms. Our experimental design is an orthogonal mixed level design with three levels for the factors BLK and RTS and two

| Problem Instance | | | Columns | | Minutes | | | | | | Solution Quality | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BLK | VOL | RTS | Root | Total | Root LP | Primal | Total | B& B | LPs | PR/LP | I-Gap | DF-Gap |
| 1.000 | 1.00 | 1 | 1,237 | 2,510 | 1.00 | 0.00 | 1.14 | 1 | 12 | LP | 0.00% | 2.07% |
| 1.000 | 1.00 | 1 | 2,523 | 2,523 | 1.04 | 0.10 | 1.14 | 2 | 12 | PR | 0.01% | 2.09% |
| 1.000 | 1.00 | 2 | 3,053 | 3,053 | 17.59 | 0.78 | 18.36 | 2 | 20 | PR | 0.05% | 1.36% |
| 1.000 | 1.00 | 2 | 3,065 | 3,065 | 18.35 | 0.81 | 19.16 | 2 | 21 | PR | 0.04% | 1.35% |
| 1.000 | 1.00 | 3 | 1,237 | 3,227 | 45.00 | 0.90 | 46.22 | 1 | 30 | PR | 0.00% | 1.36% |
| 1.000 | 1.00 | 3 | 3,213 | 3,213 | 42.50 | 1.32 | 43.83 | 2 | 28 | PR | 0.18% | 1.37% |
| 1.000 | 0.75 | 1 | 2,490 | 2,490 | 1.06 | 0.10 | 1.17 | 2 | 12 | PR | 0.03% | 1.89% |
| 1.000 | 0.75 | 1 | 2,498 | 2,498 | 1.03 | 0.10 | 1.13 | 2 | 12 | PR | 0.03% | 1.96% |
| 1.000 | 0.75 | 2 | 3,099 | 3,099 | 18.00 | 0.75 | 18.75 | 2 | 20 | PR | 0.01% | 1.28% |
| 1.000 | 0.75 | 2 | 3,027 | 3,027 | 17.56 | 0.76 | 18.32 | 2 | 20 | PR | 0.02% | 1.34% |
| 1.000 | 0.75 | 3 | 3,145 | 3,145 | 40.33 | 1.26 | 41.59 | 2 | 26 | PR | 0.04% | 1.24% |
| 1.000 | 0.75 | 3 | 3,204 | 3,204 | 40.07 | 1.04 | 41.11 | 2 | 26 | PR | 0.01% | 1.36% |
| 0.875 | 1.00 | 1 | 2,639 | 2,639 | 1.09 | 0.14 | 1.23 | 2 | 12 | PR | 0.06% | 2.47% |
| 0.875 | 1.00 | 1 | 2,649 | 2,649 | 1.17 | 0.14 | 1.31 | 2 | 13 | PR | 0.03% | 2.53% |
| 0.875 | 1.00 | 2 | 3,384 | 3,384 | 22.92 | 1.39 | 24.31 | 2 | 28 | PR | 0.15% | 1.82% |
| 0.875 | 1.00 | 2 | 3,358 | 3,385 | 24.02 | 1.56 | 38.81 | 14 | 55 | LP | 0.04% | 1.35% |
| 0.875 | 1.00 | 3 | 3,603 | 3,605 | 52.22 | 2.85 | 73.40 | 12 | 49 | PR | 0.06% | 1.73% |
| 0.875 | 1.00 | 3 | 3,578 | 3,578 | 53.77 | 1.46 | 55.22 | 2 | 36 | PR | 0.03% | 1.65% |
| 0.875 | 0.75 | 1 | 2,692 | 2,692 | 1.16 | 0.13 | 1.29 | 2 | 13 | PR | 0.03% | 2.66% |
| 0.875 | 0.75 | 1 | 2,635 | 2,635 | 1.10 | 0.13 | 1.23 | 2 | 12 | PR | 0.03% | 2.56% |
| 0.875 | 0.75 | 2 | 3,459 | 3,459 | 21.03 | 1.24 | 22.27 | 2 | 24 | PR | 0.13% | 1.89% |
| 0.875 | 0.75 | 2 | 3,441 | 3,441 | 20.85 | 1.31 | 22.16 | 2 | 24 | PR | 0.14% | 1.89% |
| 0.875 | 0.75 | 3 | 3,635 | 3,635 | 53.33 | 1.26 | 54.58 | 1 | 36 | PR | 0.04% | 1.81% |
| 0.875 | 0.75 | 3 | 4,554 | 4,747 | 49.81 | 6.27 | 241.26 | 73 | 181 | LP | 0.26% | 3.22% |
| 0.750 | 1.00 | 1 | 2,975 | 2,975 | 1.51 | 0.13 | 1.64 | 2 | 17 | PR | 0.09% | 4.27% |
| 0.750 | 1.00 | 1 | 2,949 | 2,949 | 1.53 | 0.19 | 1.73 | 2 | 17 | PR | 0.21% | 4.60% |
| 0.750 | 1.00 | 2 | 4,243 | 4,539 | 28.14 | 2.27 | 240.09 | 206 | 383 | LP | 0.26% | 3.26% |
| 0.750 | 1.00 | 2 | 4,257 | 4,354 | 24.31 | 2.32 | 64.01 | 25 | 89 | LP | 0.11% | 3.24% |
| 0.750 | 1.00 | 3 | 4,648 | 4,792 | 71.81 | 1.83 | 150.35 | 24 | 108 | PR | 0.14% | 3.00% |
| 0.750 | 1.00 | 3 | 4,631 | 4,845 | 68.44 | 2.52 | 240.61 | 41 | 188 | PR | 0.33% | 3.25% |
| 0.750 | 0.75 | 1 | 2,974 | 2,974 | 1.62 | 0.15 | 1.76 | 2 | 18 | PR | 0.15% | 4.41% |
| 0.750 | 0.75 | 1 | 2,965 | 2,977 | 99.63 | 15.55 | 173.93 | 19 | 44 | LP | 0.00% | 4.47% |
| 0.750 | 0.75 | 2 | 4,239 | 4,366 | 31.29 | 4.54 | 82.04 | 45 | 119 | LP | 0.25% | 3.17% |
| 0.750 | 0.75 | 2 | 4,335 | 4,676 | 28.39 | 1.79 | 240.09 | 222 | 384 | LP | 0.34% | 3.28% |
| 0.750 | 0.75 | 3 | 4,717 | 4,913 | 77.15 | 2.29 | 240.05 | 40 | 179 | PR | 0.38% | 3.15% |
| 0.750 | 0.75 | 3 | 4,554 | 4,747 | 49.81 | 3.19 | 241.26 | 73 | 181 | LP | 0.26% | 3.22% |

Table 5.7: Results for the random problem instances

General Linear Models Procedure
Class Level Information

| Class | Levels | Values |
|---|---|---|
| BLK | 3 | 1 0.75 0.875 |
| VOL | 2 | 1 0.75 |
| RTS | 3 | 1 2 3 |

Number of observations in data set = 36

Dependent Variable: MINUTES

| Source | DF | Sum of Squares | Mean Square | F Value | Pr > F |
|---|---|---|---|---|---|
| Model | 17 | 200629.173 | 11801.716 | 3.29 | 0.0081 |
| Error | 18 | 64664.368 | 3592.465 | | |
| Corrected Total | 35 | 265293.541 | | | |

| R-Square | C.V. | Root MSE | MINUTES Mean |
|---|---|---|---|
| 0.756254 | 87.48001 | 59.9372 | 68.5153 |

| Source | DF | Type I SS | Mean Square | F Value | Pr > F |
|---|---|---|---|---|---|
| BLK | 2 | 94846.5937 | 47423.2968 | 13.20 | 0.0003 |
| VOL | 1 | 4702.4761 | 4702.4761 | 1.31 | 0.2676 |
| BLK*VOL | 2 | 3468.4290 | 1734.2145 | 0.48 | 0.6249 |
| RTS | 2 | 68460.2363 | 34230.1182 | 9.53 | 0.0015 |
| BLK*RTS | 4 | 21864.4478 | 5466.1119 | 1.52 | 0.2380 |
| VOL*RTS | 2 | 2020.1116 | 1010.0558 | 0.28 | 0.7582 |
| BLK*VOL*RTS | 4 | 5266.8787 | 1316.7197 | 0.37 | 0.8293 |

| Source | DF | Type III SS | Mean Square | F Value | Pr > F |
|---|---|---|---|---|---|
| BLK | 2 | 94298.4761 | 47149.2381 | 13.12 | 0.0003 |
| VOL | 1 | 5351.8671 | 5351.8671 | 1.49 | 0.2380 |
| BLK*VOL | 2 | 3540.7070 | 1770.3535 | 0.49 | 0.6189 |
| RTS | 2 | 68380.0754 | 34190.0377 | 9.52 | 0.0015 |
| BLK*RTS | 4 | 20626.8726 | 5156.7182 | 1.44 | 0.2629 |
| VOL*RTS | 2 | 2058.0470 | 1029.0235 | 0.29 | 0.7543 |
| BLK*VOL*RTS | 4 | 5266.8787 | 1316.7197 | 0.37 | 0.8293 |

Figure 5.1: ANOVA Results for the random test cases from SAS

levels for the factor VOL, which in the previous results did not seem to have a strong effect. We consider all possible interactions between the factors.

We are interested only in determining which factors have a significant effect on the minutes required for execution. Consequently, we do not solve for the regression coefficients. In Figure 5.1 we present the results from applying the SAS [39] general linear models procedure to the results from the randomized test instances. The first section of the SAS output (above the dashed line) reports statistics on the adequacy of the model at explaining the variation observed in the dependent variable Minutes. It is well known that the difficulty of similar size problems can vary greatly in Integer Programming. Consequently, we are not surprised that this model accounts for only about 75% of the variation in the observed execution times (R-Square=.756254). The entry for the model in the column Pr>F of 0.0081 matches this statistical statement: we would reject the null hypothesis that the model does not explain the variation in the execution time (Minutes) at a significance level smaller than 1%.

The second section provides information about factor effects. Each of the independent variables and their interactions in our model appears on a separate line with the measure that corresponds to its effect on the model. We base the following analysis on the right hand column Pr>F for the Type I sum of squares. The strong effect of the BLK factor is readily apparent. We would reject the hypothesis that the factor BLK does not effect the the execution time (Minutes) at a significance level of less than 1%. The number of routings RTS has almost the same measurable effect. Problems with large RTS values have both more potential columns, more potential blocks after preprocessing, and larger commodity networks.

Varying VOL (classification volume at each terminal) over the range from 75% to 100% of the real-world data has only a moderate measured effect on the execution time. Similarly, the factor-level combinations in which VOL participates do not have a strong measurable effect. In fact, we would fail to reject the null hypothesises that any of the two-way or three-way interactions do not have a measurable effect at any significance level lower than 10%.

## 5.4 Improving the efficiency of our branch-and-price algorithm

For the test cases above, in our algorithm we solve the LP relaxation of each branch-and-bound node to optimality. Solving these LPs to optimality may require generating a considerable number of columns and is only required before fathoming a subtree based on its LP bound. Since some of the columns added may no longer be attractive after additional branching, branching without completely reoptimizing the LP may be advantageous after the root node. Additionally, at each iteration of adding columns, we simultaneously solve $SP_k$ for all commodities. However, it may be more efficient to only generate columns for some of the commodities at each iteration. For example, when entering a branch-and-bound node we might solve $SP_k$ for all commodities and thereafter only solve the $SP_k$ for commodities for which a new column has recently been added. When this procedure fails to identify new columns, we could either branch or repeat the procedure.

We expected the version of the Dionne-Florian (DF) lower bound which is dependent on the candidate routings to allow fathoming some branch-and-bound nodes without having to solve the LP relaxation; however, in our computational experiments, we did not

find any cases where this bound allowed early fathoming. We considered changing to the Gallo lower bound calculation for the blocking problem as described in §4.5.4. This lower bound calculation requires only slightly more overhead than the DF bound and dominates the DF bound. At the root of the branch-and-bound tree these bounds are identical. However, for several test cases, this Gallo lower bound also did not allow fathoming nodes. Consequently, we recommend not using either of these non-LP lower bounds within the branch-and-bound tree; however, computing the DF-independent lower bound at the root node is important to gauge solution quality.

# Chapter 6

## Future Directions and Summary

Since railroads only make significant changes to their blocking plan infrequently, our solution procedure for RBP was designed as a tactical decision tool. However, since our algorithm is finding good solutions in a few hours, there are several opportunities for other uses of the model. A few potential directions are to extend our model to include blocking decisions for traffic interchanged with other railroads, to update the blocking plan more frequently, to extend our model to accommodate other decisions, or to incorporate our solution as a subproblem of a larger system. We provide a brief discussion of these directions next, then discuss opportunities to apply our research to other problems and summarize our results.

### 6.1 Extensions to our model for RBP

In this section, we discuss extending our model for RBP by including more blocking decisions. We also discuss several ideas to include other decisions concurrently with RBP. We separate these ideas using Assad's [3] categorization of rail decisions into operational decisions which are made frequently, tactical decisions with intermediate time horizons, and strategic decisions with long time horizons. (See

#### 6.1.1 Including more blocking decisions

**Extending our model to include interchange traffic** Often railroads receive traffic which originated on another railroad. In these cases, typically the two railroads negotiate

which blocks this "interchanged" traffic will be sorted into and at which terminals these blocks will be interchanged. If the negotiations allow the receiving railroad to specify what blocks are used for this traffic, this interchange activity can be incorporated into our model in an analogous manner to Van Dyke [40].

For exposition, suppose our model is being used by Reading Railroad which has interchanges with Pennsylvania Railroad and Short Railroad. We define an "interchange" terminal for each terminal (in either rail system) at which Reading Railroad interchanges traffic with Short Railroad and for each terminal at which Reading Railroad interchanges traffic with Pennsylvania Railroad. The interchange terminals correspond to terminal-railroad pairs. We may model restrictions on the number of blocks for the traffic coming from each of these interchange terminals by setting the blocking capacity $B(i)$ accordingly; e.g., if Short Railroad interchanges traffic with Reading Railroad at terminal 1 and has agreed to sort the traffic for Reading Railroad into three blocks, then $B(1) = 3$. The volume capacity $V(i)$ would match the volume of the traffic flowing from this interchange terminal into the Reading Railroad system. From the point of view of Reading railroad, this traffic will have the interchange terminal as its origin. Since these interchange terminals would only handle traffic which either originates or terminates there, we designate these terminal as "end" terminals for our model. When our model is solved for Reading Railroad the resulting blocking plan will include the blocks for the interchanged traffic which minimize the number of system-wide handlings.

**Allowing parallel blocks** One of our assumptions is that blocks are completely specified by their origin and destination (OD). Thus, the limit $B(i)$ on the number of blocks for terminal $i$ is the number of destinations to which terminal $i$ can build blocks. These

blocks may be assembled on multiple tracks to allow further sorting of the traffic assigned to them. We could model parallel blocks—ones with the same OD—to allow having high-priority blocks and low-priority blocks. High-priority commodities could be assigned to the high-priority blocks which would travel on trains with the fastest service to the block's destination. In a graph of the blocking plan, the blocks with the same OD would be parallel arcs. Our model can be easily extended to allow for these parallel blocks. Each block ($a \in A$) would correspond to an (origin, destination, priority) triplet and the $B(i)$ capacities would be increased to reflect this more detailed description of the blocks. While this extended model would be the same mathematically, for the same size networks it will become more difficult. More binary variables and thus more bundle constraints (4.1b) will be needed. The number of possible forcing constraints (4.3g) would also increase.

### 6.1.2 Blending with operational decisions

If we can consistently get solutions in a few hours, then the branch-and-price procedure presented here could be used for weekly and perhaps daily changes to the blocking plan. This is much more frequent than railroads are currently able to make significant changes to their blocking plan. Our algorithm might be particularly useful for disaster recovery. For example, during the 1993 floods along the Mississippi River, the blocking plan could have been re-optimized with terminals and links which were becoming inoperable removed and the origin for each commodity updated to the terminal that it would next stop at along its current routing. At most railroads during this flood, the blocking plans were modified without optimization-based tools.

For a pre-specified blocking plan, our BLOCK formulation can be used to efficiently assign blocking paths to commodities which are consistent with their priority constraints and which lead to the lowest number of system-wide handlings. Since the binary $y_a$ variables would be fixed, the forcing constraints would not need to be added to BLOCK and branching would not be necessary. We anticipate that the solution time of BLOCK would be fast enough to permit its use in a real-time system to assign blocking paths based on a pre-defined blocking plan. However, our model does not capture the day-of-week variation that would be important in such systems.

**Including block-to-track assignments** At each terminal, the block-to-track assignment problem is to choose individual classification track(s) on which to assemble each block. This assignment may change daily based on the commodities which will arrive. Several blocks may be assigned to the same track (which may necessitate further sorting of this track) or a block may be assigned to more than one track. If the blocking plan can be re-optimized quickly, then there is an opportunity to determine the blocking plan and block-to-track assignments simultaneously.

**Ordering trains for classification** An important decision which must be made many times daily at each terminal is what order to select the trains from the receiving yard for classification. The schedules and composition of in-bound and out-bound trains, as well as, their position in the receiving or departure yard are important inputs for this decision. It is possible for a train to be split across more than one receiving yard track or to select partial trains. Armacost [2] applies two-machine sequencing theory to

this problem assuming that the blocking plan, block-to-track assignments, block-to-train assignments, and order of the blocks on the out-bound trains are fixed.

By combining the decisions of which blocks to build with the block-to-track assignments and the order to select trains for classification, a terminal's overall efficiency could be enhanced. Since the updated blocking plan would also consider the effects on other terminals, system-wide interactions would be captured.

### 6.1.3 Blending with strategic decisions

Our model could be used in sensitivity studies to predict the effect on the system-wide handlings resulting from changing the blocking capacities at specific terminals. Used in this mode, our model can help select terminals for capital improvement projects. The values of the dual variables corresponding to the blocking capacity constraints for the root LP may be useful in indicating which terminals to include in these sensitivity studies. In a similar vein, our algorithm can be used as a what-if tool to find the effect on the blocking plan when considering adding or removing new terminals or links or entire railroads through mergers.

### 6.1.4 Including other tactical decisions

As we mentioned in our introduction of RBP, the blocking plan is an integral part of the overall plan for operating a railroad. One major domestic railroad breaks the overall operating plan down into the following components. This breakdown is somewhat different from the taxonomy introduced by Assad (see Figure 3.5 on page 36).

- Blocking Plan

- Train Profiles (block-to-train assignments, train routes and schedules)
- Routing Schedule (class tracking)
- Terminal Connection Standards
- Crew Schedule
- Locomotive Schedule

Since the blocking plan is part of the integral operations plan, any extensions which capture information from or also optimize over the other components would be valuable. We perceive the *modus operandi* of the railroads to be to generate the blocking plan first; however, additional efficiency improvements are obviously possible if consideration of the other components could be made when generating the blocking plan. The difficulty is including these extensions in such a manner that the resulting model is still tractable. We discuss a few ideas to include other tactical decisions below.

**Including Block-to-Train Assignments** The composition of a train changes at each stop as blocks are "set-out" of the train and others are "set-in." A train with a long route may often reach its termination point without any of the cars which were originally included in the train. The capacity of the train depends on the number and power of the locomotives used, as well as, the characteristics of the train route. The locomotives also may change at each stop the train makes. Besides the difficulty of the changing nature of a train and its capacities, including the block-to-train assignments is also complicated by the practice of "block swaps." A block swap is the practice of using 2 or more trains, each providing partial delivery of a block—i.e., the first train delivers the block to an intermediate stop at which the second train picks it and delivers it to another stop, etc.,

until the block reaches its destination. Block swaps prevent a many-to-one assignment of blocks to trains.

**Including locomotive scheduling** Because of the high cost of locomotives, scheduling their use is an area where increased efficiency could have a dramatic impact. Whereas trains are essentially logical designations for a collection of locomotives and blocks, a locomotive must periodically return to a particular depot(s) for maintenance. It may be possible to use column generation to implicitly consider the huge number of possible routings for locomotives and have a master problem simultaneously select blocks and matching locomotive routings. A locomotive routing and its assigned blocks could then be considered a "mini-train." These "mini-trains" would then need to be combined to form trains (with schedules or timetables) which are feasible for line capacities, crew capacities, etc.

We should point out that the model Crainic *et al.* [14] propose incorporates several of the tactical decisions including the blocking problem. In their model, possible train services are specified *a priori.* Using a non-linear objective, they seek to minimize the costs to deliver all commodities. The literature for their model ([14] and [15]) does not indicate the problem sizes which have been solved or the solution times, but does state that successful experiments of their model have been conducted for Canadian and French railway data. See §3.4 for more details on their approach.

**Inclusion of our method within a larger system** Our system could be used as a subproblem of a larger system which also includes subproblems to solve some of the

other components of the operations plan. Research to develop one such system is being conducted at C.S. Draper Laboratories, Inc.

If our model is extended or incorporated into another system, it may be necessary to aggregate the network to a smaller network of important terminals. While this would decrease the resolution of the blocking plan, it is very likely that the payoff from incorporating additional components of the operations plan would more than offset the value of the lost resolution.

## 6.2 Extensions of our work to other problems

Our branch-and-price solution procedure for RBP may be applicable to the related LTL (less than truckload) problem which was described on page 44. Instead of selecting blocking paths for each commodity, LTL assigns the commodity to a sequence of truckloads. The decision for each consolidation point in the LTL system is the selection of destinations for the collection bays. Since these assignments are updated very frequently, the solution procedure would have to be faster than is required for the RBP. We have not investigated the LTL problem enough to know if the time required for our solutions procedure would be acceptable.

The branch-and-price algorithm which we presented for the budget design problem with node constraints (BDP) on page 22 is applicable to any budget design problem (even if the node budget constraints (2.1e) are not present). Previous branch-and-bound approaches to solve these problems have relied on bounds with low polynomial complexity used in a branch-and-bound search. If the forcing constraints can still be included, then the LP bounds of our restricted master problem may be much tighter than the

polynomial bounds. Additionally, our LP based bound will be more flexible since other side constraints could be added easily to the formulation. While similar inclusion into the polynomial bounds may be possible for some constraints, it may not be trivial. The node volume constraints of RBP are an example of this type of side constraints. The structure of the $SP_k$ problems will change as constraints are added. Necessary changes caused by added constraints and by application specific constraints on the legal paths may require a different approach to solving the subproblems than the alternatives we presented for the RBP.

## 6.3 Summary

We presented a branch-and-price algorithm for the directed budget design problems when additional budget constraints are placed on some nodes. The usefulness of our algorithm is demonstrated by its application to the railroad blocking problem (RBP). While there is no guarantee that our solution procedure for RBP will find an integer solution in a reasonable amount of time for any instance, in our computational tests based on real-world data it has identified a blocking plan within a few percent of a known lower bound. The search for this integer solution is aided by tight lower bounds from the LP solutions (with the forcing constraints added) at each branch-and-bound node coupled with a tight upper bound from our primal rounding heuristic. The column generation required to solve the LPs is accomplished with a fast acyclic shortest path algorithm.

## BIBLIOGRAPHY


[1] Ahuja, R.K. and V.V.S. Murty (1987), "New Lower Planes for the Network Design Problem," *Networks* 17:113-127.

[2] Armacost, A.P. (1995), "Modeling Railroad Terminal Operations: Supporting Real-Time Network Planning and Control," Masters Thesis, Massachusetts Institute of Technology, Boston, Massachusetts.

[3] Assad, A. A. (1980), "Modeling of Rail Networks: Toward a Routing/Makeup Model," *Transportation Research B* , 14B:101-114.

[4] Assad, A. A. (1982), "A class of Train-Scheduling Problems" *Transportation Science* 16:281-310.

[5] Balakrishnan, A. (1987), "LP Extreme Points and Cuts for the Fixed-Charge Network Design Problem," *Mathematical Programming* 39:263-284.

[6] Ball, M.O., T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, eds, *Handbooks in Operations Research and Management Science*, vol. 7, Network Models. Elsevier, New York, 1995.

[7] Barnhart, C., E.L. Johnson, G.L. Nemhauser, G. Sigismondi, and P.H. Vance (1993), "Formulating a Mixed Integer Programming Problem to Improve Solvability" *Operations Research* 41:1013-1019.

[8] Barnhart, C. and Y. Sheffi (1993), "A Network-Based Primal-Dual Heuristic for the Solution of MCNF Problems" *Transportation Science* 27:102-117.

[9] Barnhart, C., E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance (1996), "Branch-and-Price: Column Generation for Solving Huge Integer Programs" *Operations Research* to appear.

[10] Barnhart, C., C.A. Hane, and E. L. Johnson (1995), *Telecommunication Systems* 3:239-258.

[11] Bodin, L.D., B.L. Golden and A.D. Schuster (1980), "A Model for the Blocking of Trains," *Transportation Research* 14:115-121.

[12] Boffey, T.B. and A.I Hinxman (1979), "Solving the Optimal Network Problem," *European Journal of Operations Research* 3:386-393.

[13] CPLEX Optimization, Inc, *Using the CPLEX Callable Library*, Incline Village, NV, 1994.

[14] Crainic T.G., J.A. Ferland, and J.M. Rousseau (1984), "A Tactical Planning Model for Rail Freight Transportation," *Transportation Science* 18:1685-184.

[15] Crainic, T. (1986), "Rail Tactical Planning: Issues, Models and Tools," *Proceeding of the Internations seminar on Freight Transportation Planning and Logistics*, Bressanone, Italy.

[16] Dionne, R. and M. Florian (1979), "Exact and Approximate Algorithms for Optimal Network Design," *Networks* 9:37-59.

[17] Farvolden, J.M, W.B. Powell, and I.J. Lustig (1993), "A Primal Partitioning Solution for the Arc-Chain Formulation of a Multicommodity Network Flow Problem" *Operations Research* 41:669-693.

[18] Farvolden,J.M. and W. B. Powell (1994), "Subgradient Methods for the Service Network Design Problem" *Transportation Science* 28:256-272.

[19] Gallo, G. (1983), "Lower Planes for the Network Design Problems," *Networks* 13:411-426.

[20] Garey, M.R. and D.S. Johnson, *Computers and Intractability: A guide to the Theory of NP-Completeness*, Freeeman, San Francisco, 1979.

[21] Gorman, M.F. (1995), "An Application of Genetic and Tabu Searches to the Freight Railroad Operation Plan Problem," presented at INFORMS Spring 95 Meeting.

[22] Gorman, M.F., chair (1996) "Recent Research in Rail Applications," sponsored session at the Informs Spring 1996 meeting, Washington, DC.

[23] Haghani, A. E. (1989), "Formulation and Solution of Combined Train Routing and Makeup, and Empty Car Distribution Model," *Transportation Research* 23B:433-452.

[24] Hellstrand, J., T. Larsson, and A. Migdalas (1992), "A Characterization of the Uncapacitated Network Design Polytope," *Operations Research Letters* 12, 159-163.

[25] Hoang, H.H. (1973). "A computational Approach to the Selection of an Optimal Network," *Management Science* 19, 488-498.

[26] T.C. Hu (1974), "Optimum Communication Spanning Trees," *SIAM J. Computing* 3:188-195.

[27] Huntley, C.L., D.E. Brown, D.E. Sappington, and B.P. Markowicz (1995), "Freight Routing and Scheduling at CSX Transportation,"Interfaces 25:58-71.

[28] Johnson, D.S., J.K. Lenstra and A.H.G. Rinnooy Kan (1978), "The Complexity of the Network Design Problem," *Networks* 8:279-285.

[29] Keaton, M.H. (1989), "Designing Optimal Railroad Operating Plans: Lagrangian Relaxation and Heuristic Approaches," *Transportation Research* 23B:363-374.

[30] Keaton, M.H. (1991), "Service-Cost Tradeoffs for Carload Freight Traffic in the U.S. Rail Industry," *Transportation Research* 25A:363-374.

[31] Keaton, M.H. (1992), "Designing Railroad Operating Plans: A Dual Adjustment Method for Implementing Lagrangian Relaxation" *Transportation Research* 26A:263-279.

[32] Klincewicz, J. G. (1990), "Solving a Freight Transport Problem Using Facility Location Techniques," *Operations Research* 38:99-109.

[33] Magnanti,T.L., and R.T. Wong (1984), "Network design and transportation planning : Models and algorithms," *Transportation Science* 18:1-55.

[34] Minoux, M. (1989), "Network Synthesis and Optimum Network Design Problems: Models, Solution Methods and Applications," *Networks* 19:313-360.

[35] Padberg, M.W., T.J. Van Roy, and L.A. Wolsey (1985), "Valid Linear inequalities for Fixed Charge Problems," *Operations Research* 33:842-861.

[36] Rardin, R.L. and U. Choe (1979), "Tighter Relaxations of Fixed Charge Network Flow Problems," Report J-79-18, Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

[37] Rardin, R.L. and L.A. Wolsey (1993), "Valid Inequalities And Projecting The Multicommodity Extended Formulation for Uncapacitated Fixed Charge Network Flow Problems," *European Journal of Operations Research* 71:95-109.

[38] Savelsbergh, M.W.P., G.C. Sigismondi, and G.L. Nemhauser, "A Functional Description of MINTO, A Mixed INTeger Optimizer," COC-93-02, Georgia Institute of Technology, 1993.

[39] Statistical Analysis Software (SAS), Release 6.09, SAS Institute Inc, Cary, NC.

[40] Van Dyke, C.D. (1986), "The Automated Blocking Model: A Practical Approach to Freight Railroad Blocking Plan Development. *Transportation Research Forum* 27:116-121.

[41] Van Roy, T.J., and L.A. Wolsey (1985), "Valid Inequalities and Separation for Uncapacitated Fixed Charge Networks," *Operations Research Letters* 4:105-112.

[42] Wong, R.T. (1980) "Worst-Case Analysis of Network Design Problem Heuristics," *SIAM Journal of Algorithms and Discrete Methods* 1:51-63.